OPTIMIZATION-BASED METHODS FOR CLASSIFICATION AND REGRESSION PROBLEMS WITH IMPRECISE DATA

Doctoral Thesis

José F. Gordillo Santofimia

February, 2008

Supervisors:

Dr. Emilio Carrizosa

Dr. Frank Plastria

UNIVERSIDAD DE SEVILLA

VRIJE UNIVERSITEIT BRUSSEL

Agradecimientos

"Always pain before a child is born,
Still I'm waiting for the dawn"

Quiero dar las gracias a todas aquellas personas que, durante este período, me han hecho crecer, tanto en lo académico como en lo personal.

En primer lugar, quiero agradecer el gran trabajo que han llevado a cabo conmigo mis dos directores de tesis. Ellos han sido maestros y guías para mí, me han enseñado lo que sé sobre la labor del investigador, me han hecho aprender muchísimo y me han ayudado a superarme.

En estos años, creo que he pasado más tiempo en el Departamento de Estadística e Investigación Operativa que en mi propia casa. Allí me he encontrado muchos compañeros que se han preocupado de mí, que se han comprometido conmigo para ayudarme en mi formación, que han confiado en mí y en mi capacidad de trabajo, que han compartido conmigo muchos buenos momentos (y otras veces, simplemente, muchos momentos), que han estado ahí para escucharme. A todas esas personas, mi más sincero agradecimiento.

Muchas gracias también a la gente de MOSI Deparment en la VUB, por hacerme sentir como uno más durante los meses que estuve allí, y a todas las personas que hicieron de mi estancia en Bélgica una gran experiencia, que significó para mí toda una lección de tolerancia y de humanidad.

A mis amigos de la Peña, porque en todos estos años os habéis hecho expertos en hacerme sentir importante y querido, y a todos mis amigos, porque me siento muy afortunado de teneros.

Y como no, a mi familia, mi gran regalo.

Resumen

Métodos basados en optimización para problemas de clasificación y regresión con datos imprecisos

Este trabajo estudia aspectos de modelado y de tipo algorítmico en problemas de clasificación supervisada y de regresión. A continuación, presentamos una breve introducción y revisión de la literatura de los principales temas tratados.

0.1 Máquinas de Vectores Soporte

Las Máquinas de Vectores Soporte (en inglés, Support Vector Machines, SVMs) es una metodología basada en optimización que se ha utilizado con éxito en problemas de clasificación y regresión. En esta sección, damos una descripción general de las Máquinas de Vectores Soporte para clasificación, y en la Sección 0.2, explicamos el modelo ϵ-Regresión de Vectores Soporte (en inglés, ϵ-Support Vector Regression, ϵ-SVR), que es la adaptación de las SVMs al caso de regresión. Estas técnicas serán trabajadas en esta tesis para ser aplicadas a los casos de clasificación y regresión con datos imprecisos.

0.1.1 Descripción del problema

En las Máquinas de Vectores Soporte (ver e.g. [15, 21, 30, 31, 79, 82, 113, 114]) para problemas de clasificación, se tiene una base de datos $\Omega \subseteq \mathbb{R}^d \times \mathbb{R}$, con elementos $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, donde x_i es el vector de características e y_i es la clase a la que pertenece el elemento. Por abuso de notación, identificamos el elemento (x_i, y_i) del conjunto de datos con su índice i. En clasificación binaria, sólo hay dos clases posibles, es decir cada elemento $i \in \Omega$ tiene asignada una etiqueta $y_i = +1$, si pertenece a la clase positiva, o bien $y_i = -1$, si pertenece a la clase negativa.

Por tanto, la base de datos Ω se puede descomponer en dos grupos diferentes G_{+1} y G_{-1}. El objetivo de las Máquinas de Vectores Soporte es introducir un hiperplano, con parámetros $\omega \in \mathbb{R}^d$, $\beta \in \mathbb{R}$, separando estos dos grupos de elementos, de modo que, dado un nuevo elemento $x \in \mathbb{R}^d$, se asigna a la clase correspondiente atendiendo al $signo(f(x)) = signo(\omega^\top x + \beta)$, i.e.,

$$\begin{aligned} &\text{si } \omega^\top x + \beta > 0, \quad \text{entonces } y = +1 \\ &\text{si } \omega^\top x + \beta < 0, \quad \text{entonces } y = -1. \end{aligned}$$

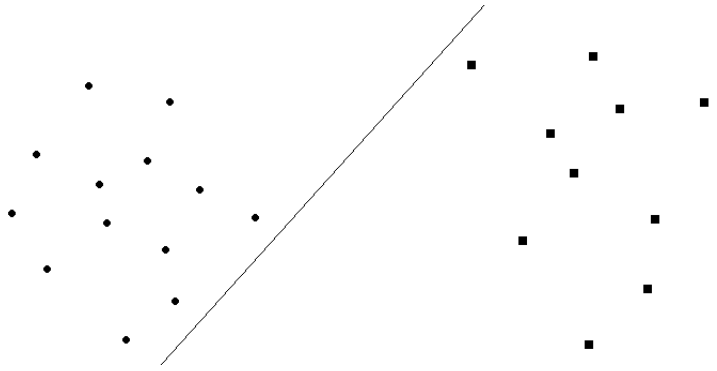

Figure 1: Un posible hiperplano separador

Cuando es posible encontrar un hiperplano separando los grupos G_{+1} y G_{-1}, se dice que los dos grupos son linealmente separables. En ese caso, existen ω, β, tales que

$$\omega^\top x_i + \beta > 0, \quad \forall i \in G_{+1}$$
$$\omega^\top x_i + \beta < 0, \quad \forall i \in G_{-1},$$

o equivalentemente,

$$y_i(\omega^\top x_i + \beta) > 0, \quad \forall i \in \Omega. \tag{1}$$

Esto resulta ser equivalente a que los cierres convexos de los conjuntos $\{x_i : \ i \in G_{+1}\}$, $\{x_i : \ i \in G_{-1}\}$ sean disjuntos (ver e.g. [21]).

En general, cuando dos grupos son linealmente separables, podemos encontrar infinitas posibles soluciones. En la Figura 1, se muestra un posible hiperplano que separa estos grupos de cuadrados y puntos, aunque, intuitivamente, no parece ser ésta la mejor solución.

La solución propuesta en el ámbito de SVMs es el hiperplano separador que maximiza el margen, donde el margen viene definido como la mínima distancia de los elementos de la base de datos al hiperplano. La Figura 2 da la idea gráfica del hiperplano con máximo margen que separa los grupos de cuadrados y círculos.

Entonces, dada una muestra de aprendizaje $I = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \subseteq \Omega$, tenemos que resolver un problema de optimización para encontrar los parámetros óptimos ω y β (óptimos en tanto que maximizan el margen) para construir la regla de clasificación.

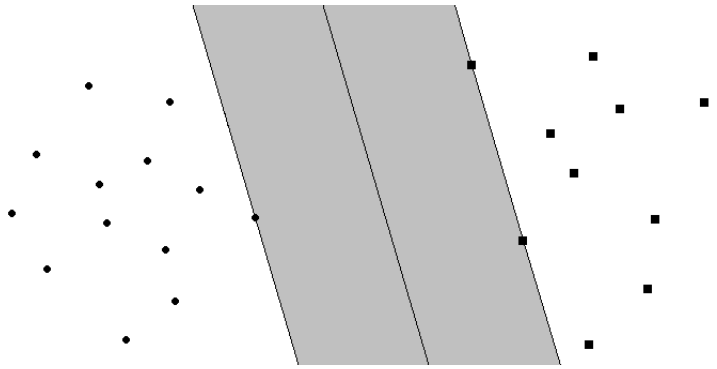

Figure 2: Clasificador con máximo margen

0.1.2 El problema de optimización

La distancia de un punto $x^* \in \mathbb{R}^d$ a un hiperplano $H = \{x \in \mathbb{R}^d : \ \omega^\top x + \beta = 0\}$ viene dada por

$$dist(x^*, H) = \frac{|\omega^\top x^* + \beta|}{\|\omega\|^0}, \tag{2}$$

donde $\|\omega\|^0$ representa la norma dual del vector ω normal al hiperplano H (ver [92]).

Por tanto, la distancia ρ_i de un elemento $i \in I$ a la región donde estará mal clasificado es

$$\rho_i(\omega, \beta) = \max\left\{\frac{y_i(\omega^\top x_i + \beta)}{\|\omega\|^0}, 0\right\}.$$

El margen en la muestra de aprendizaje I está definido como el mínimo de estas distancias,

$$\rho(\omega, \beta) = \min_{i \in I} \rho_i(\omega, \beta).$$

El objetivo es encontrar el hiperplano separador con máximo margen. Por tanto, el problema de optimización es

$$\begin{array}{ll} \max\limits_{\omega,\beta} & \rho(\omega, \beta) \\ & y_i(\omega^\top x_i + \beta) > 0, \ \forall i \in I, \end{array} \tag{3}$$

que puede ser escrito también como

$$\begin{array}{ll} \max\limits_{\omega,\beta} & \min\limits_{i \in I} \dfrac{y_i(\omega^\top x_i + \beta)}{\|\omega\|^0} \\ & y_i(\omega^\top x_i + \beta) > 0, \ \forall i \in I. \end{array} \tag{4}$$

Como el problema es positivamente homogéneo en las variables, podemos imponer que

$$\min_{i\in I}\ y_i(\omega^\top x_i+\beta)=1,$$

y entonces, el Problema (4) queda como sigue,

$$\begin{array}{ll}\max\limits_{\omega,\beta} & \dfrac{1}{\|\omega\|^0}\\ & \min\limits_{i\in I}\ y_i(\omega^\top x_i+\beta)=1,\end{array}$$

o equivalentemente,

$$\begin{array}{ll}\min\limits_{\omega,\beta} & \|\omega\|^0\\ & \min\limits_{i\in I}\ y_i(\omega^\top x_i+\beta)=1.\end{array}\tag{5}$$

Además, el Problema (5) es equivalente a

$$\begin{array}{ll}\min\limits_{\omega,\beta} & \|\omega\|^0\\ & \min\limits_{i\in I}\ y_i(\omega^\top x_i+\beta)\geq 1.\end{array}\tag{6}$$

En efecto, la región factible de (5) está incluida en la de (6), luego el valor óptimo z de (6) es menor o igual que el valor óptimo z' de (5). Además, cualquier (ω,β), factible para (6), tal que $\min\limits_{i\in I}\ y_i(\omega^\top x_i+\beta)>1$, no puede ser óptimo para (6), ya que $(\overline{\omega},\overline{\beta})$, definido como

$$\begin{aligned}\overline{\omega} &= \frac{1}{\min\limits_{i\in I}\ y_i(\omega^\top x_i+\beta)}\cdot\omega\\ \overline{\beta} &= \frac{1}{\min\limits_{i\in I}\ y_i(\omega^\top x_i+\beta)}\cdot\beta,\end{aligned}$$

es factible y verifica que $\|\overline{\omega}\|^0<\|\omega\|^0$. Por tanto, $z'=z$.

Finalmente, este problema puede ser reformulado como

$$\begin{array}{ll}\min\limits_{\omega,\beta} & \|\omega\|^0\\ & y_i(\omega^\top x_i+\beta)\geq 1,\ \forall i\in I.\end{array}\tag{7}$$

Cuando se usa la norma Euclídea, el problema a resolver es equivalente al siguiente problema cuadrático convexo con restricciones lineales,

$$\begin{array}{ll}\min\limits_{\omega,\beta} & \dfrac{1}{2}\displaystyle\sum_{j=1}^{d}\omega_j^2\\ & y_i(\omega^\top x_i+\beta)\geq 1,\ \forall i\in I.\end{array}\tag{8}$$

0.1.3 Formulación Margen-Débil

La factibilidad no está garantizada en este problema. Debemos, por tanto, introducir unas variables de holgura ξ en las restricciones y una penalización en la función objetivo. Así, se obtiene la siguiente formulación de Margen Débil (ver e.g. [15, 30]),

$$\begin{aligned} \min_{\omega,\beta,\xi} \quad & \frac{1}{2}\sum_{j=1}^{d}\omega_j^2 + C\sum_{i\in I}\xi_i \\ & y_i(\omega^\top x_i + \beta) \geq 1 - \xi_i, \ \forall i \in I \\ & \xi_i \geq 0, \ \forall i \in I, \end{aligned} \tag{9}$$

donde C es una constante del modelo que equilibra el margen para los puntos correctamente clasificados y la penalización para los errores cometidos.

0.1.4 Formulación dual

A continuación, obtendremos la formulación dual para el Problema (9) (ver e.g. [15, 30, 31]). La formulación dual nos permite usar funciones no lineales como clasificadores mediante la introducción de núcleos.

Introducimos un multiplicador no negativo para cada restricción del Problema (9), y se construye la función Lagrangiana como sigue,

$$L(\omega,\beta,\xi) = \frac{1}{2}\sum_{j=1}^{d}\omega_j^2 + C\sum_{i\in I}\xi_i + \sum_{i\in I}\lambda_i(1-\xi_i - y_i\sum_{j=1}^{d}\omega_j x_{ij} - y_i\beta) - \sum_{i\in I}\mu_i\xi_i.$$

Las condiciones de optimalidad se obtienen igualando a cero las derivadas parciales de la función L,

$$\frac{\partial L}{\partial \omega_j} = \omega_j - \sum_{i\in I}\lambda_i y_i x_{ij} = 0, \ j = 1,\ldots,d \tag{10}$$

$$\frac{\partial L}{\partial \beta} = -\sum_{i\in I}\lambda_i y_i = 0 \tag{11}$$

$$\frac{\partial L}{\partial \xi_i} = C - \lambda_i - \mu_i = 0, \ \forall i \in I. \tag{12}$$

De la restricción (10), obtenemos la expresión para calcular el vector ω, usando las variables duales,

$$\omega_j = \sum_{i\in I}\lambda_i y_i x_{ij}, \ j = 1,\ldots,d. \tag{13}$$

De (11), se obtiene una restricción que debe ser incluida en la formulación dual,

$$\sum_{i\in I}\lambda_i y_i = 0. \tag{14}$$

Como los multiplicadores μ_i son no negativos, la expresión (12) indica que los multiplicadores λ_i están acotados por la constante del modelo C,

$$0 \leq \lambda_i \leq C, \ \forall i \in I. \tag{15}$$

Así, usando estas restricciones, la función objetivo para la formulación dual es la que sigue,

$$\begin{aligned} \tilde{L}(\lambda) &= -\frac{1}{2}\sum_{j=1}^{d}(\sum_{i\in I}\lambda_i y_i x_{ij})(\sum_{l\in I}\lambda_l y_l x_{lj}) + \sum_{i\in I}\lambda_i \\ &= -\frac{1}{2}\sum_{i,l\in I}\lambda_i\lambda_l y_i y_l x_i^\top x_l + \sum_{i\in I}\lambda_i. \end{aligned}$$

Con esta función objetivo y añadiendo las restricciones (14)-(15), obtenemos la siguiente formulación dual como un problema de maximización cuadrático convexo en las variables λ,

$$\begin{aligned} \max_{\lambda} \quad & \sum_{i\in I}\lambda_i - \frac{1}{2}\sum_{i,l\in I}\lambda_i\lambda_l y_i y_l x_i^\top x_l \\ & \sum_{i\in I}\lambda_i y_i = 0 \\ & 0 \leq \lambda_i \leq C, \ \forall i \in I. \end{aligned} \tag{16}$$

Dada una solución óptima del Problema (16), se puede recuperar una solución óptima del Problema (9). En efecto, el vector ω se puede obtener con la expresión (13), y para β, usamos que se tienen que verificar las condiciones de Karush-Kuhn-Tucker,

$$\lambda_i \cdot (1 - \xi_i - y_i\sum_{j=1}^{d}\omega_j x_{ij} - y_i\beta) = 0, \quad \forall i \in I \tag{17}$$

$$\xi_i \cdot (C - \lambda_i) = 0, \quad \forall i \in I \tag{18}$$

$$0 \leq \lambda_i \leq C, \quad \forall i \in I. \tag{19}$$

Si $\exists i \in I$ tal que $0 < \lambda_i < C$, entonces, por (1.18), se tiene que $\xi_i = 0$, y de (1.17), podemos recuperar el valor para β,

$$\beta = y_i(1 - y_i\sum_{j=1}^{d}\omega_j x_{ij}) = y_i - \sum_{j=1}^{d}\omega_j x_{ij}. \tag{20}$$

En caso contrario, si $\lambda_i \in \{0, C\}$, $\forall i \in I$, cuando $\lambda_i = 0$, por (1.18), la variable de holgura es también $\xi_i = 0$, y dado que (ω, β, ξ) es factible para (9),

$$1 - y_i\sum_{j=1}^{d}\omega_j x_{ij} - y_i\beta \leq 0, \text{ i.e., } y_i\beta \geq 1 - y_i\sum_{j=1}^{d}\omega_j x_{ij},$$

por tanto,

$$\text{si } y_i = +1, \quad \text{entonces } \beta \geq y_i - \sum_{j=1}^{d} \omega_j x_{ij} \tag{21}$$

$$\text{si } y_i = -1, \quad \text{entonces } \beta \leq y_i - \sum_{j=1}^{d} \omega_j x_{ij}, \tag{22}$$

y tomando máximo y mínimo, respectivamente, en las expresiones (21) y (22), se tiene que

$$\max_{\{i \in G_{+1}: \lambda_i = 0\}} \{1 - \sum_{j=1}^{d} \omega_j x_{ij}\} \leq \beta \leq \min_{\{i \in G_{-1}: \lambda_i = 0\}} \{-1 - \sum_{j=1}^{d} \omega_j x_{ij}\}. \tag{23}$$

Por otro lado, si $\lambda_i = C$, entonces la variable de holgura puede ser positiva y su correspondiente restricción en (1.17) se vuelve activa, es decir,

$$1 - \xi_i - y_i \sum_{j=1}^{d} \omega_j x_{ij} - y_i \beta = 0, \text{ i.e., } y_i \beta \leq 1 - y_i \sum_{j=1}^{d} \omega_j x_{ij}.$$

Con un razonamiento similar al realizado para $\lambda_i = 0$, se obtiene que

$$\max_{\{i \in G_{-1}: \lambda_i = C\}} \{-1 - \sum_{j=1}^{d} \omega_j x_{ij}\} \leq \beta \leq \min_{\{i \in G_{+1}: \lambda_i = C\}} \{1 - \sum_{j=1}^{d} \omega_j x_{ij}\}. \tag{24}$$

En el otro sentido, dado λ óptimo para (16), con $\lambda_i \in \{0, C\}$, $\forall i \in I$, ω como está definido en (13), entonces cualquier β verificando (23)-(24) es óptimo para (9), es decir, existe ξ tal que (ω, β, ξ) y λ verifican conjuntamente el sistema KKT (1.17)-(1.19).

En efecto, para $i \in I$ tal que $\lambda_i = 0$, (1.17)-(1.18) se verifican trivialmente para $\xi_i = 0$.

Como β verifica (24), se tiene que

$$\beta \leq 1 - \sum_{j=1}^{d} \omega_j x_{ij}, \quad \forall i \in G_{+1} : \ \lambda_i = C \tag{25}$$

$$\beta \geq -1 - \sum_{j=1}^{d} \omega_j x_{ij}, \quad \forall i \in G_{-1} : \ \lambda_i = C. \tag{26}$$

Entonces, para cada $i \in G_{+1}$ tal que $\lambda_i = C$, por (1.17), se obtiene el valor de ξ_i,

$$\xi_i = 1 - \sum_{j=1}^{d} \omega_j x_{ij} - \beta,$$

que es no negativo por (25).

Análogamente, para cada $i \in G_{-1}$ tal que $\lambda_i = C$, por (1.17), calculamos el valor de ξ_i como

$$\xi_i = 1 + \sum_{j=1}^{d} \omega_j x_{ij} + \beta,$$

que es mayor o igual a cero por (26).

Por tanto, si cada $\lambda_i \in \{0, C\}$, entonces se puede escoger cualquier β que verifique (23)-(24).

De las condiciones KKT (1.17)-(1.19), se puede observar que los ejemplos que tienen multiplicador asociado $\lambda_i = C$ se encuentran fuera del semiespacio que contiene los ejemplos de su grupo, y aquellos ejemplos con multiplicador asociado $0 < \lambda_i < C$ se encuentran exactamente en la frontera de dicho semiespacio. A los ejemplos con multiplicador mayor que cero se les llama *vectores soporte* y son los únicos que necesitamos para calcular el clasificador. Es decir, si borramos de la muestra todos los ejemplos con $\lambda_i = 0$ y recalculamos el clasificador, obtendríamos el mismo resultado.

0.1.5 Formulación dual basada en núcleos

Los núcleos se usan para proyectar los datos desde el espacio de entrada $\mathcal{X} \subseteq \mathbb{R}^d$ a un espacio de características de mayor dimensión, en el cual se pueden explotar propiedades más abstractas de los datos. Así, al estar proyectando los datos al espacio de características mediante una proyección no lineal, se pueden detectar relaciones no lineales entre los datos mediante una función lineal en dicho espacio de características (ver [31, 58, 100]).

Definición 0.1 *Sea $\phi : \mathcal{X} \subseteq \mathbb{R}^d \to \mathcal{F}$ una proyección (generalmente no lineal) desde el espacio de entrada $\mathcal{X}$ hacia el espacio de características $\mathcal{F}$. Un núcleo K es una función $K : \mathcal{X} \times \mathcal{X} \subseteq \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, tal que para cada $x, y \in \mathcal{X}$,*

$$K(x, y) = \phi(x)^\top \phi(y).$$

En realidad, no necesitaremos la expresión explícita de la proyección ϕ, sino sólo un algoritmo para evaluar el núcleo K en cada par de valores x e y.

Para introducir una estructura de núcleo en el Problema (16), basta con cambiar $x_i^\top x_l$ en la función objetivo por una estructura de núcleo más general evaluada en dicho par de valores $K(x_i, x_l)$.

$$\begin{aligned} \max_{\lambda} \quad & \sum_{i\in I}\lambda_i - \frac{1}{2}\sum_{i,l\in I}\lambda_i\lambda_l y_i y_l K(x_i,x_l) \\ & \sum_{i\in I}\lambda_i y_i = 0 \\ & 0 \le \lambda_i \le C,\ \forall i \in I. \end{aligned} \tag{27}$$

La expresión (13) se convierte ahora en

$$\omega = \sum_{i\in I}\lambda_i y_i \phi(x_i),\ j = 1,\ldots,d.$$

que no se puede evaluar, excepto si tenemos una representación explícita de ϕ.

Sin embargo, dado un nuevo elemento, se puede predecir la clase a la que éste pertenece, ya que el núcleo sí es conocido. En efecto, la función $f(x) = \omega^\top x + \beta$ se transforma ahora en

$$f(x) = \omega^\top x + \beta = \sum_{i\in I}\lambda_i y_i \phi(x_i)^\top \phi(x) + \beta = \sum_{i\in I}\lambda_i y_i K(x_i,x) + \beta.$$

Para obtener β, si $\exists i$ tal que $0 < \lambda_i < C$, de (20), se tiene que

$$\beta = y_i - \sum_{l\in I}\lambda_l y_l \phi(x_l)^\top \phi(x_i) = y_i - \sum_{l\in I}\lambda_l y_l K(x_l,x_i).$$

Cuando $\lambda_i \in \{0, C\}$, $\forall i \in I$, se puede escoger cualquier β tal que

$$\begin{aligned} \max_{\{i\in G_{+1}:\lambda_i=0\}}\{1 - \sum_{l\in I}\lambda_l y_l K(x_l,x_i)\} \quad &\le \beta \le \quad \min_{\{i\in G_{-1}:\lambda_i=0\}}\{-1 - \sum_{l\in I}\lambda_l y_l K(x_l,x_i)\} \\ \max_{\{i\in G_{-1}:\lambda_i=C\}}\{-1 - \sum_{l\in I}\lambda_l y_l K(x_l,x_i)\} \quad &\le \beta \le \quad \min_{\{i\in G_{+1}:\lambda_i=C\}}\{1 - \sum_{l\in I}\lambda_l y_l K(x_l,x_i)\}. \end{aligned}$$

Por último, dado un nuevo elemento, podemos predecir su etiqueta en base a

$$x \hookrightarrow sign(f(x)) := sign(\sum_{i\in I}\lambda_i y_i K(x_i,x) + \beta), \tag{28}$$

donde la única diferencia es que el valor $\omega^\top x$ se ha calculado usando para ello la expresión del núcleo, evitando así usar la expresión de la proyección ϕ.

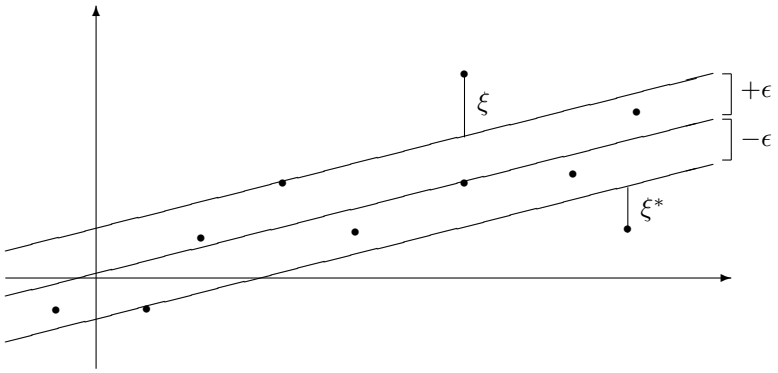

Figure 3: ϵ-Regresión de Vectores Soporte

0.2 Regresión de Vectores Soporte

0.2.1 Formulación del problema

En el modelo estándar de ϵ-Regresión de Vectores Soporte, (ver e.g. [31, 42, 52, 102, 113, 114]), se tiene una base de datos $\Omega \subseteq \mathbb{R}^d \times \mathbb{R}$, con elementos $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, donde x_i es el conjunto de variables predictoras e y_i es la variable dependiente, cuyo valor se predice usando el valor de x_i.

El objetivo de ϵ-Regresión de Vectores Soporte es encontrar $\omega \in \mathbb{R}^d$ y $\beta \in \mathbb{R}$ tales que, para cada ejemplo $i \in \Omega$, la función afín $f(x) = \omega^\top x + \beta$ produzca una predicción $f(x_i)$ que no se desvíe demasiado (lo máximo que puede desviarse es una cierta cantidad ϵ) respecto del verdadero valor observado y_i.

Dada una muestra de aprendizaje $I = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, extraída de la base de datos Ω, vamos a formular el problema de optimización que hay que resolver para obtener los parámetros óptimos ω y β para la regresión.

Como la desviación entre y_i y $f(x_i)$ debe ser como mucho ϵ, se obtiene el siguiente conjunto de restricciones

$$|\omega^\top x_i + \beta - y_i| \leq \epsilon, \ \forall i \in I.$$

El problema de optimización a resolver, como se indica en [102], es el siguiente,

$$\begin{aligned} \min_{\omega,\beta} \quad & \frac{1}{2}\sum_{j=1}^{d} \omega_j^2 \\ & y_i - \omega^\top x_i - \beta \leq \epsilon, \ \forall i \in I \\ & \omega^\top x_i + \beta - y_i \leq \epsilon, \ \forall i \in I. \end{aligned} \tag{29}$$

Este problema de optimización puede ser no factible. Por tanto, se deben introducir unas variables de holgura ξ, ξ^* en las restricciones (como se hizo en la Subsección 0.1.3,

para el caso de Margen Débil en SVMs) y hay que añadir un término de penalización a la función objetivo. El problema de optimización resultante tiene la siguiente forma (ver e.g. [102, 113]),

$$\begin{aligned} \min_{\omega,\beta,\xi,\xi^*} \quad & \frac{1}{2}\sum_{j=1}^{d}\omega_j^2 + C\sum_{i\in I}(\xi_i + \xi_i^*) \\ & y_i - \omega^\top x_i - \beta \leq \epsilon + \xi_i,\ \forall i \in I \\ & \omega^\top x_i + \beta - y_i \leq \epsilon + \xi_i^*,\ \forall i \in I \\ & \xi_i, \xi_i^* \geq 0,\ \forall i \in I, \end{aligned} \tag{30}$$

con C y ϵ las constantes del modelo, donde ϵ representa la máxima desviación permitida para los ejemplos de la muestra de aprendizaje y C representa el equilibrio entre la pendiente de la función de predicción y la suma de las desviaciones mayores que ϵ.

La Figura 3 explica gráficamente el modelo. Buscamos un hiperplano que ajuste los puntos del conjunto de datos, pero sólo se penalizarán aquellos puntos cuya desviación respecto del valor predicho (el punto correspondiente que se encuentra sobre el hiperplano) sea mayor que ϵ. Es decir, los puntos que se encuentran fuera de la banda definida por el hiperplano y el parametro ϵ, denominado *tubo ϵ-insensible*, son penalizados mediante la correspondiente variable de holgura (variable ξ para los puntos por encima del tubo, y ξ^* para los puntos por debajo del tubo).

La formulación (30), introducida por [113], corresponde al caso en que se trabaja con la llamada *función de pérdida ϵ-insensible*, definida como

$$|\xi|_\epsilon = \begin{cases} 0 & \text{if } |\xi| \leq \epsilon, \\ |\xi| - \epsilon & \text{en caso contrario.} \end{cases}$$

Una función de pérdida es una función de coste que penaliza los errores en la tarea de predicción. En particular, con la función de pérdida ϵ-insensible, se permiten aquellas desviaciones menores que la cantidad fijada ϵ, mientras que aquellas desviaciones mayores se penalizan linealmente. La Figura 4 muestra la forma de esta función de pérdida.

Otras funciones de pérdida, como la función Gaussiana o la función de Huber (con penalizaciones cuadráticas de los errores) o la función Laplaciana (que se puede ver como un caso particular de la función de pérdida ϵ-insensible, para $\epsilon = 0$) también han sido usadas en la literatura de este tema (ver [52, 102]).

Sin embargo, la ventaja de la función de pérdida ϵ-insensible es la dispersión de los vectores soporte, porque para esta función no todos los puntos serán vectores soporte, al contrario de lo que pasa con otras funciones (ver e.g. [52]).

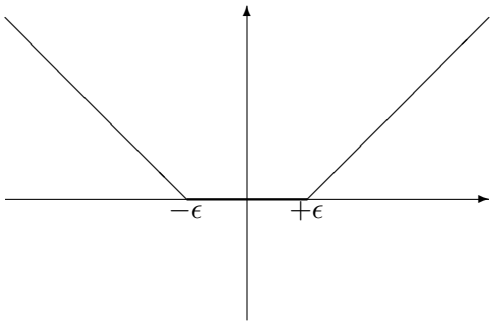

Figure 4: Función de pérdida ϵ-insensible

0.2.2 Formulación dual

A continuación, presentamos la formulación dual del Problema (30) (ver e.g. [31, 102]). Esta formulación dual también se puede usar para obtener una solución óptima de nuestro problema y nos permitirá manejar funciones no lineales en tareas de regresión, mediante la introducción de núcleos.

En primer lugar, introducimos multiplicadores no negativos de Lagrange para cada restricción del Problema (30) y obtenemos la siguiente función Lagrangiana,

$$\begin{aligned} L(\omega, \beta, \xi, \xi^*) &= \frac{1}{2}\sum_{j=1}^{d}\omega_j^2 + C\sum_{i\in I}(\xi_i + \xi_i^*) + \sum_{i\in I}\lambda_i(y_i - \sum_{j=1}^{d}\omega_j x_{ij} - \beta - \epsilon - \xi_i) \\ &\quad + \sum_{i\in I}\lambda_i^*(\sum_{j=1}^{d}\omega_j x_{ij} + \beta - y_i - \epsilon - \xi_i^*) - \sum_{i\in I}(\mu_i\xi_i + \mu_i^*\xi_i^*). \end{aligned}$$

Igualamos a cero las derivadas parciales de la función Lagrangiana, obteniendo

$$\frac{\partial L}{\partial \omega_j} = \omega_j - \sum_{i\in I}\lambda_i x_{ij} + \sum_{i\in I}\lambda_i^* x_{ij} = 0,\ j = 1, \ldots, d \tag{31}$$

$$\frac{\partial L}{\partial \beta} = -\sum_{i\in I}\lambda_i + \sum_{i\in I}\lambda_i^* = 0 \tag{32}$$

$$\frac{\partial L}{\partial \xi_i} = C - \lambda_i - \mu_i = 0,\ \forall i \in I \tag{33}$$

$$\frac{\partial L}{\partial \xi_i^*} = C - \lambda_i^* - \mu_i^* = 0,\ \forall i \in I. \tag{34}$$

De la restricción (31), obtenemos una expresión para calcular el vector ω como una combinación lineal de los multiplicadores Lagrangianos,

$$\omega_j = \sum_{i\in I}(\lambda_i - \lambda_i^*)x_{ij},\ j = 1, \ldots, d. \tag{35}$$

La restricción (32) se incluye en el problema como,

$$\sum_{i\in I}\lambda_i = \sum_{i\in I}\lambda_i^*, \tag{36}$$

y, como los multiplicadores μ_i, μ_i^* son no negativos, las restricciones (33)-(34) indican que los multiplicadores λ_i, λ_i^* están acotados por el parámetro C,

$$0 \le \lambda_i \le C, \quad \forall i \in I \tag{37}$$

$$0 \le \lambda_i^* \le C, \quad \forall i \in I. \tag{38}$$

Usando las restricciones obtenidas, la función objetivo para el problema dual queda como sigue,

$$\begin{aligned}\tilde{L}(\lambda,\lambda^*) &= -\frac{1}{2}\sum_{j=1}^{d}\left(\sum_{i\in I}(\lambda_i-\lambda_i^*)x_{ij}\right)\left(\sum_{l\in I}(\lambda_l-\lambda_l^*)x_{lj}\right)\\ &\quad -\epsilon\sum_{i\in I}(\lambda_i+\lambda_i^*)+\sum_{i\in I}(\lambda_i-\lambda_i^*)y_i\\ &= -\frac{1}{2}\sum_{i,l\in I}(\lambda_i-\lambda_i^*)(\lambda_l-\lambda_l^*)x_i^\top x_l-\epsilon\sum_{i\in I}(\lambda_i+\lambda_i^*)+\sum_{i\in I}(\lambda_i-\lambda_i^*)y_i,\end{aligned}$$

y añadiéndole las restricciones (36)-(38), la formulación dual es

$$\begin{aligned}\max_{\lambda,\lambda^*}\quad & -\frac{1}{2}\sum_{i,l\in I}(\lambda_i-\lambda_i^*)(\lambda_l-\lambda_l^*)x_i^\top x_l-\epsilon\sum_{i\in I}(\lambda_i+\lambda_i^*)+\sum_{i\in I}(\lambda_i-\lambda_i^*)y_i\\ & \sum_{i\in I}(\lambda_i-\lambda_i^*)=0\\ & 0\le\lambda_i,\lambda_i^*\le C,\quad \forall i\in I,\end{aligned} \tag{39}$$

que es un problema de maximización cuadrático convexo en las variables λ, λ^*.

Dada una solución óptima (λ,λ^*) del Problema (39), podemos recuperar una solución óptima del Problema (30). Primero, usamos (35) para calcular ω. Para las variables β, ξ y ξ^*, usamos que se tienen que verificar las siguientes condiciones de Karush-Kuhn-Tucker,

$$\lambda_i\cdot(y_i-\sum_{j=1}^{d}\omega_j x_{ij}-\beta-\epsilon-\xi_i)=0,\quad \forall i\in I \tag{40}$$

$$\lambda_i^*\cdot(\sum_{j=1}^{d}\omega_j x_{ij}+\beta-y_i-\epsilon-\xi_i^*)=0,\quad \forall i\in I \tag{41}$$

$$\xi_i\cdot(C-\lambda_i)=0,\quad \forall i\in I \tag{42}$$

$$\xi_i^*\cdot(C-\lambda_i^*)=0,\quad \forall i\in I \tag{43}$$

$$0\le\lambda_i,\lambda_i^*\le C,\quad \forall i\in I. \tag{44}$$

Si $\exists i : 0 < \lambda_i < C$, la expresión (42) implica que $\xi_i = 0$, y de (40), recuperamos el valor de β como

$$\beta = y_i - \sum_{j=1}^{d} \omega_j x_{ij} - \epsilon. \tag{45}$$

Si $\exists i : 0 < \lambda_i^* < C$, la expresión (43) implica que $\xi_i^* = 0$ y se puede obtener el valor de β de (41) como

$$\beta = y_i - \sum_{j=1}^{d} \omega_j x_{ij} + \epsilon. \tag{46}$$

Si cada λ_i y λ_i^* están en $\{0, C\}$, cuando $\lambda_i = 0$, por (42), se tiene que $\xi_i = 0$, y dado que $(\omega, \beta, \xi, \xi^*)$ es factible para (30),

$$y_i - \sum_{j=1}^{d} \omega_j x_{ij} - \beta - \epsilon \leq 0, \text{ i.e., } \beta \geq y_i - \sum_{j=1}^{d} \omega_j x_{ij} - \epsilon. \tag{47}$$

Si $\lambda_i = C$, la correspondiente variable de holgura puede ser positiva y la restricción asociada en (40) se vuelve activa,

$$y_i - \sum_{j=1}^{d} \omega_j x_{ij} - \beta - \epsilon - \xi_i = 0, \text{ i.e., } \beta \leq y_i - \sum_{j=1}^{d} \omega_j x_{ij} - \epsilon. \tag{48}$$

Con un razonamiento similar con los multiplicadores λ_i^*, se obtiene que

$$\text{si } \lambda_i^* = 0, \quad \text{entonces } \beta \leq y_i - \sum_{j=1}^{d} \omega_j x_{ij} + \epsilon \tag{49}$$

$$\text{si } \lambda_i^* = C, \quad \text{entonces } \beta \geq y_i - \sum_{j=1}^{d} \omega_j x_{ij} + \epsilon. \tag{50}$$

Si tomamos máximo en las expresiones (47) y (50), y tomamos mínimo en las expresiones (48) y (49), β debe verificar las siguientes restricciones,

$$\begin{array}{lll} \max\limits_{\{i \in I : \lambda_i = 0\}} \{y_i - \sum\limits_{j=1}^{d} \omega_j x_{ij} - \epsilon\} & \leq \beta \leq & \min\limits_{\{i \in I : \lambda_i^* = 0\}} \{y_i - \sum\limits_{j=1}^{d} \omega_j x_{ij} + \epsilon\} \\ \max\limits_{\{i \in I : \lambda_i^* = C\}} \{y_i - \sum\limits_{j=1}^{d} \omega_j x_{ij} + \epsilon\} & \leq \beta \leq & \min\limits_{\{i \in I : \lambda_i = C\}} \{y_i - \sum\limits_{j=1}^{d} \omega_j x_{ij} - \epsilon\}, \end{array} \tag{51}$$

y por tanto, β pertenece a estos intervalos. De hecho, igual que se mostró para el caso de SVMs en la Subsección 0.1.4, se tiene que, cuando $\lambda_i, \lambda_i^* \in \{0, C\}$, $\forall i \in I$, se puede escoger cualquier β verificando (51) como solución óptima del Problema (30).

De las condiciones KKT (40)-(44), se observa que los ejemplos con multiplicador asociado $\lambda_i = C$ (respectivamente, $\lambda_i^* = C$) se encuentran fuera del tubo ϵ-insensible, mientras que los ejemplos con ambos multiplicadores iguales a cero están incluidos estrictamente en el tubo ϵ-insensible. Los ejemplos cuyo multiplicador asociado verifica $0 < \lambda_i < C$ ó $0 < \lambda_i^* < C$ se encuentran en la frontera del tubo.

Los ejemplos con un multiplicador estrictamente positivo se llaman *vectores soporte*, y son necesarios para calcular la función de predicción. De hecho, si borráramos de la muestra de aprendizaje aquellos ejemplos con $\lambda_i = \lambda_i^* = 0$, la función de predicción resultante sería la misma.

Por último, al menos uno de los multiplicadores tiene que ser cero, es decir,

$$\lambda_i \cdot \lambda_i^* = 0, \ \forall i \in I,$$

ya que un mismo punto no puede ser a la vez un vector soporte para ambos lados del tubo ϵ-insensible, con $\epsilon > 0$.

0.2.3 Formulación dual basada en núcleos

A continuación, construiremos la formulación dual para el Problema (30) usando una estructura de núcleo. Para ello, basta con remplazar $x_i^\top x_l$ en la función objetivo del Problema (39) por otra estructura de núcleo $K(x_i, x_l)$, y obtendremos la siguiente formulación dual basada en núcleos,

$$\begin{array}{ll} \max\limits_{\lambda,\lambda^*} & -\frac{1}{2}\sum\limits_{i,l\in I}(\lambda_i - \lambda_i^*)(\lambda_l - \lambda_l^*)K(x_i, x_l) - \epsilon\sum\limits_{i\in I}(\lambda_i + \lambda_i^*) + \sum\limits_{i\in I}(\lambda_i - \lambda_i^*)y_i \\ & \sum\limits_{i\in I}(\lambda_i - \lambda_i^*) = 0 \\ & 0 \leq \lambda_i, \lambda_i^* \leq C, \quad \forall i \in I. \end{array} \tag{52}$$

La expresión para ω, obtenida de (35), no se puede usar explícitamente, ya que la expresión de la proyección ϕ es desconocida,

$$\omega = \sum_{i\in I}(\lambda_i - \lambda_i^*)\phi(x_i).$$

Sin embargo, no la necesitaremos para obtener la predicción para cualquier nuevo elemento x.

Para obtener la expresión de β, si $\exists i : 0 < \lambda_i < C$ ó $0 < \lambda_i^* < C$, obtenemos la expresión (53) ó (54), respectivamente,

$$\beta = y_i - \sum_{l\in I}(\lambda_l - \lambda_l^*)K(x_l, x_i) - \epsilon. \tag{53}$$

$$\beta = y_i - \sum_{l\in I}(\lambda_l - \lambda_l^*)K(x_l, x_i) + \epsilon. \tag{54}$$

Cuando cada $\lambda_i, \lambda_i^* \in \{0, C\}$, podemos escoger cualquier β que verifique las siguientes expresiones,

$$\max_{\{i\in I:\lambda_i=0\}}\{y_i - \sum_{l\in I}(\lambda_l - \lambda_l^*)K(x_l, x_i) - \epsilon\} \leq \beta \leq \min_{\{i\in I:\lambda_i^*=0\}}\{y_i - \sum_{l\in I}(\lambda_l - \lambda_l^*)K(x_l, x_i) + \epsilon\}$$
$$\max_{\{i\in I:\lambda_i^*=C\}}\{y_i - \sum_{l\in I}(\lambda_l - \lambda_l^*)K(x_l, x_i) + \epsilon\} \leq \beta \leq \min_{\{i\in I:\lambda_i=C\}}\{y_i - \sum_{l\in I}(\lambda_l - \lambda_l^*)K(x_l, x_i) - \epsilon\}.$$

Entonces, dado un nuevo elemento x, su predicción, usando una estructura de núcleo, es la siguiente,

$$f(x) = \sum_{l\in I}(\lambda_l - \lambda_l^*)K(x_l, x) + \beta. \tag{55}$$

0.3 Clasificación y regresión con datos imprecisos

0.3.1 Intervalos y datos imprecisos

En determinadas situaciones, los datos no se pueden expresar mediante vectores de características puntuales, sino que debemos introducir datos intervalares. Así, usaremos intervalos para expresar rangos, como por ejemplo el rango de temperatura durante un día, los intervalos de edad para un grupo de personas o el coste de determinados productos estudiado en el conjunto de establecimientos de una ciudad (ya que se encontrarán variaciones en el precio entre las distintas tiendas). También usaremos intervalos cuando se le han tomado varias medidas de una misma variable a un mismo individuo, y se pretende resumir dichas medidas, por ejemplo, las fluctuaciones en la presión sanguínea o en el pulso de un paciente, o el peso de un recién nacido durante su primera semana de vida.

Los intervalos también aparecen de manera natural en el caso de datos imprecisos, o cuando realizamos una estimación de un determinado parámetro mediante un intervalo de confianza, y, en general, cada vez que tengamos incertidumbre o vaguedad en nuestro problema.

Otro caso de datos intervalares lo encontramos en el ámbito del Análisis de Datos Simbólicos ([11, 13]), cuando tenemos que resumir grandes bases de datos de manera que el conjunto de datos resultante tenga un tamaño más manejable y retenga información suficiente respecto de la base de datos original. Existen distintas propuestas para hacer esta agregación de datos, como el uso de variables clásicas (valores puntuales), variables con múltiples valores (variables nominales que pueden tomar varios resultados), variables intervalares (los datos son agregados en intervalos, éste es el caso de nuestro interés) o variables modales (una variable puntual, intervalar o nominal que puede tomar distintos variables con una probabilidad asociada).

0.3.2 Clasificación con datos imprecisos: casos de interés y literatura

Consideremos un problema de clasificación supervisada en el cual los elementos del conjunto de datos no son puntos, sino conjuntos en $\mathbb{R}^d$, tales conjuntos representando algún tipo de datos imprecisos. Tenemos que asignar una etiqueta (+1 ó -1, en el caso de dos clases) al conjunto completo, según el comportamiento de la mayoría de sus elementos con respecto a cierta regla de clasificación.

Hay tres casos de interés que pueden ser abordados con este modelo, que son los siguientes: datos intervalares, datos afectados por algún tipo de ruido o perturbación y datos con valores perdidos.

- **Datos intervalares:** En particular, un caso de interés que puede ser modelado mediante esta metodología se tiene cuando los elementos X_i a clasificar vienen definidos como un producto Cartesiano de intervalos, $X_i = \prod_{j=1}^{d}[l_{ij}, u_{ij}] \subset \mathbb{R}^d$, con $l_{ij} \leq u_{ij}$, es decir, l_{ij} y u_{ij} representan, respectivamente, las cotas superior e inferior de cada coordenada, pudiendo coincidir en algunos casos.

  La clasificación con datos intervalares ha sido objeto de estudio en la literatura, desde distintos enfoques. Recientemente, en [3], se ha propuesto una formulación con SVMs para el problema de clasificación con intervalos, en un trabajo independiente al nuestro, sólo para el caso linealmente separable. En [43], se aplica análisis discriminante lineal a este tipo de problemas de clasificación considerando tres técnicas diferentes: asignando una distribución uniforme a cada intervalo, expandiendo el conjunto de datos al correspondiente conjunto de vértices y describiendo cada intervalo mediante su centro y su rango. En [39], se construye un núcleo de Función de Base Radial mediante la distancia de Hausdorff entre intervalos, y se aplica a la clasificación de intervalos. Otro núcleo para intervalos, basado en la operación intersección, es descrito en [96].

  Otras técnicas, que usan Redes Neuronales, son descritas en [95, 101]. Dos métodos que permiten el uso de datos intervalares como entradas para un perceptrón con múltiples capas se incluyen en [95] (uno de ellos está basado en una descripción de los intervalos como valores puntuales y el otro está basado en una comprensión probabilística de los intervalos). En [101], se crea un sistema experto neuronal para diagnóstico, donde la base de conocimiento es una Red Neuronal que se construye automáticamente a través de un algoritmo de aprendizaje. Así mismo, en [90] se describen métodos de clasificación para datos simbólicos.

- **Datos perturbados:** Otro caso de interés, que puede ser modelado bajo nuestro enfoque, se tiene cuando los datos están afectados por algún tipo de ruido o perturbación. En ese caso, hay que construir un clasificador robusto, insensible

al ruido en los datos. Un modelo de Máquinas de Vectores Soporte Robustas ha sido estudiado en [108, 109], donde hay que resolver un problema de optimización mediante Programación Cónica de Segundo Orden.

Otra propuesta diferente para clasificación robusta se puede encontrar en [46], donde se formula un problema de clasificación binaria en el que los datos son desconocidos, pero están acotados por hiperrectángulos. En ese trabajo, los autores diseñan un clasificador robusto minimizando el valor para el peor caso de una función de pérdida dada. Se consideran tres funciones diferentes, incluyendo la pérdida lineal de Hinge (ver [31]) para SVMs, que proporciona una cota superior sobre el número esperado de futuros errores de clasificación incorrecta. En [68], se formula otro problema de clasificación en el que los datos vienen dados por la media y covarianza de cada clase, que se suponen conocidos. El objetivo es minimizar la probabilidad (máxima) en el peor caso de tener futuros puntos mal clasificados, usando para ello técnicas de Programación Cónica de Segundo Orden. Geométricamente, se corresponde con el problema de minimizar el máximo de las distancias de Mahalanobis entre las dos clases. Siguiendo una estrategia similar, en [8] los valores que puede tomar un dato se describen mediante un conjunto de incertidumbre, definido a su vez por un elipsoide acotado cuyos parámetros son su localización (valor esperado o centro del elipsoide) y su forma (matriz de covarianza o matriz de la longitud de los ejes al cuadrado).

- **Valores perdidos:** Otra situación que se puede incluir dentro del ámbito de la clasificación con datos imprecisos es el caso en el que se tienen valores perdidos (ver [73] para un estudio sobre el análisis estadístico en bases de datos con valores perdidos), es decir, cuando la base de datos está formada por vectores de características pero algunas de sus coordenadas no aparecen en el conjunto de datos. En la literatura, se muestran distintas técnicas que se pueden usar para manejar datos perdidos en problemas de clasificación (ver [74, 75] para una visión general sobre el tema). Aunque la técnica más popular es la imputación mediante valores puntuales (usando para ello el resto de valores de la base de datos) para remplazar las coordenadas perdidas, la imputación mediante intervalos nos permitirá estudiar este problema con las técnicas desarrolladas para clasificación con datos intervalares.

Nuestro objetivo con el modelo que proponemos en esta tesis para clasificación con datos imprecisos es formular un modelo más general, que permita estudiar los casos de datos intervalares, datos con perturbación y valores perdidos como casos particulares de este modelo, y que consiga mejorar los resultados obtenidos en bases de datos reales por otros trabajos previos de la literatura. Además, también consideraremos la extensión al problema multi-clase.

0.3.3 Regresión con datos imprecisos: casos de interés y literatura

Consideremos un problema de regresión en el cual los elementos de la base de datos no son puntos, sino conjuntos en $\mathbb{R}^d$, y la variable dependiente no toma un valor puntual, sino todo un intervalo. Es decir, tanto la variable dependiente como las predictoras están afectadas por algún tipo de imprecisión. Dado un nuevo elemento, un conjunto en $\mathbb{R}^d$, hay que asignarle una salida intervalar, teniendo en cuenta para ello al conjunto completo.

Como en el caso de clasificación, distintos casos de interés se pueden incluir dentro de esta descripción del modelo: datos intervalares (datos con entrada intervalar y salida intervalar), datos con entrada puntual y salida intervalar (hay imprecisión sólo en la variable dependiente), datos con algún tipo de perturbación y valores perdidos.

- **Datos intervalares:** Como en el problema de clasificación, un caso de interés que se puede abordar con esta metodología es cuando cada elemento X_i de la base de datos es un producto Cartesiano, $X_i = \prod_{j=1}^{d}[l_{ij}, u_{ij}] \subset \mathbb{R}^d$, con $l_{ij} \leq u_{ij}$, l_{ij} y u_{ij} representando, respectivamente, las cotas inferior y superior de cada coordenada. Además, la variable dependiente también viene dada por un intervalo $Y_i = [\tilde{l}_i, \tilde{u}_i]$, con $\tilde{l}_i \leq \tilde{u}_i$.

  En la literatura, se ha estudiado el caso de regresión para datos con entrada intervalar y salida intervalar, bajo distintos enfoques. Desde la perspectiva del Análisis de Datos Simbólicos, el primer trabajo que se publicó en el tema fue [9]. Consideremos el modelo de regresión lineal clásico (ver e.g. [41]),

  $$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_d x_{id} + \varepsilon_i,\ i = 1, \ldots, n,$$

  donde y_i es la variable dependiente, $x_i = (x_{i1}, \ldots, x_{id})^\top \in \mathbb{R}^d$ es el vector de las variables predictoras, $\beta_0, \beta_1, \ldots, \beta_d$ son los coeficientes del modelo de regresión y ε_i es el error. La propuesta de [9] consiste en ajustar el modelo de regresión lineal clásico sobre el punto medio de los intervalos de cada variable del conjunto de datos. Las predicciones para las cotas superior e inferior para la variable dependiente se calculan sobre el modelo obtenido. Este modelo se mejora en [34], en donde se usan dos modelos de regresión lineal, uno para predecir el punto medio de la salida y el otro para predecir el rango. Las predicciones para las cotas superior e inferior para la variable dependiente son recuperadas con el punto medio y el rango. En [70], se muestra una comparación entre ambos modelos. Otras extensiones de estos modelos se pueden encontrar en [10, 71].

  Relacionado con este problema, encontramos también el concepto de análisis de regresión difuso, en el cual se han desarrollado distintas propuestas. Éstas se

pueden clasificar en dos grupos principales: el enfoque posibilístico (propuesto inicialmente por [105]), en el cual la función objetivo a minimizar es una medida de la dispersión de las predicciones, y el enfoque de mínimos cuadrados (introducidos en [24, 36]), en el cual se minimiza una distancia, sobre números dispersos, entre la salida real y la predicha.

Las SVMs también se han aplicado a los modelos de regresión lineal múltiple difusa (ver [60, 61]). En estos trabajos, dos modelos distintos han sido estudiados: cuando las variables predictoras y dependiente son números difusos triangulares simétricos (entrada difusa y salida difusa) y cuando las variables predictoras son precisas y la variable dependiente es un número difuso triangular (entrada precisa y salida difusa). La metodología estándar de ϵ-SVR se aplica imponiendo que la moda y los extremos de los intervalos deben verificar las restricciones habituales. En el caso de entrada precisa y salida difusa, se introducen regresores no lineales mediante métodos de núcleo.

- **Entrada puntual y salida imprecisa:** La situación en la cual las variables predictoras son puntuales y sólo la variable dependiente es intervalar se puede ver como un caso particular del caso anterior (datos intervalares). Sin embargo, este modelo merece un estudio más profundo para poder así introducir el uso de funciones no lineales en tareas de regresión mediante estructuras de núcleo.

  En la bibliografía, nos encontramos en primer lugar un problema relacionado que maneja el concepto de análisis de regresión intervalar, que es la versión más simple del análisis de regresión posibilística, y que fue introducido por Tanaka et al. (ver [69, 103, 104]). Dada una base de datos con entrada y salida precisa, el objetivo del análisis de regresión intervalar es predecir la variable dependiente con un intervalo, usando las variables predictoras. Para ello, los coeficientes del modelo usados para la regresión son también intervalos. Cada coeficiente viene expresado por su centro y su radio.

  En el modelo original, se da una formulación de programación lineal para resolver el problema, en la cual el objetivo es minimizar la suma de los radios de las salidas (predicciones), con la restricción de que el valor real de la variable dependiente debe estar incluida en la predicción (ver [103]). Posteriormente, en [104], se da una formulación como un problema cuadrático que incluye en la función objetivo un término para minimizar la suma de las distancias al cuadrado desde el centro de la predicción intervalar al valor real de la variable dependiente.

  Otras mejoras han sido realizadas para estudiar el papel de los valores atípicos en el proceso de regresión. En [69], se construyen dos modelos de regresión para cada base de datos usando técnicas de cuantiles, dando dos salidas intervalares para cada observación, con la más pequeña incluida en la mayor. El primer modelo se construye con una proporción dada de los datos (así, se puede estudiar

el comportamiento general de los datos, sin contener valores atípicos), mientras que el segundo modelo se construye con todas las observaciones. Entonces, dada una base de datos, se asignan dos intervalos como predicción, y dicha predicción puede verse como una salida difusa trapezoidal.

Las SVMs se han aplicado al problema descrito en [69] para construir los dos modelos (ver [63]) y también al caso general de análisis de regresión intervalar. En [65], se resuelve un modelo de ϵ-SVR (con $\epsilon = 0$) para obtener un valor preciso inicial de la salida, que será el centro de una salida intervalar con radio igual a un valor ϵ, calculado usando los errores de regresión obtenidos. Esta salida intervalar se da como semilla inicial a dos redes de Función de Base Radial que proporcionan las cotas superior e inferior de la salida. En [64], la formulación cuadrática de [104] se integra con el modelo estándar de ϵ-SVR.

- **Datos perturbados:** Otra situación interesante relacionada con el modelo de regresión con imprecisión es el caso de datos afectados por algún tipo de ruido o perturbación. En tal caso, hay que construir un regresor robusto, que sea insensible a este ruido o perturbación. En [107, 108], se estudia un modelo de Regresión de Vectores Soporte Robusta, con ruido en la entrada (variables predictoras). Aunque se supone que los puntos están afectados por incertidumbre o ruido, dicha perturbación se considera acotada por una hiperesfera de radio conocido. El modelo se formula como un problema de optimización y se resuelve con Programación Cónica de Segundo Orden.

- **Valores perdidos:** El caso en el que la base de datos tiene valores perdidos también se puede incluir dentro de esta metodología. En vez de hacer imputación con valores puntuales, como en el caso habitual, para las coordenadas perdidas, éstas pueden ser remplazadas mediante intervalos que se construyen con los valores no perdidos del conjunto de datos.

Como en el caso de clasificación, nuestro modelo generaliza los problemas de regresión para los casos de datos imprecisos descritos anteriormente. La Regresión de Vectores Soporte aplicada al caso de datos intervalares consigue mejorar los resultados en los conjuntos de datos de referencia que han sido usados en algunos trabajos previos. Además, conseguimos explotar la estructura del problema en el caso de entrada puntual y salida imprecisa, donde introducimos regresores no lineales mediante el uso de núcleos.

0.4 Aprendizaje con Múltiples Ejemplos

0.4.1 El Problema con Múltiples Ejemplos

Consideremos un problema de clasificación en el cual los elementos que componen el conjunto de datos pueden estar representados por varios vectores de características (a este conjunto de vectores se le llama *bolsa*), y sólo algunos de ellos (incluso solamente uno en ciertos casos), son los responsables de la clase a la que pertenece el elemento. Para abordar esta tarea, un conjunto de herramientas agrupadas bajo el nombre de Aprendizaje con Múltiples Ejemplos se ha desarrollado durante la última década.

Un Problema con Múltiples Ejemplos (en inglés, Multi-Instance Problem) es un problema de clasificación supervisada en el cual los elementos a clasificar son bolsas de ejemplos que son vectores midiendo d atributos distintos (ver e.g. [38, 118, 121] para una descripción). Atendiendo a estas medidas, hay que asignar una etiqueta ($+1$ ó -1 en el caso de dos clases) a cada bolsa.

Aunque nos encontramos con diferentes maneras de asignar una etiqueta al conjunto completo de vectores (ver [118]) en nuestro problema, la estrategia más popular está basada en la denominada hipótesis MI (MI-assumption, ver [118]), que indica que una bolsa se considera positiva (etiqueta $+1$) si al menos uno de sus ejemplos verifica una cierta condición, y negativa (etiqueta -1) en caso contrario.

El Problema con Múltiples Ejemplos se puede ver como un caso especial del problema de clasificación con imprecisión definido en la Subsección 0.3.2. La diferencia con respecto al tipo de situación descrita en 0.3.2 se encuentra en la cardinalidad de los elementos de la base de datos. En ambos modelos, los elementos son conjuntos en $\mathbb{R}^d$, pero, mientras que en el Problema con Múltiples Ejemplos los elementos son conjuntos discretos (bolsas de ejemplos), en el modelo descrito en 0.3.2 (datos intervalares, datos perturbados) los elementos son conjuntos continuos en $\mathbb{R}^d$ (cajas o esferas). En el primer modelo, la cardinalidad de cada elemento es finita, y en el segundo, la cardinalidad es infinita.

0.4.2 Clasificación con Múltiples Ejemplos: origen del problema y literatura

El Aprendizaje con Múltiples Ejemplos tuvo su primera aplicación en el campo de predicción de actividad de drogas en el trabajo realizado por Dietterich et al. en [38]. En dicho problema, las bolsas son moléculas, mientras que los ejemplos son distintas conformaciones de baja energía, es decir, formas que la molécula puede adoptar mediante rotación de sus enlaces. Sólo algunas de estas conformaciones se pueden unir a un determinado lugar de unión (que forma parte de una molécula mayor), y tras

dicha unión, la molécula se vuelve activa, produciendo una determinada droga. Pero los bioquímicos sólo conocen si la molécula está o no cualificada para producir la droga, y no disponen de ninguna información sobre las conformaciones que producen tal actividad. El objetivo es, por tanto, dada una nueva molécula, ser capaz de predecir si será o no activa, teniendo en cuenta su conjunto completo de conformaciones.

En [38], se construye un hiperrectángulo tal que tiene que contener al menos un ejemplo de cada bolsa positiva y no puede contener ningún ejemplo de las bolsas negativas. Se proponen tres algoritmos. El primero construye el menor rectángulo que cubre todos los ejemplos de las bolsas positivas, y entonces excluye todos los ejemplos negativos, eliminando, por pasos, aquel ejemplo negativo que requiere eliminar un menor número de ejemplos positivos. El segundo algoritmo es una modificación del primero, que asigna un coste por eliminar cada ejemplo positivo (lo cual resulta útil, por ejemplo, para evitar excluir del rectángulo el último ejemplo que quedaba de una bolsa positiva). El tercer algoritmo construye el menor rectángulo que cubre al menos un ejemplo de cada bolsa positiva, usando un procedimiento de retroalimentación para escoger los ejemplos positivos y seleccionando las características más relevantes en cada paso.

Además de la predicción de actividad de drogas, el Aprendizaje con Múltiples Ejemplos se ha aplicado con éxito a otros campos diferentes. En clasificación o reconocimiento de imágenes (ver [2, 26, 27, 45, 77, 78, 89, 119]), se tiene una serie de imágenes, y el objetivo es entrenar un clasificador que detecte un determinado objeto. Las imágenes son las bolsas del problema, que están segmentadas en conjuntos de píxeles (blobs), que hacen el papel de ejemplos. Clasificamos una imagen como positiva si contiene al menos un determinado blob, característico del objeto a detectar, y como negativa en caso contrario.

En [122], se describe un problema de minería de web, en el cual el objetivo es recomendar sobre páginas web índice a un usuario, basándose en la navegación previa de dicho usuario. Las páginas índice (que son páginas web con enlaces a otras páginas, y que contienen sólo los títulos o breve información sobre el contenido de dichas páginas enlazadas) son las bolsas del problema, mientras que cada página enlazada es un ejemplo. Cada página enlazada está representada por sus d términos más frecuentes. Un usuario está interesado en una página índice si lo está en al menos uno de sus enlaces (verificando así la hipótesis MI).

Otras diferentes aplicaciones se han llevado a cabo en los campos del reconocimiento de escritura a mano (ver [67]), categorización de texto (ver [2]) o en predicción de enfermedades (ver [118]).

Desde la aparición de [38], varios autores han desarrollado nuevos algoritmos para intentar mejorar los resultados obtenidos por Dietterich et al., siguiendo diversas es-

trategias.

- **Densidad Diversa:** En [77], se introdujo el concepto de *Densidad Diversa* para tratar con este tipo de problemas. Un punto con alta Densidad Diversa está cerca de muchos ejemplos de diferentes bolsas positivas, y lejos de los ejemplos de las bolsas negativas. El problema es encontrar el punto con la máxima Densidad Diversa y, cuando tenemos una nueva bolsa, ésta es clasificada como positiva si la menor distancia de la bolsa a ese punto es menor que un determinado umbral, y como negativa en otro caso. En [120], se propone un algoritmo de Esperanza-Maximización para maximizar la Densidad Diversa, transformando el problema con múltiples ejemplos en uno con ejemplos únicos, y usando el algoritmo EM para maximizar la responsabilidad de cada ejemplo en la asignación de la correspondiente etiqueta de su bolsa. Esta misma estrategia se usa también en [89].

- **k-ésimo Vecino Más Cercano:** Dos variantes del algoritmo del k-ésimo Vecino Más Cercano (en inglés, k-Nearest Neighbour, k-NN) se proponen en [115] para resolver problemas con ejemplos múltiples. Se usan las distancias de Hausdorff minimal y maximal para medir la proximidad entre bolsas. Un enfoque bayesiano se usa en el denominado algoritmo k-NN Bayesiano. Para el algoritmo k-NN Citación, los conceptos de referencias y citadores se aplican a la metodología de este algoritmo, estudiando para cada bolsa no sólo sus vecinos, sino también aquellas bolsas que la ven a ella como un vecino. Estos conceptos se usan también en [122].

- **Máquinas de Vectores Soporte:** En [2], se proponen dos formulaciones como un problema cuadrático entero mixto. El enfoque *mi-SVM* introduce variables enteras para modelar las etiquetas individuales de cada ejemplo de las bolsas positivas, mientras que la formulación *MI-SVM* selecciona un representante de cada bolsa positiva como aquél que determina el signo de la etiqueta de su bolsa. En ambos algoritmos se calculan los hiperplanos óptimos, dada una asignación inicial para las variables enteras, y se actualizan esos valores enteros asignando una etiqueta positiva al ejemplo con el mayor valor para la regla de clasificación.

  En [76], se propone un hiperplano separador mediante SVMs, usando que una bolsa positiva está clasificada correctamente si al menos un elemento del cierre convexo de los ejemplos de la bolsa está incluido en el semiespacio positivo. Se obtiene un conjunto de restricciones bilineales y, por turnos, se mantiene constante un conjunto de variables y se resuelve el correspondiente problema lineal. Este algoritmo de linealización sucesiva converge en pocas iteraciones a un óptimo local.

  En [32], un problema de categorización de imágenes se resuelve mediante SVMs y el algoritmo k-medias. Dada una imagen, se representa como un conjunto

de regiones de imagen y, mediante el algoritmo k-medias, los descriptores de la región son asignados a un número predeterminado de conglomerados. Cada imagen viene descrita por un vector de características, que cuenta el número de regiones en cada cluster. El algoritmo *uno contra todos* para SVMs es usado para resolver el problema de asignar cada imagen a un conglomerado.

Las SVMs Transductivas, una modificación del algoritmo estándar que fuerza a los datos no etiquetados (que provienen de las bolsas positivas) a estar lo más lejos posible del hiperplano separador, se usa en [14], obteniéndose buenos resultados cuando las bolsas positivas están dispersas (pocos ejemplos positivos).

- **Núcleos:** Se han definido diferentes núcleos en este tipo de datos. Así, podemos encontrar un núcleo general en [50], que separa las bolsas positivas y negativas bajo hipótesis naturales. El procedimiento de núcleo que se describe en [67] consiste en proyectar la base de datos en un espacio de Hilbert mediante un primer núcleo, ajustar un modelo Gaussiano a cada bolsa en ese espacio de características y definir un segundo núcleo como la afinidad de Bhattacharyya entre esos modelos Gaussianos.

  En [26], se define una medida de similaridad entre una bolsa y un ejemplo, así como una proyección en términos de dicha medida, transformando el problema con múltiples ejemplos en uno con ejemplos únicos, que se resuelve con SVMs. Una estrategia similar se sigue en [27], pero para definir la proyección, se usa la máxima distancia entre una bolsa y un conjunto de ejemplos prototipos obtenidos mediante Densidad Diversa. Otros núcleos sobre conjuntos de vectores se pueden encontrar en [45].

Muchas otras técnicas diferentes han sido propuestas en la literatura. Entre otras, podemos citar Optimización DC [29], Aprendizaje Proposicional [48], Aprendizaje Generalizado con Múltiples Ejemplos [116], etc.

En esta tesis, se describe una estrategia diferente para resolver este tipo de problemas, por la cual la regla de clasificación viene definida en términos de una bola separadora que maximiza el margen entre los dos grupos a clasificar. Se propone un algoritmo basado en las condiciones necesarias de optimalidad del problema. Este tipo de soluciones resulta también útil para tratar con problemas de Teoría de Localización, en particular, en Localización Semi-Nociva. Así, nuestro objetivo es mostrar la aplicabilidad de este tipo de técnicas de clasificación en otro interesante área de la Investigación Operativa, como es el caso de la Teoría de Localización.

0.5 Resumen de la tesis

En esta tesis, desarrollamos varias herramientas para resolver problemas de clasificación y regresión donde los elementos del conjunto de datos no son vectores de características puntuales, sino conjuntos en $\mathbb{R}^d$ con ciertas propiedades geométricas. El clasificador o regresor se define siguiendo la estrategia, utilizada con éxito en Máquinas de Vectores Soporte, de maximizar el margen. En cada caso, se formula un problema de optimización, que hay que resolver para encontrar el clasificador o regresor óptimo.

Se obtienen varios tipos de problemas de optimización para estos problemas: programas convexos cuadráticos cuando buscamos hiperplanos para tareas de clasificación o regresión (cuya solución se obtendrá directamente usando un solucionador, como CPLEX o LOQO [112]) o programas no lineales y no lineales enteros mixtos cuando buscamos hiperesferas separadoras (donde desarrollaremos algoritmos exactos o heurísticos para obtener una solución óptima).

En el Capítulo 2, estudiamos el problema de clasificación supervisado con datos imprecisos, donde los elementos a clasificar son conjuntos con ciertas propiedades geométricas. En particular, este modelo se puede aplicar para tratar con datos afectados por algún tipo de ruido y en el caso de datos intervalares. Se definen dos reglas de clasificación, una difusa y otra precisa, en términos de un hiperplano separador, y se introduce una formulación del problema de identificación de la regla mediante maximización del margen, extendiendo así las técnicas estándares en SVMs para vectores de características puntuales. Estos resultados se muestran en nuestro artículo [16].

En el Capítulo 3, que está basado en nuestro trabajo [17], se considera el problema de regresión con imprecisión en los datos. Los elementos de la base de datos son conjuntos en $\mathbb{R}^d$, y la variable dependiente viene dada por un intervalo. Los datos intervalares y los datos afectados por algún tipo de ruido o incertidumbre son estudiados como dos casos particulares de nuestro modelo. La formulación propuesta está basada en el enfoque estándar de ϵ-Regresión de Vectores Soporte. En el caso de datos intervalares, se obtendrán dos formulaciones diferentes, según se escoja la manera de medir la distancia entre el intervalo de predicción y el real: la distancia del máximo o la distancia de Hausdorff.

Estas metodologías descritas en los Capítulos 2 y 3 resultan también útiles en la práctica para manejar el caso en el que tenemos valores perdidos en una base de datos y usamos imputación por intervalos para rellenar las coordenadas perdidas. Además, se prueba que nuestros modelos generalizan las formulaciones dadas en [107, 108, 109] para problemas de clasificación y regresión con datos afectados por algún tipo de perturbación, que se supone desconocida pero acotada para una norma dada.

El problema de regresión donde la incertidumbre sólo afecta a la variable dependiente de los elementos de la base de datos se estudia en el Capítulo 4, que está basado en nuestra referencia [19]. Se muestra un modelo basado en el enfoque estándar de ϵ-SVR, donde tenemos que construir dos hiperplanos para predecir el valor intervalar de la variable dependiente. Usando la distancia de Hausdorff para medir el error entre el intervalo de predicción y el real, se obtiene un problema de optimización cuadrática convexa.

Aunque este problema se puede ver como un caso particular de la formulación dada en el Capítulo 3, la ventaja de introducir este nuevo modelo se debe a que permite trabajar con regresores no lineales mediante el uso de estructuras de núcleo. Así, se pueden considerar relaciones más abstractas entre los datos para construir un regresor adecuado.

El Problema de Clasificación con Múltiples Ejemplos es el tema de estudio del Capítulo 5. Consideramos un problema de clasificación donde los elementos a clasificar son bolsas de ejemplos que son a su vez vectores que miden d atributos diferentes. Este modelo muestra un modo diferente de representar imprecisión en los datos, ya que para cada elemento de la base de datos, sólo algunos de sus ejemplos son realmente los que determinan la etiqueta asignada a la bolsa completa (y el resto pueden ser vistos como ruido). Sin embargo, no se pueden aplicar directamente el mismo tipo de herramientas descritas en el Capítulo 2, ya que los elementos en Clasificación con Múltiples Ejemplos son conjuntos discretos en $\mathbb{R}^d$, y los elementos en el modelo general de clasificación con imprecisión son conjuntos continuos.

La regla de clasificación viene definida en función de una bola, cuyo centro y radio son los parámetros a calcular. Dada una bolsa, la asignaremos a la clase positiva si al menos uno de sus ejemplos está incluido estrictamente dentro de la bola, y la etiquetaremos como negativa en caso contrario. Esta cuestión es modelada como un problema de optimización de margen. Se obtienen varias condiciones necesarias de optimalidad que conducen a un algoritmo polinomial en dimensión fija. Para resolver el problema, se propone un algoritmo de Búsqueda de Entorno Variable (ver [81]) basado en las condiciones de optimalidad del problema. Este trabajo es la base de nuestra referencia [18].

Finalmente, como una aplicación de la metodología desarrollada para problemas de clasificación con imprecisión, estudiamos un problema de localización en el Capítulo 6, que está basado en nuestro trabajo [51]. Tenemos que localizar una planta semi-nociva en el plano Euclídeo para dar servicio a un grupo de clientes. Simultáneamente, un conjunto de áreas pobladas, con figuras aproximadas mediante polígonos, deben ser protegidas de los efectos negativos de esa planta. Los clientes se pueden ver como la clase positiva, cuyos elementos son puntos en $\mathbb{R}^2$, mientras que las áreas pobladas

pueden ser vistas como la clase negativa, cuyos elementos son conjuntos continuos en $\mathbb{R}^2$, y hay que obtener una bola separadora, similar a la construida en el Capítulo 5.

El problema se formula como un modelo de maximización de margen. Se estudian las condiciones necesarias de optimalidad y se obtiene un conjunto dominante finito de soluciones, que conduce a un algoritmo en tiempo polinomial.

Se han realizado experimentos computacionales con conjuntos de datos de referencia y artificiales para cada modelo, obteniendo buenos resultados, lo que muestra que las herramientas desarrolladas son competitivas.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

This work studies modeling and algorithmic issues of supervised classification and regression problems. A short introduction and literature review of the main topics addressed follows.

1.1 Support Vector Machines

Support Vector Machines is an optimization-based methodology which has been successfully applied to classification and regression problems. In this section, a general description of Support Vector Machines for classification is given, and in Section 1.2, the ϵ-Support Vector Regression model (the adaptation of Support Vector Machines to the regression case) is explained. These techniques will be adapted in this thesis to the cases of classification and regression with imprecise data.

1.1.1 Description of the problem

In Support Vector Machines, SVMs for short (see e.g. [15, 21, 30, 31, 79, 82, 113, 114]), for classification problems, a database $\Omega \subseteq \mathbb{R}^d \times \mathbb{R}$ is given, with elements $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, where x_i is the feature vector and y_i is the class which the element belongs to. By abuse of notation, we identify the element (x_i, y_i) of the database with its index i. In binary classification, there are only two possible classes, that is, each element $i \in \Omega$ has assigned a label $y_i = +1$, if it belongs to the positive class, or $y_i = -1$, if belonging to the negative class.

Then, the database Ω can be decomposed into two different groups G_+ and G_-. The aim of SVMs is to introduce a hyperplane, with parameters $\omega \in \mathbb{R}^d$, $\beta \in \mathbb{R}$, separating these two groups of elements, in such a way that, given a new element $x \in \mathbb{R}^d$, one assigns to the corresponding class according to $sign(f(x)) = sign(\omega^\top x + \beta)$, i.e.,

$$\begin{array}{ll} \text{if } \omega^\top x + \beta > 0, & \text{then } y = +1 \\ \text{if } \omega^\top x + \beta < 0, & \text{then } y = -1. \end{array}$$

When it is possible to find a hyperplane separating the groups G_+ and G_-, one says that the two groups are linearly separable. In such case, there exist ω, β, such that

$$\begin{array}{l} \omega^\top x_i + \beta > 0, \quad \forall i \in G_+ \\ \omega^\top x_i + \beta < 0, \quad \forall i \in G_-, \end{array}$$

or equivalently,

$$y_i(\omega^\top x_i + \beta) > 0, \quad \forall i \in \Omega. \tag{1.1}$$

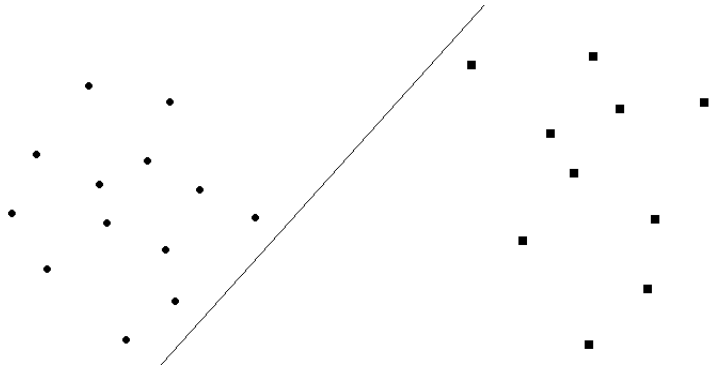

Figure 1.1: A possible separating hyperplane

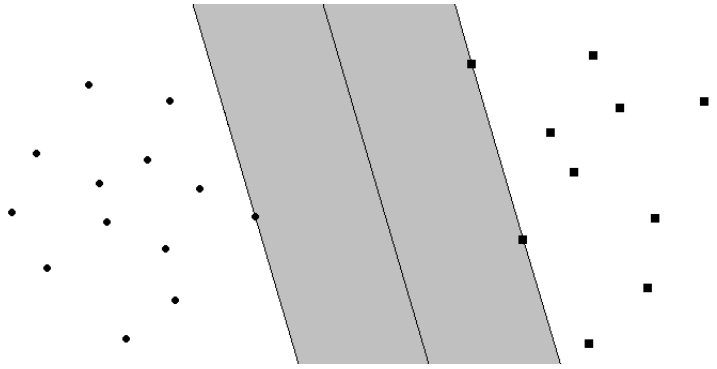

Figure 1.2: Classifier with maximum margin

This is shown to be equivalent to stating that the convex hulls of the sets $\{x_i : \ i \in G_+\}$, $\{x_i : \ i \in G_-\}$ are disjoint (see e.g. [21]).

Usually, when the two groups are linearly separable, infinite possible solutions can be found. In Figure 1.1, we show a possible hyperplane to separate these groups of squares and points, although, intuitively, it does not seem to be the best solution.

The solution proposed in a SVM framework is the separating hyperplane which maximizes the margin, where the margin is defined as the minimum distance from the elements of the database to the hyperplane. Figure 1.2 gives the graphical idea of the hyperplane with maximum margin separating the groups of squares and circles.

Then, given a training sample $I = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \subseteq \Omega$, we solve an optimization problem to obtain optimal parameters ω and β (optimal in the sense that the margin is maximized) for building the classification rule.

1.1.2 The optimization problem

The distance from a point $x^* \in \mathbb{R}^d$ to a hyperplane $H = \{x \in \mathbb{R}^d : \ \omega^\top x + \beta = 0\}$ is given by

$$dist(x^*, H) = \frac{|\omega^\top x^* + \beta|}{\|\omega\|^0}, \tag{1.2}$$

where $\|\omega\|^0$ represents the dual norm of the normal vector ω to the hyperplane H (see [92]).

Hence, the distance ρ_i from an element $i \in I$ to the region where it will be misclassified is

$$\rho_i(\omega, \beta) = \max\left\{\frac{y_i(\omega^\top x_i + \beta)}{\|\omega\|^0}, 0\right\}.$$

The margin on the training sample I is defined as the minimum of these distances,

$$\rho(\omega, \beta) = \min_{i \in I} \rho_i(\omega, \beta).$$

The aim is to find the separating hyperplane with maximum margin. Hence, the optimization problem is

$$\begin{array}{ll} \max\limits_{\omega,\beta} & \rho(\omega, \beta) \\ \text{s.t.} & y_i(\omega^\top x_i + \beta) > 0, \ \forall i \in I, \end{array} \tag{1.3}$$

which can also be written as

$$\begin{array}{ll} \max\limits_{\omega,\beta} & \min\limits_{i \in I} \dfrac{y_i(\omega^\top x_i + \beta)}{\|\omega\|^0} \\ \text{s.t.} & y_i(\omega^\top x_i + \beta) > 0, \ \forall i \in I. \end{array} \tag{1.4}$$

Since the problem is positively homogeneous in the variables, one can impose that

$$\min_{i \in I} \ y_i(\omega^\top x_i + \beta) = 1,$$

and then, Problem (1.4) remains as follows,

$$\begin{array}{ll} \max\limits_{\omega,\beta} & \dfrac{1}{\|\omega\|^0} \\ \text{s.t.} & \min\limits_{i \in I} \ y_i(\omega^\top x_i + \beta) = 1, \end{array}$$

or equivalently,

$$\begin{array}{ll} \min\limits_{\omega,\beta} & \|\omega\|^0 \\ \text{s.t.} & \min\limits_{i \in I} \ y_i(\omega^\top x_i + \beta) = 1. \end{array} \tag{1.5}$$

Moreover, Problem (1.5) is equivalent to

$$\begin{array}{ll} \min\limits_{\omega,\beta} & \|\omega\|^0 \\ \text{s.t.} & \min\limits_{i\in I}\ y_i(\omega^\top x_i+\beta)\geq 1. \end{array} \tag{1.6}$$

Indeed, the feasible region of (1.5) is included in the one of (1.6), so the optimal value z of (1.6) is smaller than or equal to the optimal value z' of (1.5). Furthermore, any (ω,β), feasible for (1.6), such that $\min\limits_{i\in I}\ y_i(\omega^\top x_i+\beta)>1$, is non-optimal for (1.6), since $(\overline{\omega},\overline{\beta})$, defined as

$$\begin{aligned} \overline{\omega} &= \frac{1}{\min\limits_{i\in I}\ y_i(\omega^\top x_i+\beta)}\cdot\omega \\ \overline{\beta} &= \frac{1}{\min\limits_{i\in I}\ y_i(\omega^\top x_i+\beta)}\cdot\beta, \end{aligned}$$

is feasible and satisfies that $\|\overline{\omega}\|^0<\|\omega\|^0$. Hence, $z'=z$.

Finally, this problem can be rewritten as

$$\begin{array}{ll} \min\limits_{\omega,\beta} & \|\omega\|^0 \\ \text{s.t.} & y_i(\omega^\top x_i+\beta)\geq 1,\ \forall i\in I. \end{array} \tag{1.7}$$

When using the Euclidean norm, the problem to be solved is equivalent to the following quadratic convex program with linear constraints,

$$\begin{array}{ll} \min\limits_{\omega,\beta} & \frac{1}{2}\sum\limits_{j=1}^{d}\omega_j^2 \\ \text{s.t.} & y_i(\omega^\top x_i+\beta)\geq 1,\ \forall i\in I. \end{array} \tag{1.8}$$

1.1.3 Soft-Margin formulation

Feasibility is not guaranteed in this problem. Some slack variables ξ must be introduced in the constraints and the objective function must be penalized. Then, the following Soft-Margin formulation is obtained (see e.g. [15, 30]),

$$\begin{array}{ll} \min\limits_{\omega,\beta,\xi} & \frac{1}{2}\sum\limits_{j=1}^{d}\omega_j^2+C\sum\limits_{i\in I}\xi_i \\ \text{s.t.} & y_i(\omega^\top x_i+\beta)\geq 1-\xi_i,\ \forall i\in I \\ & \xi_i\geq 0,\ \forall i\in I, \end{array} \tag{1.9}$$

with C being a constant in the model which trades off between a large margin and a small error penalty.

1.1.4 Dual formulation

Below, we derive the dual formulation for Problem (1.9) (see e.g. [15, 30, 31]). The dual formulation allows us to use non-linear functions as classifiers via the introduction of kernels.

Non-negative multipliers are introduced for every constraint in problem (1.9) and the Lagrangean function is built as follows,

$$L(\omega, \beta, \xi) = \frac{1}{2}\sum_{j=1}^{d}\omega_j^2 + C\sum_{i\in I}\xi_i + \sum_{i\in I}\lambda_i(1 - \xi_i - y_i\sum_{j=1}^{d}\omega_j x_{ij} - y_i\beta) - \sum_{i\in I}\mu_i\xi_i.$$

Optimality conditions are obtained by setting the partial derivatives of L equal to zero,

$$\frac{\partial L}{\partial \omega_j} = \omega_j - \sum_{i\in I}\lambda_i y_i x_{ij} = 0,\ j = 1, \ldots, d \tag{1.10}$$

$$\frac{\partial L}{\partial \beta} = -\sum_{i\in I}\lambda_i y_i = 0 \tag{1.11}$$

$$\frac{\partial L}{\partial \xi_i} = C - \lambda_i - \mu_i = 0,\ \forall i \in I. \tag{1.12}$$

From constraint (1.10), we obtain the expression to compute the vector ω from the dual variables,

$$\omega_j = \sum_{i\in I}\lambda_i y_i x_{ij},\ j = 1, \ldots, d. \tag{1.13}$$

From (1.11), we obtain a constraint to be included in the dual formulation,

$$\sum_{i\in I}\lambda_i y_i = 0. \tag{1.14}$$

Since the multipliers μ_i are non-negative, expression (1.12) states that the multipliers λ_i are bounded by the constant of the model C,

$$0 \leq \lambda_i \leq C,\ \forall i \in I. \tag{1.15}$$

Then, by using these constraints, the objective function for the dual formulation remains as follows,

$$\begin{aligned}\tilde{L}(\lambda) &= -\frac{1}{2}\sum_{j=1}^{d}(\sum_{i\in I}\lambda_i y_i x_{ij})(\sum_{l\in I}\lambda_l y_l x_{lj}) + \sum_{i\in I}\lambda_i \\ &= -\frac{1}{2}\sum_{i,l\in I}\lambda_i\lambda_l y_i y_l x_i^\top x_l + \sum_{i\in I}\lambda_i.\end{aligned}$$

With this objective function and adding the constraints (1.14)-(1.15), we obtain the following dual formulation as a concave quadratic maximization program in the variables λ,

$$\begin{array}{ll} \max\limits_{\lambda} & \sum\limits_{i\in I}\lambda_i - \dfrac{1}{2}\sum\limits_{i,l\in I}\lambda_i\lambda_l y_i y_l x_i^\top x_l \\ \text{s.t.} & \sum\limits_{i\in I}\lambda_i y_i = 0 \\ & 0 \le \lambda_i \le C,\ \forall i \in I. \end{array} \tag{1.16}$$

Given an optimal solution of Problem (1.16), one can recover an optimal solution of Problem (1.9). Indeed, the vector ω can be computed with expression (1.13), and for β, we use that the Karush-Kuhn-Tucker conditions must be satisfied.

$$\lambda_i \cdot (1 - \xi_i - y_i \sum_{j=1}^{d} \omega_j x_{ij} - y_i\beta) = 0, \quad \forall i \in I \tag{1.17}$$

$$\xi_i \cdot (C - \lambda_i) = 0, \quad \forall i \in I \tag{1.18}$$

$$0 \le \lambda_i \le C, \quad \forall i \in I. \tag{1.19}$$

If $\exists i \in I$ such that $0 < \lambda_i < C$, then, by (1.18), one has that $\xi_i = 0$, and from (1.17), one can recover the value of β,

$$\beta = y_i(1 - y_i\sum_{j=1}^{d} \omega_j x_{ij}) = y_i - \sum_{j=1}^{d} \omega_j x_{ij}. \tag{1.20}$$

Otherwise, if $\lambda_i \in \{0, C\}$, $\forall i \in I$, when $\lambda_i = 0$, by (1.18), the slack variable is also $\xi_i = 0$, and since (ω, β, ξ) is feasible for (1.9),

$$1 - y_i \sum_{j=1}^{d} \omega_j x_{ij} - y_i\beta \le 0, \text{ i.e., } y_i\beta \ge 1 - y_i \sum_{j=1}^{d} \omega_j x_{ij},$$

then,

$$\text{if } y_i = +1, \quad \text{then } \beta \ge y_i - \sum_{j=1}^{d} \omega_j x_{ij} \tag{1.21}$$

$$\text{if } y_i = -1, \quad \text{then } \beta \le y_i - \sum_{j=1}^{d} \omega_j x_{ij}, \tag{1.22}$$

and by taking the maximum and minimum, respectively, in expressions (1.21) and (1.22), one has that

$$\max_{\{i\in G_+ : \lambda_i = 0\}} \{1 - \sum_{j=1}^{d} \omega_j x_{ij}\} \le \beta \le \min_{\{i\in G_- : \lambda_i = 0\}} \{-1 - \sum_{j=1}^{d} \omega_j x_{ij}\}. \tag{1.23}$$

On the other hand, if $\lambda_i = C$, then the slack variable can be positive and the corresponding constraint in (1.17) becomes active, that is,

$$1 - \xi_i - y_i \sum_{j=1}^{d} \omega_j x_{ij} - y_i \beta = 0, \text{ i.e., } y_i \beta \leq 1 - y_i \sum_{j=1}^{d} \omega_j x_{ij}.$$

With a similar reasoning to that derived for $\lambda_i = 0$, one obtains that

$$\max_{\{i \in G_- : \lambda_i = C\}} \{-1 - \sum_{j=1}^{d} \omega_j x_{ij}\} \leq \beta \leq \min_{\{i \in G_+ : \lambda_i = C\}} \{1 - \sum_{j=1}^{d} \omega_j x_{ij}\}. \tag{1.24}$$

In the other direction, given λ optimal for (1.16), with $\lambda_i \in \{0, C\}$, $\forall i \in I$, ω as defined in (1.13), then any β satisfying (1.23)-(1.24) is optimal for (1.9), that is, there exists ξ such that (ω, β, ξ) and λ jointly satisfy the KKT system (1.17)-(1.19).

Indeed, for $i \in I$ such that $\lambda_i = 0$, (1.17)-(1.18) are trivially satisfied with $\xi_i = 0$.

Since β satisfies (1.24), one has that

$$\beta \leq 1 - \sum_{j=1}^{d} \omega_j x_{ij}, \quad \forall i \in G_+ : \ \lambda_i = C \tag{1.25}$$

$$\beta \geq -1 - \sum_{j=1}^{d} \omega_j x_{ij}, \quad \forall i \in G_- : \ \lambda_i = C. \tag{1.26}$$

Then, for every $i \in G_+$ such that $\lambda_i = C$, by (1.17), one obtains the value of ξ_i,

$$\xi_i = 1 - \sum_{j=1}^{d} \omega_j x_{ij} - \beta,$$

which is non-negative by (1.25).

Analogously, for every $i \in G_-$ such that $\lambda_i = C$, by (1.17), we compute the value of ξ_i as

$$\xi_i = 1 + \sum_{j=1}^{d} \omega_j x_{ij} + \beta,$$

which is bigger than or equal to zero by (1.26).

Hence, if every $\lambda_i \in \{0, C\}$, then one may choose any β satisfying (1.23)-(1.24).

From the KKT conditions (1.17)-(1.19), we can observe that the instances with associated multiplier $\lambda_i = C$ lie outside the region containing the instances of its group, and the instances with associated multiplier $0 < \lambda_i < C$ lie exactly on the boundary of this region. These instances with multiplier bigger than zero are called *support vectors* and they are necessary to compute the classifier. In fact, if we erase all the instances with $\lambda_i = 0$ from the training sample and we recompute the classifier, the result would be the same.

1.1.5 Dual kernel-based formulation

Kernels are used to project the data from the input space $\mathcal{X} \subseteq \mathbb{R}^d$ into a high dimensional feature space, where more abstract features of the data can be exploited. This way, since the data are projected to the feature space via a usually non-linear mapping, non-linear relations between the data can be extracted by means of a linear function in the feature space (see [31, 58, 100]).

Definition 1.1 *Let $\phi : \mathcal{X} \subseteq \mathbb{R}^d \to \mathcal{F}$ be a (usually non-linear) mapping from the input space $\mathcal{X}$ into the feature space $\mathcal{F}$. A kernel K is a function $K : \mathcal{X} \times \mathcal{X} \subseteq \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, such that for every $x, y \in \mathcal{X}$,*

$$K(x, y) = \phi(x)^\top \phi(y).$$

The explicit expression of the mapping ϕ will not be needed, but only an algorithm to evaluate the kernel K in each pair of values x and y.

For introducing a kernel structure in Problem (1.16), we only need to change $x_i^\top x_l$ in the objective function by a more general kernel structure evaluated in that pair of values $K(x_i, x_l)$

$$\begin{array}{ll} \max\limits_{\lambda} & \sum\limits_{i \in I} \lambda_i - \dfrac{1}{2} \sum\limits_{i,l \in I} \lambda_i \lambda_l y_i y_l K(x_i, x_l) \\ \text{s.t.} & \sum\limits_{i \in I} \lambda_i y_i = 0 \\ & 0 \leq \lambda_i \leq C, \ \forall i \in I. \end{array} \tag{1.27}$$

Expression (1.13) becomes now

$$\omega = \sum_{i \in I} \lambda_i y_i \phi(x_i), \ j = 1, \ldots, d.$$

which cannot be evaluated unless that an explicit representation of ϕ is given.

But, given a new element, one can predict the class which it belongs to, since the kernel is known. Indeed, the function $f(x) = \omega^\top x + \beta$ is now transformed into

$$f(x) = \omega^\top x + \beta = \sum_{i \in I} \lambda_i y_i \phi(x_i)^\top \phi(x) + \beta = \sum_{i \in I} \lambda_i y_i K(x_i, x) + \beta.$$

For obtaining β, if $\exists i$ such that $0 < \lambda_i < C$, from (1.20), we obtain

$$\beta = y_i - \sum_{l \in I} \lambda_l y_l \phi(x_l)^\top \phi(x_i) = y_i - \sum_{l \in I} \lambda_l y_l K(x_l, x_i).$$

In case that $\lambda_i \in \{0, C\}$, $\forall i \in I$, one may choose any β such that

$$\begin{array}{lll} \max\limits_{\{i \in G_+ : \lambda_i = 0\}} \{1 - \sum\limits_{l \in I} \lambda_l y_l K(x_l, x_i)\} & \leq \beta \leq & \min\limits_{\{i \in G_- : \lambda_i = 0\}} \{-1 - \sum\limits_{l \in I} \lambda_l y_l K(x_l, x_i)\} \\ \max\limits_{\{i \in G_- : \lambda_i = C\}} \{-1 - \sum\limits_{l \in I} \lambda_l y_l K(x_l, x_i)\} & \leq \beta \leq & \min\limits_{\{i \in G_+ : \lambda_i = C\}} \{1 - \sum\limits_{l \in I} \lambda_l y_l K(x_l, x_i)\}. \end{array}$$

Finally, given a new element, the label is predicted according to

$$x \hookrightarrow sign(f(x)) := sign(\sum_{i \in I} \lambda_i y_i K(x_i, x) + \beta), \tag{1.28}$$

where the only difference is that the value $\omega^\top x$ has been computed by using the kernel expression, avoiding to use the expression of the mapping ϕ.

1.2 Support Vector Regression

1.2.1 Formulation of the problem

In the standard ϵ-Support Vector Regression, ϵ-SVR for short (see e.g. [31, 42, 52, 102, 113, 114]), a database $\Omega \subseteq \mathbb{R}^d \times \mathbb{R}$ is given, with elements $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, where x_i is the set of predictor variables and y_i is the dependent variable, whose value is to be predicted from the value of x_i.

The aim of ϵ-SVR is to find $\omega \in \mathbb{R}^d$ and $\beta \in \mathbb{R}$ such that, for each instance $i \in \Omega$, the affine function $f(x) = \omega^\top x + \beta$ yields a small deviation (at most ϵ) between the observed value y_i and the predicted value $f(x_i)$.

Given a training sample $I = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, extracted from the database Ω, we formulate the optimization problem to solve to obtain the optimal parameters ω and β for the regression task.

Since the deviation between y_i and $f(x_i)$ must be at most ϵ, the following set of constraints is obtained

$$|\omega^\top x_i + \beta - y_i| \leq \epsilon, \ \forall i \in I.$$

The optimization problem to solve, as stated in [102], is the following,

$$\begin{array}{ll} \min\limits_{\omega, \beta} & \frac{1}{2} \sum\limits_{j=1}^{d} \omega_j^2 \\ \text{s.t.} & y_i - \omega^\top x_i - \beta \leq \epsilon, \ \forall i \in I \\ & \omega^\top x_i + \beta - y_i \leq \epsilon, \ \forall i \in I. \end{array} \tag{1.29}$$

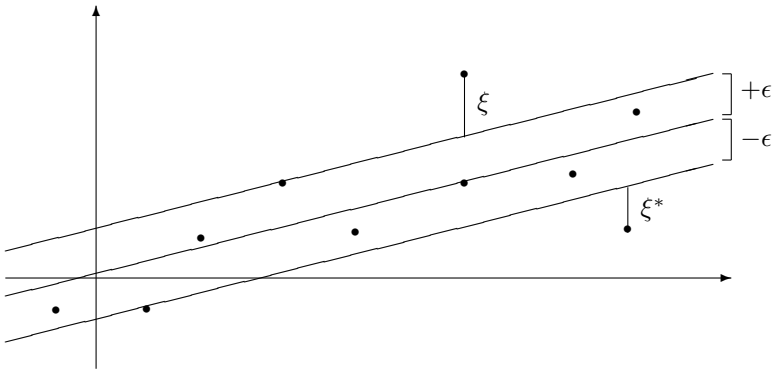

Figure 1.3: ϵ-Support Vector Regression

This optimization problem can be infeasible. Hence, one must introduce some slack variables ξ, ξ^* in the constraints (as done in Subsection 1.1.3, in the Soft-Margin case for SVMs) and a penalty term must be added to the objective function. The optimization problem has then the following form (see e.g. [102, 113]),

$$\begin{array}{ll} \min\limits_{\omega,\beta,\xi,\xi^*} & \frac{1}{2}\sum\limits_{j=1}^{d}\omega_j^2 + C\sum\limits_{i\in I}(\xi_i+\xi_i^*) \\ \text{s.t.} & y_i - \omega^\top x_i - \beta \leq \epsilon + \xi_i,\ \forall i \in I \\ & \omega^\top x_i + \beta - y_i \leq \epsilon + \xi_i^*,\ \forall i \in I \\ & \xi_i, \xi_i^* \geq 0,\ \forall i \in I, \end{array} \tag{1.30}$$

with C and ϵ constants of the model, where ϵ is the maximum allowed deviation for the instances of the training sample and C represents the trade-off between the flatness of the prediction function and the sum of deviations larger than ϵ.

Figure 1.3 explains graphically the model. We seek a hyperplane to fit the points of the dataset, but only the points whose deviation from the predicted value (the corresponding point lying on the hyperplane) is larger than ϵ will be penalized. That is, the points outside the band defined by the hyperplane and the parameter ϵ, the so-called *ϵ-insensitive tube*, will be penalized via the corresponding slack variable (variable ξ for points above the tube, and ξ^* for points below the tube).

Formulation (1.30), introduced by [113], corresponds to deal with the so-called *ϵ-insensitive loss function*, which is defined as

$$|\xi|_\epsilon = \begin{cases} 0 & \text{if } |\xi| \leq \epsilon, \\ |\xi| - \epsilon & \text{otherwise.} \end{cases}$$

A loss function is a cost function to penalize the errors in the prediction task. In particular, with the ϵ-insensitive loss function, deviations smaller than the fixed amount ϵ are allowed, while bigger deviations are penalized linearly. Figure 1.4 shows the shape of this loss function.

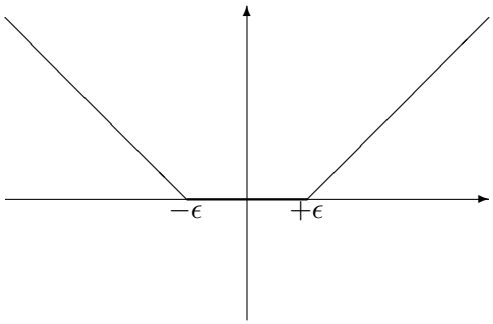

Figure 1.4: ϵ-insensitive loss function

Other loss functions, like the Gaussian function or Huber's function (with quadratic penalizations of the errors) or the Laplacian function (which can be seen as the particular case for $\epsilon = 0$ of the ϵ-insensitive loss function) have also been applied in the literature of this topic (see [52, 102]).

However, the advantage of the ϵ-insensitive loss function is the sparseness of the support vectors, because for this loss function not every point will be a support vector, unlike for the other loss functions (see e.g. [52]).

1.2.2 Dual formulation

Below, the dual of Problem (1.30) is formulated (see e.g. [31, 102]). This dual formulation can also be used to obtain an optimal solution of our problem and will allow us to handle non-linear functions for regression tasks via the introduction of kernels.

First, we introduce non-negative Lagrange multipliers for every constraint of Problem (1.30) and we derive the Lagrangean function as follows,

$$\begin{aligned} L(\omega, \beta, \xi, \xi^*) &= \frac{1}{2}\sum_{j=1}^{d}\omega_j^2 + C\sum_{i\in I}(\xi_i + \xi_i^*) + \sum_{i\in I}\lambda_i(y_i - \sum_{j=1}^{d}\omega_j x_{ij} - \beta - \epsilon - \xi_i) \\ &\quad + \sum_{i\in I}\lambda_i^*(\sum_{j=1}^{d}\omega_j x_{ij} + \beta - y_i - \epsilon - \xi_i^*) - \sum_{i\in I}(\mu_i\xi_i + \mu_i^*\xi_i^*). \end{aligned}$$

The partial derivatives of the Lagrangean function are set equal to zero, and we

obtain

$$\frac{\partial L}{\partial \omega_j} = \omega_j - \sum_{i\in I}\lambda_i x_{ij} + \sum_{i\in I}\lambda_i^* x_{ij} = 0,\ j = 1,\dots,d \tag{1.31}$$

$$\frac{\partial L}{\partial \beta} = -\sum_{i\in I}\lambda_i + \sum_{i\in I}\lambda_i^* = 0 \tag{1.32}$$

$$\frac{\partial L}{\partial \xi_i} = C - \lambda_i - \mu_i = 0,\ \forall i \in I \tag{1.33}$$

$$\frac{\partial L}{\partial \xi_i^*} = C - \lambda_i^* - \mu_i^* = 0,\ \forall i \in I. \tag{1.34}$$

From constraint (1.31), we obtain an expression to compute the vector ω as a linear combination of the Lagrangean multipliers,

$$\omega_j = \sum_{i\in I}(\lambda_i - \lambda_i^*)x_{ij},\ j = 1,\dots,d. \tag{1.35}$$

Constraint (1.32) is included in the problem as

$$\sum_{i\in I}\lambda_i = \sum_{i\in I}\lambda_i^*, \tag{1.36}$$

and, since the multipliers μ_i, μ_i^* are non-negative, constraints (1.33)-(1.34) state that the multipliers λ_i, λ_i^* are bounded by the parameter C,

$$0 \le \lambda_i \le C, \quad \forall i \in I \tag{1.37}$$

$$0 \le \lambda_i^* \le C, \quad \forall i \in I. \tag{1.38}$$

By using the constraints derived, the objective function for the dual problem remains as follows,

$$\begin{aligned}\tilde{L}(\lambda,\lambda^*) &= -\frac{1}{2}\sum_{j=1}^{d}\left(\sum_{i\in I}(\lambda_i - \lambda_i^*)x_{ij}\right)\left(\sum_{l\in I}(\lambda_l - \lambda_l^*)x_{lj}\right) \\ &\quad -\epsilon\sum_{i\in I}(\lambda_i + \lambda_i^*) + \sum_{i\in I}(\lambda_i - \lambda_i^*)y_i \\ &= -\frac{1}{2}\sum_{i,l\in I}(\lambda_i - \lambda_i^*)(\lambda_l - \lambda_l^*)x_i^\top x_l - \epsilon\sum_{i\in I}(\lambda_i + \lambda_i^*) + \sum_{i\in I}(\lambda_i - \lambda_i^*)y_i,\end{aligned}$$

and adding the constraints (1.36)-(1.38), the dual formulation is

$$\begin{aligned}\max_{\lambda,\lambda^*}\quad & -\frac{1}{2}\sum_{i,l\in I}(\lambda_i - \lambda_i^*)(\lambda_l - \lambda_l^*)x_i^\top x_l - \epsilon\sum_{i\in I}(\lambda_i + \lambda_i^*) + \sum_{i\in I}(\lambda_i - \lambda_i^*)y_i \\ \text{s.t.}\quad & \sum_{i\in I}(\lambda_i - \lambda_i^*) = 0 \\ & 0 \le \lambda_i, \lambda_i^* \le C, \quad \forall i \in I,\end{aligned} \tag{1.39}$$

which is a concave quadratic maximization problem in the variables λ, λ^*.

Given an optimal solution (λ, λ^*) of Problem (1.39), we can recover an optimal solution of Problem (1.30). First, we use (1.35) to compute ω. For the variables β, ξ and ξ^*, we use that the following Karush-Kuhn-Tucker conditions must be satisfied.

$$\lambda_i \cdot (y_i - \sum_{j=1}^{d} \omega_j x_{ij} - \beta - \epsilon - \xi_i) = 0, \quad \forall i \in I \tag{1.40}$$

$$\lambda_i^* \cdot (\sum_{j=1}^{d} \omega_j x_{ij} + \beta - y_i - \epsilon - \xi_i^*) = 0, \quad \forall i \in I \tag{1.41}$$

$$\xi_i \cdot (C - \lambda_i) = 0, \quad \forall i \in I \tag{1.42}$$

$$\xi_i^* \cdot (C - \lambda_i^*) = 0, \quad \forall i \in I \tag{1.43}$$

$$0 \leq \lambda_i, \lambda_i^* \leq C, \quad \forall i \in I. \tag{1.44}$$

If $\exists i : 0 < \lambda_i < C$, expression (1.42) implies that $\xi_i = 0$, and from (1.40), we recover the value of β as

$$\beta = y_i - \sum_{j=1}^{d} \omega_j x_{ij} - \epsilon. \tag{1.45}$$

If $\exists i : 0 < \lambda_i^* < C$, expression (1.43) implies that $\xi_i^* = 0$ and the value of β can be obtained from (1.41) as

$$\beta = y_i - \sum_{j=1}^{d} \omega_j x_{ij} + \epsilon. \tag{1.46}$$

If every λ_i and λ_i^* belongs to $\{0, C\}$, when $\lambda_i = 0$, by (1.42), one has that $\xi_i = 0$, and since $(\omega, \beta, \xi, \xi^*)$ is feasible for (1.30),

$$y_i - \sum_{j=1}^{d} \omega_j x_{ij} - \beta - \epsilon \leq 0, \text{ i.e., } \beta \geq y_i - \sum_{j=1}^{d} \omega_j x_{ij} - \epsilon. \tag{1.47}$$

If $\lambda_i = C$, the corresponding slack variable can be positive and the associated constraint in (1.40) becomes active,

$$y_i - \sum_{j=1}^{d} \omega_j x_{ij} - \beta - \epsilon - \xi_i = 0, \text{ i.e., } \beta \leq y_i - \sum_{j=1}^{d} \omega_j x_{ij} - \epsilon. \tag{1.48}$$

With a similar reasoning with the multipliers λ_i^*, one obtains that

$$\text{if } \lambda_i^* = 0, \quad \text{then } \beta \leq y_i - \sum_{j=1}^{d} \omega_j x_{ij} + \epsilon \tag{1.49}$$

$$\text{if } \lambda_i^* = C, \quad \text{then } \beta \geq y_i - \sum_{j=1}^{d} \omega_j x_{ij} + \epsilon. \tag{1.50}$$

If we take maximum in expressions (1.47) and (1.50), and we take minimum in expressions (1.48) and (1.49), β must satisfy the following constraints,

$$\begin{array}{rcl} \max\limits_{\{i\in I:\lambda_i=0\}}\{y_i-\sum\limits_{j=1}^{d}\omega_j x_{ij}-\epsilon\} & \leq \beta \leq & \min\limits_{\{i\in I:\lambda_i^*=0\}}\{y_i-\sum\limits_{j=1}^{d}\omega_j x_{ij}+\epsilon\} \\ \max\limits_{\{i\in I:\lambda_i^*=C\}}\{y_i-\sum\limits_{j=1}^{d}\omega_j x_{ij}+\epsilon\} & \leq \beta \leq & \min\limits_{\{i\in I:\lambda_i=C\}}\{y_i-\sum\limits_{j=1}^{d}\omega_j x_{ij}-\epsilon\}, \end{array} \tag{1.51}$$

and hence, β must belong to these intervals. In fact, as shown for SVMs in Subsection 1.1.4, one has that, when $\lambda_i, \lambda_i^* \in \{0, C\}$, $\forall i \in I$, any β satisfying (1.51) can be chosen as optimal solution of Problem (1.30).

From the KKT conditions (1.40)-(1.44), one can observe that the instances with an associated multiplier $\lambda_i = C$ (respectively, $\lambda_i^* = C$) lie outside the ϵ-insensitive tube, while the instances with both multipliers equal to zero are strictly included in the ϵ-insensitive tube. The instances whose associated multiplier satisfies $0 < \lambda_i < C$ or $0 < \lambda_i^* < C$ lie on the boundary of the tube.

The instances with one multiplier strictly bigger than zero are called *support vectors*, and they are necessary to compute the prediction function. In fact, if we erase from the training sample the instances with $\lambda_i = \lambda_i^* = 0$, the resulting prediction function would be the same.

Finally, at least one of the multipliers must be equal to zero, that is,

$$\lambda_i \cdot \lambda_i^* = 0, \ \forall i \in I,$$

since the same point cannot be a support vector for both sides of the ϵ-insensitive tube, for any $\epsilon > 0$.

1.2.3 Dual kernel-based formulation

Below, we build the dual formulation for Problem (1.30) by using a kernel structure. For this, it is enough to replace $x_i^\top x_l$ in the objective function of Problem (1.39) by another general kernel structure $K(x_i, x_l)$, and we obtain the following dual kernel-based formulation,

$$\begin{array}{ll} \max\limits_{\lambda,\lambda^*} & -\frac{1}{2}\sum\limits_{i,l\in I}(\lambda_i-\lambda_i^*)(\lambda_l-\lambda_l^*)K(x_i,x_l)-\epsilon\sum\limits_{i\in I}(\lambda_i+\lambda_i^*)+\sum\limits_{i\in I}(\lambda_i-\lambda_i^*)y_i \\ \text{s.t.} & \sum\limits_{i\in I}(\lambda_i-\lambda_i^*)=0 \\ & 0\leq\lambda_i,\lambda_i^*\leq C, \quad \forall i\in I. \end{array} \tag{1.52}$$

The expression for ω, obtained from (1.35), cannot be used explicitly, since the expression of the mapping ϕ is unknown,

$$\omega = \sum_{i \in I} (\lambda_i - \lambda_i^*) \phi(x_i).$$

However, we will not need it to give the prediction for any new element x.

For obtaining the expression of β, if $\exists i : 0 < \lambda_i < C$ or $0 < \lambda_i^* < C$, we obtain expression (1.53) or (1.54), respectively,

$$\beta = y_i - \sum_{l \in I} (\lambda_l - \lambda_l^*) K(x_l, x_i) - \epsilon. \tag{1.53}$$

$$\beta = y_i - \sum_{l \in I} (\lambda_l - \lambda_l^*) K(x_l, x_i) + \epsilon. \tag{1.54}$$

When every $\lambda_i, \lambda_i^* \in \{0, C\}$, any β satisfying the following expressions can be selected,

$$\max_{\{i \in I : \lambda_i = 0\}} \{y_i - \sum_{l \in I} (\lambda_l - \lambda_l^*) K(x_l, x_i) - \epsilon\} \leq \beta \leq \min_{\{i \in I : \lambda_i^* = 0\}} \{y_i - \sum_{l \in I} (\lambda_l - \lambda_l^*) K(x_l, x_i) + \epsilon\}$$

$$\max_{\{i \in I : \lambda_i^* = C\}} \{y_i - \sum_{l \in I} (\lambda_l - \lambda_l^*) K(x_l, x_i) + \epsilon\} \leq \beta \leq \min_{\{i \in I : \lambda_i = C\}} \{y_i - \sum_{l \in I} (\lambda_l - \lambda_l^*) K(x_l, x_i) - \epsilon\}.$$

Then, given a new element x, its prediction, by using the kernel structure, is the following,

$$f(x) = \sum_{l \in I} (\lambda_l - \lambda_l^*) K(x_l, x) + \beta. \tag{1.55}$$

1.3 Classification and regression with imprecise data

1.3.1 Intervals and imprecise data

In certain situations, data cannot be expressed via single feature vectors, and interval data must be introduced. This way, intervals are used for expressing ranges, such as the range of temperature during a day, age intervals for a group of individuals or the cost of certain items in the set of shops of a town (since there will be some variations in the price between the different shops). Intervals can also be used when several measures of the same variable have been taken from an individual, and one wants to summarize these measurements, for example, the fluctuations of blood pressure or pulse rate of a patient, or the weight of a newborn during his/her first week.

Intervals also naturally occur in case of imprecise data, or when an estimation of a certain parameter must be performed via a confidence interval, and, in general, whenever uncertainty or vagueness arises in our problem.

Another case of interval data appears in the framework of Symbolic Data Analysis ([11, 13]), when one needs to summarize large databases in such a way that the resulting dataset has a more manageable size and it retains enough knowledge from the original database. Different approaches exist to aggregate the data by using classical variables (single values), multi-valued variables (categorical variables which can have several results), interval-valued variables (the data are aggregated into intervals and this is the case of our interest) or modal variables (a single, interval or nominal variable which can have different values with different probabilities associated).

1.3.2 Classification with imprecise data: cases of interest and literature

Consider a supervised classification problem where elements of the dataset are not single points, but sets in $\mathbb{R}^d$, these sets representing any kind of imprecise data. A label (+1 or -1, in the two-class case) must be assigned to the complete set, according to the behaviour of the most of its elements with respect to a certain classification rule.

Three cases of interest which can be approached with this model are the following: interval data, data affected by some kind of perturbation and data with missing values.

- **Interval data:** In particular, one interesting case which can be modeled via this methodology is when elements X_i to be classified are defined as a Cartesian product of intervals, $X_i = \prod_{j=1}^{d}[l_{ij}, u_{ij}] \subset \mathbb{R}^d$, with $l_{ij} \leq u_{ij}$, that is, l_{ij} and u_{ij} represent, respectively, the lower and upper bounds of each coordinate, these bounds coinciding in some cases.

  Classification with interval data has been studied in the literature by following different strategies. Recently, in [3], a formulation with Support Vector Machines has been proposed for the classification problem with intervals in an independent work to ours, only for the linearly separable case. In [43], linear discriminant analysis is applied to this type of classification problems by considering three different techniques: assigning a uniform distribution to each interval, expanding the dataset into the corresponding set of vertices and describing each interval via its center and its range. In [39], a Radial Basis Function kernel is built via a Hausdorff distance between intervals, and is applied to classification of intervals. Another kernel for intervals, based on the intersection operation, is described in [96].

  Other techniques, involving Neural Networks, are described in [95, 101]. Two

methods that allow to use interval data as inputs for a multi-layer perceptron are included in [95] (one of them based on a description of the intervals as single points and the other based on a probabilistic understanding of the intervals). A neural expert system for diagnosis is created in [101], where the knowledge base is a Neural Network which is built automatically through a learning algorithm. Likewise, classification methods for symbolic data are explained in [90].

- **Perturbed data:** Another case of interest, which can be modeled with this approach, is when data are affected by some kind of noise or perturbation. In that case, one must build a robust classifier, insensitive to this noise in the data. One model of Robust Support Vector Machines has been studied in [108, 109], where an optimization problem must be solved via Second Order Cone Programming.

  One can find another approach to robust classification in [46], where a binary classification problem is stated in which the data are unknown, but are bounded in hyper-rectangles. In that paper, the authors design a robust classifier by minimizing the worst-case value of a given loss function. Three different functions are considered, including the linear Hinge loss (see [31]) for SVMs, which provides an upper bound on the number of future expected misclassification errors. In [68], another binary classification problem is formulated where the data are given by the mean and covariance of each class, which are assumed to be known. The objective is minimizing the worst-case (maximum) probability of future misclassified data points, and Second Order Cone Programming techniques are used to solve it. Geometrically, it can be seen as the problem of minimizing the maximum of the Mahalanobis distances between the two classes. With a similar strategy, in [8] the values of a data point are described by a data uncertainty set, which is defined via a bounded ellipsoid parameterized by its location (expected value or center of the ellipsoid) and its shape (covariance matrix or the matrix of squared axis lengths).

- **Missing values:** Another situation which can be included in the framework of classification with imprecise data is the case in which there exist missing values (see [73] for a study on statistical analysis in datasets with missing values), that is, when the database is formed by feature vectors but some of their coordinates do not appear in the dataset. Different techniques have been used in the literature to handle missing data in classification problems (for a survey on the topic, see [74, 75]). Although the most popular technique is to impute single values (based on the rest of values of the database) to replace the missing coordinates, the imputation via intervals allows us to study this problem with the tools for interval-data classification.

Our aim with the model proposed in this thesis for classification with imprecise data is to formulate a more general model, such that the cases of interval data, data with

perturbation and missing values can be studied as particular cases, and with a better performance on real databases which have already been used in previous works of the literature. Furthermore, we also consider the extension to the multi-class problem.

1.3.3 Regression with imprecise data: cases of interest and literature

Consider a regression problem where the elements of the database are not single points, but sets in $\mathbb{R}^d$, and the dependent variable is not a single value, but an interval. That is, the dependent and the predictor variables are all affected by some kind of imprecision. Given a new element, a set in $\mathbb{R}^d$, an interval output must be assigned to that set, by taking into account the complete set.

As in the classification framework, different cases of interest can be included in this description of the model, like the following ones: interval data (data with interval input and interval output), single-input and interval-output data (imprecision only in the dependent variable), data with some kind of perturbation and missing values.

- **Interval data:** As in the classification problem, one case of interest which can be approached with this methodology is when each element X_i of the database is a Cartesian product, $X_i = \prod_{j=1}^{d}[l_{ij}, u_{ij}] \subset \mathbb{R}^d$, with $l_{ij} \leq u_{ij}$, l_{ij} and u_{ij} representing, respectively, the lower and upper bounds of each coordinate. Furthermore, the dependent variable is also an interval $Y_i = [\tilde{l}_i, \tilde{u}_i]$, with $\tilde{l}_i \leq \tilde{u}_i$.

  Regression for interval-input and interval-output data has been studied in the literature, by using different strategies. From a Symbolic Data Analysis perspective, the first work published on this topic appeared in [9]. Consider the classical linear regression model (see e.g. [41]),

  $$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_d x_{id} + \varepsilon_i,\ i = 1, \ldots, n,$$

  where y_i is the dependent variable, $x_i = (x_{i1}, \ldots, x_{id})^\top \in \mathbb{R}^d$ is the vector of the predictor variables, $\beta_0, \beta_1, \ldots, \beta_d$ are the coefficients of the regression model and ε_i is the error. The approach in [9] consisted in fitting the classical linear regression model on the midpoint of the intervals of each variable of the dataset. The predicted lower and upper bounds for the dependent variable were computed on the obtained model. This model is improved in [34], where two linear regression models are used, one for predicting the midpoint of the output and the other one for predicting the range. The predicted lower and upper bounds for the dependent variable are recovered with the midpoint and range. In [70], a comparison between these two models is shown. Other extensions of these models can be found in [10, 71].

Related to this problem, we also find the concept of fuzzy regression analysis, where several approaches have been developed. Roughly, they can be classified in two main groups: the possibilistic approach (initially proposed in [105]), where the objective function to be minimized is a measurement of the spread of the predicted output, and the least-squares approach (introduced in [24, 36]), which minimizes a distance, on fuzzy numbers, between the real and the predicted output.

SVMs have also been applied to fuzzy multiple linear regression models (see [60, 61]). Two different models have been studied in these works: when the predictor and the dependent variables are symmetric triangular fuzzy numbers (fuzzy input and fuzzy output) and when the predictor variables are crisp and the dependent variable is a triangular fuzzy number (crisp input and fuzzy output). The standard ϵ-SVR methodology is applied by imposing that the mode and the extremes of the intervals must satisfy the usual constraints. In the crisp-input and fuzzy-output case, non-linear regressors are introduced via kernel methods.

- **Single-input and imprecise-output:** The situation in which the predictor variables are single-valued and only the dependent variable is interval-valued can be studied as a particular case of the previous one (interval data). However, this model deserves a deeper study to introduce non-linear functions for regression tasks via kernel structures.

  In the bibliography, a first problem related to this involves the concept of interval regression analysis, which is the simplest version of possibilistic regression analysis, introduced by Tanaka et al. (see [69, 103, 104]). Given a database with crisp input and output, the aim of interval regression analysis is to predict the dependent variable via an interval by using the predictor variables. For this, the coefficients of the model used for the regression are also intervals. Each coefficient is expressed via its center and its radius.

  In the original model, a linear programming formulation is given to solve the problem, where the objective is minimizing the sum of radii of the predicted outputs, with the constraint that the real value of the dependent variable must be included in the predicted output (see [103]). Later, in [104], a quadratic formulation is given to include in the objective function a term to minimize the sum of the squared distances from the center of the predicted output to the real value of the dependent variable.

  Other improvements have been performed to study the role of outliers in the regression process. In [69], two regression models are built for each database by using quantile techniques, and two interval outputs are given for each observation, with the smallest one included in the biggest one. The first model is built with a given proportion of the data (this way, we can study the general behaviour of

the data, without containing outliers), whereas the second model is built with all the observations. Then, given a database, two intervals will be assigned as a prediction, and this can be seen as a trapezoidal fuzzy output.

Support Vector Machines have been applied to the problem in [69] to build the two models (see [63]) and to the general interval regression analysis case. In [65], an ϵ-SVR is solved (with $\epsilon = 0$) to obtain an initial crisp value of the output, which will be the center of an interval output with radius equal to a value ϵ, computed by using the obtained regression errors. This interval output is given as initial seed to two Radial Basis Function networks which identify the upper and lower sides of the output. In [64], the quadratic formulation of [104] is integrated with the standard ϵ-SVR approach.

- **Perturbed data:** Another interesting situation related to the regression model with imprecision is the case of data affected by some kind of noise or perturbation. A robust regressor must be constructed, insensitive to this noise or perturbation. In [107, 108], a Robust Support Vector Regression model has been studied, with noise in the input data (predictor variables). Although the data points are assumed to be uncertain or noisy, such perturbation is bounded by a given hypersphere of known radius. An optimization problem is formulated and solved via Second Order Cone Programming.

- **Missing values:** The case where there exist some missing values in the database can also be included in this methodology. Instead of imputing single values, as usual, for the missing coordinates, they can be replaced by intervals built by using the non-missing values in the dataset.

As in the classification case, our model generalizes the regression problems for the cases of imprecise data described above. Support Vector Regression applied to the interval-data case gets better results on benchmark datasets than those obtained in some previous works. Furthermore, we get to exploit the structure of the problem in the single-input and interval-output case, where non-linear regressors are introduced via the use of kernels.

1.4 Multiple Instance Learning

1.4.1 The Multiple Instance Problem

Consider a classification problem where each of the elements which compose the dataset may be represented by several feature vectors (called a *bag*), and only some of them (even only one in some cases) are responsible for the class that the element belongs to. For approaching this task, a set of tools grouped under the name of Multiple Instance Learning have been developed during the last decade.

A Multiple Instance Problem is a supervised classification problem where the elements to be classified are bags of instances which are vectors measuring d different attributes (see e.g. [38, 118, 121] for a description). According to these measurements, a label ($+1$ or -1 in the two-class case) must be assigned to each bag.

Although there are several ways to assign a label to the complete set of vectors (see [118]), in our problem, the most popular strategy is based on the so-called MI assumption (see [118]), which states that a bag is considered as positive (label +1) if at least one of its instances satisfies a determined condition and as negative (label -1) otherwise.

The Multi-Instance Problem can be seen as a special case of the classification problem with imprecision defined in Subsection 1.3.2. The difference with respect to the type of situation described in 1.3.2 lies in the cardinality of the elements of the database. In both models, the elements are sets in $\mathbb{R}^d$, but, whereas in a Multiple Instance Problem the elements are discrete sets (bags of instances), in the model described in 1.3.2 (interval data, perturbed data) the elements are continuous sets in $\mathbb{R}^d$ (boxes or spheres). In the former model, the cardinality of each element is finite, and in the latter model, the cardinality is infinite.

1.4.2 Multi-Instance Classification: origin of the problem and literature

Multi-Instance Learning was first applied to drug activity prediction by Dietterich et al. in [38]. In that problem, the bags are molecules, whereas the instances are different low-energy conformations, that is, shapes that the molecule can adopt by rotating its bonds. Only some of these conformations can bind to a target binding site (which is a part of a larger molecule) and then, the molecule becomes active, producing a determined drug. But the only information biochemists know is if a molecule is qualified or not to make a drug, not having any information about the correct conformations. The aim is thus, given a new molecule, to be able to predict if it will be active or inactive by taking into account the whole set of its conformations.

In [38], an axis-parallel hyper-rectangle is built such that it must contain at least one instance of each positive bag and it cannot contain any instance of the negative bags. Three algorithms are derived for this. The first algorithm constructs the smallest rectangle covering all the instances of the positive bags, and then it excludes all the negative instances, by eliminating, in turns, the negative instance which requires eliminating the smallest number of positive instances. The second algorithm is a modification of the first one, which assigns a cost for eliminating each positive instance (this is useful, for example, to avoid excluding from the rectangle a positive instance which is the last survivor from its bag). The third algorithm builds the smallest rectangle covering at least one instance of each positive bag, by using a backfitting procedure to choose the positive instances and by selecting the most relevant features in each step.

Apart from drug activity prediction, Multi-Instance Learning has been successfully applied to other different fields. In image classification or retrieval (see [2, 26, 27, 45, 77, 78, 89, 119]), one has a series of images, and the goal is to train a classifier to detect a given object. The pictures are the bags of the multi-instance problem, which are segmented in sets of pixels (blobs), the instances of the problem. An image is classified as positive if it contains at least a determined blob, which is characteristic from the object to detect, and as negative otherwise.

In [122], a problem in web mining is described, where the aim is to provide recommendation on web index pages to an user, based on the browsing history of that web user. The index pages (which are web pages with links to other pages, containing only the titles or brief information about the content of those linked pages) are the bags of the multi-instance problem, while each linked page is an instance. Each linked page is represented by its d more frequent terms. A user is interested in an index page if he or she is interested in at least one of the links (satisfying thus the MI assumption).

Other different applications have been studied in handwriting recognition (see [67]), text categorization (see [2]) or disease prediction (see [118]).

Since the appearance of [38], several authors have developed new algorithms to try to improve the results obtained by Dietterich et al., following different strategies.

- **Diverse Density:** In [77], the concept of *Diverse Density* is introduced to deal with multi-instance problems. A point with high Diverse Density is near to many instances of different positive bags, and far from the instances of the negative bags. The problem is thus to find the point with the maximum Diverse Density and, when having a new bag, it is classified as positive if the smallest distance from the bag to that point is smaller than a certain threshold, and as negative otherwise. In [120], an Expectation-Maximization algorithm is proposed to maximize the Diverse Density, by transforming the multi-instance problem into a

single instance and by using the EM algorithm to maximize the instance responsibility for the corresponding label of each bag. This same strategy is also used in [89].

- **k-Nearest Neighbour:** Two variants of the k-NN algorithm are proposed in [115] to solve multi-instance problems. The minimal and maximal Hausdorff distances are defined to measure the proximity between bags. A Bayesian framework is used for the so-called *Bayesian k-NN* algorithm. For the *Citation k-NN* algorithm, the concepts of references and citers are applied to the k-NN methodology, by studying for each bag not only its neighbours, but also the bags for which the concerned bag is a neighbour. These concepts are also used in [122].

- **Support Vector Machines:** In [2], two formulations as a mixed integer quadratic program are proposed. The *mi-SVM* approach introduces integer variables for modeling the individual labels of the instances of the positive bags, while the *MI-SVM* formulation selects one representative from each positive bag as that one which determines the sign of the label of its bag. Both algorithms compute optimal hyperplanes, given an initial assignment for the integer variables, and they update those integer values by assigning a positive label to the instance with the highest value for the classification rule.

  A separating hyperplane via SVMs is also proposed in [76], by using that a positive bag is correctly classified if at least one element of the convex hull of the instances of the bag is included in the positive halfspace. A set of bilinear constraints are derived and, in turns, one set of variables is held constant and the underlying linear program is solved. This successive linearization algorithm converges in a few iterations to a local optimum.

  In [32], an image categorization problem is solved by using SMVs and the k-means algorithm. Given an image, this is represented as a set of image patches, and, via the k-means clustering algorithm, the patch descriptors are assigned to a predetermined number of clusters. Every image is described via a feature vector, counting the number of its patches in each cluster. The *one-versus-rest* SVM algorithm is used to solve the problem of assigning each image to a cluster.

  Transductive SVMs, a modification of SVM which forces to unlabeled data (coming from positive bags) to be as far as possible from the separating hyperplane, are used in [14], obtaining good results when the positive bags are sparse (few positive instances).

- **Kernels:** Different kernels have been defined on this type of data. A general kernel on multi-instance data is defined in [50], which separates positive and negative bags under natural assumptions. The kernel procedure described in [67] consists in mapping the database to a Hilbert space via a first kernel, fitting a

Gaussian model to each bag in that feature space and defining a second kernel as the Bhattacharyya's affinity between such Gaussian models.

In [26], a similarity measurement is defined between a bag and an instance, and a mapping is defined in terms of that measure, transforming the multi-instance problem into a single-instance, solved via SVMs. A similar strategy is followed in [27], but using, to define the mapping, the maximum distance between a bag and a set of instances prototypes obtained via Diverse Density. Other kernels on sets of vectors can be found in [45].

Many other different techniques have been proposed. Among others, we may cite DC Optimization [29], Propositional Learning [48], Generalized Multi-Instance Learning [116], etc.

A different strategy to solve this kind of problems is described in this thesis, where the classification rule is defined in terms of a separating ball which maximizes the margin between the two groups to classify. An algorithm based on the necessary conditions of optimality is proposed. This type of solution is also useful to deal with problems in Location Theory, in particular, in Semi-Obnoxious Location. This way, our aim is to show the applicability of this type of classification techniques in another interesting area of Operations Research, as is the case of Location Theory.

1.5 Thesis overview

In this thesis, we develop several tools to solve classification and regression problems where the elements of the dataset are not single feature vectors, but sets in $\mathbb{R}^d$ with certain geometrical properties. The classifier or regressor is defined by following the strategy successfully used in Support Vector Machines of maximizing the margin. In each case, an optimization problem is formulated, which must be solved to find an optimal classifier or regressor.

Several types of optimization problems are derived for these problems: quadratic convex programs when we seek hyperplanes for classification or regression tasks (whose solution will be obtained directly by using a mathematical programming solver, such as CPLEX or LOQO [112]) or nonlinear and mixed-integer nonlinear programs when we seek separating hyperspheres (where we will develop exact or heuristic algorithms to derive an optimal solution).

In Chapter 2, we study the supervised classification problem with imprecise data, where the elements to be classified are sets with certain properties. In particular, this model can be applied to deal with data affected by some kind of noise and in the

case of interval-valued data. Two classification rules, a fuzzy one and a crisp one, are defined in terms of a separating hyperplane, and a formulation of the rule identification problem by margin maximization is introduced, extending the standard techniques in Support Vector Machines used for single feature vectors. These results are shown in our paper [16].

In Chapter 3, based on our work [17], the regression problem with imprecision on data is considered. The elements of the database are sets in $\mathbb{R}^d$, and the dependent variable is given by an interval. Interval data and data affected by some kind of noise or uncertainty are studied as two particular cases of our model. The proposed formulation is based on the standard ϵ-Support Vector Regression approach. In the interval-data case, two different formulations will be obtained, according to the way of measuring the distance between the prediction and the actual intervals: the maximum distance and the Hausdorff distance.

These methodologies described in Chapters 2 and 3 are also proved to be useful in practice when handling missing values in a database, by using imputation based on intervals. Furthermore, our models are proved to generalize the formulations given in [107, 108, 109] for classification and regression problems with data affected by some kind of perturbations, which are supposed to be unknown but bounded for a given norm.

The regression problem where uncertainty only affects to the dependent variable of the elements of the database is studied in Chapter 4, which is based on our reference [19]. A model based on the standard ϵ-Support Vector Regression approach is given, where two hyperplanes need to be constructed to predict the interval-valued dependent variable. By using the Hausdorff distance to measure the error between predicted and real intervals, a convex quadratic optimization problem is obtained.

Although this problem can be seen as a particular case of the formulation given in Chapter 3, the advantage of having this new model is to allow non-linear regressors to be introduced via the use of kernels structures. This way, more abstract relations between the data can be considered to construct an adequate regressor.

The Multiple Instance Classification Problem is the topic of Chapter 5. We consider a classification problem where the elements to be classified are bags of instances which are vectors measuring d different attributes. This model shows a different way of representing imprecision in data, since for each element of the database, only some of its instances are really responsible for the label assigned to the complete bag (and the rest of them can be seen as noise). However, we cannot apply directly the same kind of tools described in Chapter 2, since the elements in Multi-Instance Classification are discrete sets in $\mathbb{R}^d$, and the elements in the general classification model with imprecision are continuous sets.

The classification rule is defined in terms of a ball, whose center and radius are the parameters to be computed. Given a bag, it is assigned to the positive class if at least one element is strictly included inside the ball, and it is labelled as negative otherwise. We model this question as a margin optimization problem. Several necessary optimality conditions are derived leading to a polynomial algorithm in fixed dimension. A Variable Neighbourhood Search algorithm (see [81]) is proposed, based on the optimality conditions for the problem, to solve the model. This work is the basis of our reference [18].

Finally, as an application of the methodology developed for classification problems with imprecision, we study a location problem in Chapter 6, based on our work [51]. A semi-obnoxious facility must be located in the Euclidean plane to give service to a group of customers. Simultaneously, a set of populated areas, with shapes approximated via polygons, must be protected from the negative effects derived from that facility. The customers can be seen as the positive class, whose elements are points in $\mathbb{R}^2$, while the populated areas can be considered as the negative class, whose elements are continuous sets in $\mathbb{R}^2$, and a separating ball, similar to that constructed in Chapter 5, must be obtained.

The problem is formulated as a margin maximization model. Necessary optimality conditions are studied and a finite dominating set of solutions is obtained, leading to a polynomial time algorithm.

Computational experiments have been performed with benchmark and artificial datasets for every model, obtaining good results and showing that the tools we have developed are competitive.

Chapter 2

Support Vector Machines with imprecise data

Contents

2.1 Introduction

In this chapter, a classification problem is considered in which the elements to be classified are sets with certain geometrical properties. Support Vector Machines are extended to this type of data, and we study, as particular cases, the problems with data affected by some kind of perturbation and the interval-data case. This latter is studied in depth and several numerical experiments are performed. Moreover, this method is proved to be useful to deal with missing values by using imputation via intervals.

The chapter is structured as follows. In Section 2.2, our model is explained, the classification rule is defined and the corresponding optimization problem is derived. A general formulation is given, and it is particularized to the case of interval data and perturbed data. Afterwards, we focus on the case of interval data. In Section 2.3, an overview on multi-class classification problems is given, before solving a problem for a real interval-valued database via our technique (adapted to the multi-class case). In Sections 2.4 and 2.5, more computational experiments are performed, firstly for a database where data are considered imprecise and its elements have been transformed into intervals, and secondly by erasing at random some coordinates in a real database and by substituting the missing coordinates by intervals built with the remaining elements of the database. These intervals are based on different percentiles and on the mean and the deviation. Likewise, we perform another numerical experiment with a database with missing values, where the blanks are replaced by these intervals. We finish in Section 2.6 with some discussion and concluding remarks.

2.2 Modeling the problem

2.2.1 Defining the classification rule

In the standard approach to classification, described in Subsection 1.1.1, each instance in the database Ω is of the form $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$. Now we consider a database $\Omega \subset \mathbb{R}^d \times \mathbb{R}$ formed by elements $i = (X_i, Y_i) \in \Omega$, where Y_i is the corresponding class defined by means of a label +1 or -1 and X_i is the form $X_i = x_i + B_i$, with $x_i \in \mathbb{R}^d$ and with B_i a subset of $\mathbb{R}^d$ with certain geometrical properties, namely, it is convex, symmetric with respect to the origin and contains the origin. In other words, B_i is the unit ball of a symmetric gauge γ_i (see [59]), that is,

$$B_i \;=\; \{s \in \mathbb{R}^d : \gamma_i(s) \leq 1\}. \tag{2.1}$$

Two different particular cases are of main interest:

1. γ_i is given by

$$\gamma_i(s_1,\ldots,s_d) = \max_{j=1,\ldots,d} \frac{2|s_j|}{u_{ij}-l_{ij}}, \text{ for } l_{ij} < u_{ij},\ j=1,\ldots,d, \tag{2.2}$$

and then

$$\begin{aligned} B_i &= \{s \in \mathbb{R}^d : \gamma_i(s) \leq 1\} = \{s \in \mathbb{R}^d : \max_{j=1,\ldots,d} \frac{2|s_j|}{u_{ij}-l_{ij}} \leq 1\} \\ &= \{s \in \mathbb{R}^d : |s_j| \leq \frac{u_{ij}-l_{ij}}{2},\ \forall j = 1,\ldots,d\}. \end{aligned} \tag{2.3}$$

2. γ_i is given by

$$\gamma_i(s_1,\ldots,s_d) = \frac{1}{r_i}\sum_{j=1}^{d}(|s_j|^p)^{\frac{1}{p}}, \text{ for some } p,\ 1 \leq p \leq \infty, \text{ for } r_i > 0, \tag{2.4}$$

and then

$$\begin{aligned} B_i &= \{s \in \mathbb{R}^d : \gamma_i(s) \leq 1\} = \{s \in \mathbb{R}^d : \frac{1}{r_i}\sum_{j=1}^{d}(|s_j|^p)^{\frac{1}{p}} \leq 1\} \\ &= \{s \in \mathbb{R}^d : \|s\|_p \leq r_i\}. \end{aligned} \tag{2.5}$$

In case of γ_i as defined in (2.2), taking x_i such that $x_{ij} = \frac{l_{ij}+u_{ij}}{2}$, $j = 1,\ldots,d$, one has that $X_i = x_i + B_i$, with B_i as in (2.3), and thus is a Cartesian product of intervals, that is, $X_i = \prod_{j=1}^{d}[l_{ij}, u_{ij}]$. Interval data can be used, for example, when the data have been aggregated or summarized in an interval (in the Symbolic Data Analysis framework, see e.g.[11, 13, 43]) or there exist missing values in our database and the missing coordinates are replaced by intervals built with the remaining instances of the same group.

The second case, with γ_i of the form (2.4), is interesting to model the case of data affected by some kind of noise. This is the idea of the so-called Robust SVMs, which were introduced in [108, 109] to classifiers, via Support Vector Machines, when the data have suffered some perturbations. These perturbations are supposed to be unknown, but a bound of them is known, for a given norm in the input space.

In that case, we can write $X_i = x_i + B_i$, with x_i the original value of the instance and B_i, as defined in (2.5), a ball representing the unknown perturbation and r_i being a positive constant which bounds the perturbation in p-norm, since $x \in X_i$ iff $x = x_i + s$, with $\gamma_i(s) \leq 1$, or equivalently, $\|s\|_p \leq r_i$, for each $i \in \Omega$.

The classification rule is defined in terms of a separating hyperplane, whose parameters $\omega \in \mathbb{R}^d$, $\beta \in \mathbb{R}$ must be computed, as done in the standard SVMs approach (see Section 1.1). Once ω, β are determined, elements are classified according to a rule, which extends naturally the one in which each element is a singleton. Two variants are considered:

Crisp classification :

Given an element $X \subset \mathbb{R}^d$, classify X in G_+ if $\max_{x \in X}(\omega^\top x + \beta) > -\min_{x \in X}(\omega^\top x + \beta)$, and in G_- otherwise.

Fuzzy classification :

Given an element $X \subset \mathbb{R}^d$, compute $I(X) := [\min_{x \in X}(\omega^\top x + \beta), \max_{x \in X}(\omega^\top x + \beta)]$,

- if $0 \notin I(X)$,
  - classify in G_+ (with intensity equal to 1), if $\min_{x \in X}(\omega^\top x + \beta) > 0$,
  - classify in G_- (with intensity equal to 1), if $\max_{x \in X}(\omega^\top x + \beta) < 0$, (2.6)
- if $0 \in I(X)$,
  - classify in G_+, with intensity $\dfrac{\max_{x \in X}(\omega^\top x + \beta)}{\max_{x \in X}(\omega^\top x + \beta) - \min_{x \in X}(\omega^\top x + \beta)}$,
  - classify in G_-, with intensity $\dfrac{-\min_{x \in X}(\omega^\top x + \beta)}{\max_{x \in X}(\omega^\top x + \beta) - \min_{x \in X}(\omega^\top x + \beta)}$. (2.7)

We can simplify the fuzzy rule via the following *clamp* function. Given three numbers x, y and z, we define the *clamp* function for these three values as $clamp(x, y, z) = \min(\max(x, y), z)$.

Then, given an element X, the intensity for classifying it in G_+ is

$$intensity_+ := clamp(0, fuzzy\ value, 1), \tag{2.8}$$

where

$$fuzzy\ value := \frac{\max_{x \in X}(\omega^\top x + \beta)}{\max_{x \in X}(\omega^\top x + \beta) - \min_{x \in X}(\omega^\top x + \beta)}. \tag{2.9}$$

Alternatively, we classify X in G_- with $intensity_- := 1 - intensity_+$.

Thus the fuzzy rule allows for a degree of uncertainty in the classification, while the crisp rule does not.

Observe that the crisp rule simply classifies the element in the group with the highest fuzzy intensity, since the crisp rule includes expression (2.6).

Figure 2.1 shows an example in dimension 2 of the problem with boxes (Cartesian product of intervals). The white boxes represent the elements of one group, whereas

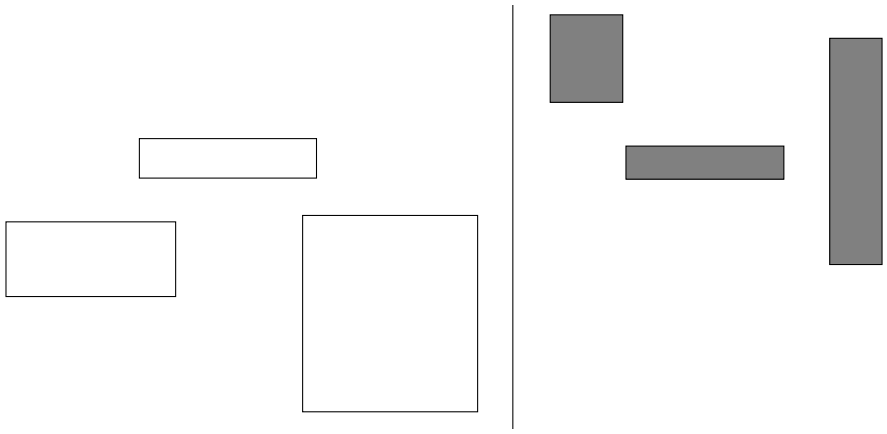

Figure 2.1: A separating hyperplane

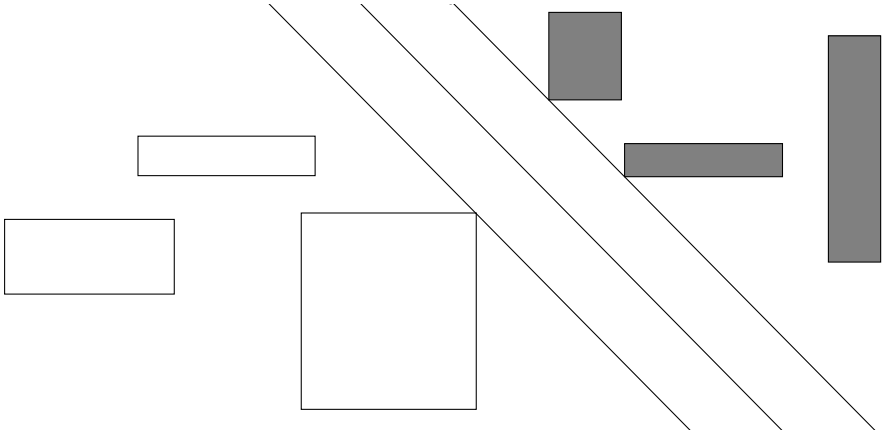

Figure 2.2: Maximizing the margin

the grey boxes represent the elements of the other group. Our problem is to build a hyperplane separating the two sets of boxes. Observe that when the two sets are linearly separable (as in Figure 2.1), that is, when a hyperplane can be constructed which separates strictly the two sets, the fuzzy part of the classification rule (expression (2.7)) is not necessary.

Different hyperplanes separating the two groups may exist. In order to choose the hyperplane, we maximize a margin (see Figure 2.2) as performed in Support Vector Machines (see Section 1.1).

Then, given a training sample $I = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)\} \subseteq \Omega$, we will solve the optimization problem to obtain optimal parameters ω and β (optimal in the sense that the margin is maximized) for constructing the classification rule.

2.2.2 The optimization problem

Let $\|\cdot\|$ be a norm in $\mathbb{R}^d$. Let us first consider the separable case, i.e., let us assume first that ω, β exist such that the two groups can be separated via a hyperplane $H = \{x \in \mathbb{R}^d : \ \omega^\top x + \beta = 0\}$.

According to the linear classification rule defined in (2.6), given $X_i \in I$, it is assigned to the group

$$\begin{array}{lll} G_+ & : & \text{if } \min\limits_{x\in X_i} (\omega^\top x + \beta) > 0, \\ G_- & : & \text{if } \max\limits_{x\in X_i} (\omega^\top x + \beta) < 0, \quad \text{or equivalently,} \quad \text{if } \min\limits_{x\in X_i} -(\omega^\top x + \beta) > 0. \end{array}$$

The distance from a point $x^* \in \mathbb{R}^d$ to a hyperplane $H = \{x \in \mathbb{R}^d : \ \omega^\top x + \beta = 0\}$ is given in expression (1.2). Hence, given the training sample I, the optimization problem to solve is

$$\max_{\omega,\beta} \ \min \left\{ \min_{i\in G_+} \ \min_{x\in X_i} \ \frac{\omega^\top x + \beta}{\|\omega\|^0}, \min_{k\in G_-} \ \min_{x\in X_k} \ \frac{-(\omega^\top x + \beta)}{\|\omega\|^0} \right\}, \tag{2.10}$$

where $\|\cdot\|^0$ represents the dual norm of $\|\cdot\|$.

The objective in (2.10) is homogeneous in its variables. Hence, one can assume without loss of generality that

$$\min \left\{ \min_{i\in G_+} \ \min_{x\in X_i} \ (\omega^\top x + \beta), \min_{k\in G_-} \ \min_{x\in X_k} \ -(\omega^\top x + \beta) \right\} \geq 1,$$

and the problem can be expressed as

$$\begin{array}{ll} \max\limits_{\omega,\beta} & \dfrac{1}{\|\omega\|^0} \\ \text{s.t.} & \min\limits_{x\in X_i} \ \omega^\top x + \beta \geq 1, \ \forall i \in G_+ \\ & \min\limits_{x\in X_k} \ -(\omega^\top x + \beta) \geq 1, \ \forall k \in G_-. \end{array} \tag{2.11}$$

Taking as $\|\cdot\|$ the Euclidean norm, the problem is equivalent to

$$\begin{array}{ll} \min\limits_{\omega,\beta} & \dfrac{1}{2} \ \omega^\top \omega \\ \text{s.t.} & \min\limits_{x\in X_i} \ \omega^\top x + \beta \geq 1, \ \forall i \in G_+ \\ & \min\limits_{x\in X_k} \ -(\omega^\top x + \beta) \geq 1, \ \forall k \in G_-. \end{array} \tag{2.12}$$

A test to study if there exists a hyperplane separating the two groups will be given later, in Theorem 2.2. Anyway, Problem (2.12) can be adapted to the one in which the

two groups are not linearly separable. Indeed, in case of non-separability, we introduce one slack variable per element, and we penalize the objective function,

$$\begin{array}{ll} \min\limits_{\omega,\beta,\xi,\eta} & \frac{1}{2}\,\omega^\top\omega + \frac{C_+}{n}\sum\limits_{i\in G_+}\xi_i + \frac{C_-}{m}\sum\limits_{k\in G_-}\eta_k \\ \text{s.t.} & \min\limits_{x\in X_i}\ \omega^\top x+\beta \geq 1-\xi_i,\ \forall i\in G_+ \\ & \min\limits_{x\in X_k}\ -(\omega^\top x+\beta) \geq 1-\eta_k,\ \forall k\in G_- \\ & \xi_i,\eta_k\geq 0,\ \forall i\in G_+,\ \forall k\in G_-, \end{array} \tag{2.13}$$

where we denote by n the cardinal of G_+ and by m the cardinal of G_-, and C_+, C_- are constants.

2.2.3 Obtaining an equivalent formulation

Problem (2.13) has a convex quadratic objective and nonlinear constraints. In the following result, we give an equivalent formulation of Problem (2.13) by building the dual of the problems appearing in the constraints of Problem (2.13). Recall that the dual gauge of γ_i in ω is defined by $\gamma_i^0(\omega) = \max\limits_{\gamma_i(u)\leq 1}\ (\omega^\top u)$.

Theorem 2.1 *Problem (2.13) admits the following equivalent formulation,*

$$\begin{array}{ll} \min\limits_{\omega,\beta,\xi,\eta} & \frac{1}{2}\sum\limits_{j=1}^{d}\omega_j^2 + \frac{C_+}{n}\sum\limits_{i\in G_+}\xi_i + \frac{C_-}{m}\sum\limits_{k\in G_-}\eta_k \\ s.t. & \omega^\top x_i - \gamma_i^0(\omega)+\beta \geq 1-\xi_i,\ \forall i\in G_+ \\ & -\omega^\top x_k - \gamma_k^0(\omega)-\beta \geq 1-\eta_k,\ \forall k\in G_- \\ & \xi_i,\eta_k\geq 0,\ \forall i\in G_+,\ \forall k\in G_-, \end{array} \tag{2.14}$$

where γ_i is the gauge associated to the element $i\in I$ and γ_i^0 is its dual gauge.

Proof.
To prove the result, we change the constraints in Problem (2.13) by using that $X_i = x_i + B_i$, with B_i the unit ball induced by gauge γ_i for each X_i, and by changing to its dual gauge.

One has that

$$\min_{x\in x_i+B_i}\omega^\top x = \min_{\gamma_i(u)\leq 1}\omega^\top(x_i+u) = \omega^\top x_i + \min_{\gamma_i(u)\leq 1}\omega^\top u = \omega^\top x_i - \max_{\gamma_i(u)\leq 1}(-\omega^\top u).$$

By using that $\gamma_i^0(-\omega) = \max\limits_{\gamma_i(u)\leq 1}(-\omega^\top u)$ (with γ_i^0 the dual gauge of γ_i), and since $\gamma_i^0(-\omega)=\gamma_i^0(\omega)$, one obtains that

$$\min_{x\in x_i+B_i}\omega^\top x = \omega^\top x_i - \gamma_i^0(-\omega) = \omega^\top x_i - \gamma_i^0(\omega). \tag{2.15}$$

Then, by using (2.15), one has that the set of constraints for G_+ in Problem (2.13),

$$\min_{x \in x_i + B_i} \omega^\top x + \beta \geq 1 - \xi_i,$$

is equivalent to

$$\omega^\top x_i - \gamma_i^0(\omega) + \beta \geq 1 - \xi_i,$$

for every $i \in G_+$.

We proceed analogously for the set of constraints for the group G_- in Problem (2.13),

$$\min_{x \in x_k + B_k} -(\omega^\top x + \beta) \geq 1 - \eta_k, \tag{2.16}$$

and by using (2.15), we obtain that (2.16) is equivalent to

$$-\omega^\top x_k - \gamma_k^0(-\omega) - \beta \geq 1 - \eta_k, \text{ i.e. } , -\omega^\top x_k - \gamma_k^0(\omega) - \beta \geq 1 - \eta_k,$$

for every $k \in G_-$, since γ_k^0 is a symmetric gauge. □

Below, we consider the two cases of interest for the two definitions of γ_i in (2.2) and (2.4). The first one is the particular case in which the elements of the database are boxes, that is, $X_i = \prod_{j=1}^d [l_{ij}, u_{ij}]$, for every $i \in I$.

Corollary 2.1 *Let γ_i be the gauge defined in (2.2). Then, Problem (2.13) admits the following equivalent formulation as a convex quadratic problem*

$$\begin{array}{ll} \min\limits_{\sigma,\tau,\beta,\xi,\eta} & \frac{1}{2}\sum\limits_{j=1}^d (\sigma_j - \tau_j)^2 + \frac{C_+}{n}\sum\limits_{i \in G_+} \xi_i + \frac{C_-}{m}\sum\limits_{k \in G_-} \eta_k \\ s.t. & \sum\limits_{j=1}^d \sigma_j l_{ij} - \sum\limits_{j=1}^d \tau_j u_{ij} + \beta \geq 1 - \xi_i, \ \forall i \in G_+ \\ & \sum\limits_{j=1}^d \tau_j l_{kj} - \sum\limits_{j=1}^d \sigma_j u_{kj} - \beta \geq 1 - \eta_k, \ \forall k \in G_- \\ & \xi_i, \eta_k, \sigma_j, \tau_j \geq 0, \ \forall i \in G_+, \ \forall k \in G_-, \ j = 1, \ldots, d. \end{array} \tag{2.17}$$

Proof.
Firstly, observe that, if $\gamma_i(s) = \max\limits_{j=1,\ldots,d} \dfrac{2|s_j|}{u_{ij} - l_{ij}}$, then its dual gauge is

$$\gamma_i^0(s) = \sum_{j=1}^d \frac{u_{ij} - l_{ij}}{2}|s_j|. \tag{2.18}$$

And now, it is sufficient to replace the particular values of x_i and $\gamma_i^0(\omega)$ in the constraints of Problem (2.14).

In the first set of constraints,

$$\omega^\top x_i - \gamma_i^0(\omega) + \beta \geq 1 - \xi_i, \ \forall i \in G_+,$$

we replace $x_{ij} = \dfrac{l_{ij} + u_{ij}}{2}$, $j = 1, \ldots, d$ and $\gamma_i^0(\omega) = \displaystyle\sum_{j=1}^{d} |\omega_j| \frac{u_{ij} - l_{ij}}{2}$, and we obtain the constraint

$$\sum_{j=1}^{d} \omega_j (\frac{l_{ij} + u_{ij}}{2}) - \sum_{j=1}^{d} |\omega_j| (\frac{u_{ij} - l_{ij}}{2}) + \beta \geq 1 - \xi_i, \ \forall i \in G_+.$$

Let us define $\sigma_j = \max\{0, \omega_j\}$ and $\tau_j = \max\{0, -\omega_j\}$, for $j = 1, \ldots, d$. One has that $\omega_j = \sigma_j - \tau_j$ and $|\omega_j| = \sigma_j + \tau_j$, and the set of constraints can be written as

$$\sum_{j=1}^{d} [(\sigma_j - \tau_j)(\frac{l_{ij} + u_{ij}}{2}) - (\sigma_j + \tau_j)(\frac{u_{ij} - l_{ij}}{2})] + \beta \geq 1 - \xi_i, \ \forall i \in G_+.$$

After some calculations, we obtain the first set of constraints in Problem (2.17).

Now, we must proceed analogously for the set of constraints

$$-\omega^\top x_k - \gamma_k^0(\omega) - \beta \geq 1 - \eta_k, \ \forall k \in G_-.$$

Replacing the values of x_k and $\gamma_k^0(\omega)$ in these constraints, and by introducing the variables σ and τ, one obtains

$$-\sum_{j=1}^{d} [(\sigma_j - \tau_j)(\frac{l_{kj} + u_{kj}}{2}) + (\sigma_j + \tau_j)(\frac{u_{kj} - l_{kj}}{2})] - \beta \geq 1 - \eta_k, \ \forall k \in G_-,$$

which, after arranging terms, leads to the second set of constraints in Problem (2.17).

□

Remark 2.1 *When we defined γ_i in (2.2), we assumed that $l_{ij} < u_{ij}$, $\forall j = 1, \ldots, d$. In the case of degenerate boxes (that is, $l_{ij} = u_{ij}$ for some coordinates), denote by J_F the set of indexes with $l_{ij} = u_{ij}$ and denote by J_V the set of indexes with $l_{ij} < u_{ij}$. Let us define γ_i as*

$$\gamma_i(s_1, \ldots, s_d) = \begin{cases} \max\limits_{j \in J_V} \dfrac{2|s_j|}{u_{ij} - l_{ij}}, & \text{if } s_j = 0, \ \forall j \in J_F \\ +\infty, & \text{otherwise.} \end{cases}$$

One has that $\gamma_i^0(s)$ has the same form as (2.18) and then, formulation (2.17) remains valid.

The problem of interval data will be studied more deeply in the following sections.

The second case of interest is when the data are affected by some kind of perturbations. Then, as a straightforward consequence of Theorem 2.1, we obtain the result previously derived by [108, 109]:

Corollary 2.2 *Let γ_i be the gauge defined in (2.4). Then, Problem (2.13) can be written as follows,*

$$\begin{array}{ll} \min\limits_{\omega,\beta,\xi} & \frac{1}{2}\sum\limits_{j=1}^{d}\omega_j^2 + \frac{C_+}{n}\sum\limits_{i\in G_+}\xi_i + \frac{C_-}{m}\sum\limits_{k\in G_-}\eta_k \\ s.t. & \omega^\top x_i - r_i\|\omega\|_q + \beta \geq 1-\xi_i,\ \forall i \in G_+ \\ & -\omega^\top x_k - r_k\|\omega\|_q - \beta \geq 1-\eta_k,\ \forall k \in G_- \\ & \xi_i, \eta_k \geq 0,\ \forall i \in G_+,\ \forall k \in G_-, \end{array} \tag{2.19}$$

where $\|\cdot\|_q$ is the dual norm of $\|\cdot\|_p$, i.e., $\frac{1}{p}+\frac{1}{q}=1$.

Proof.
It is sufficient to observe that, if $\gamma_i = \frac{1}{r_i}\|\cdot\|_p$, then its dual gauge is $\gamma_i^0 = r_i\|\cdot\|_q$, with $\|\cdot\|_q$ the dual norm of $\|\cdot\|_p$, p and q satisfying that $\frac{1}{p}+\frac{1}{q}=1$.

□

Problem (2.19) is equivalent to the formulation given in [108, 109], which was obtained by building the robust counterpart of the problem (by following robust optimization methods, [6, 7]). Hence, formulation (2.13) for any kind of gauge γ_i is more general than that obtained for robust SVMs.

2.2.4 Test of linear separability

From now on, we will only consider the case of interval data. Given an interval-valued database, a test of linear separability can be applied to the dataset by solving the linear program proposed in the following result.

Theorem 2.2 *Given $I \subseteq \Omega$ a training sample of interval data, that is, $X_i = \prod_{j=1}^{d}[l_{ij}, u_{ij}]$, for every $i \in I$, there exists a hyperplane separating the two groups, G_+ and G_-, iff*

the problem

$$\begin{array}{ll} \max\limits_{\lambda,\mu} & \sum\limits_{i\in G_+}\lambda_i + \sum\limits_{k\in G_-}\mu_k \\ s.t. & \sum\limits_{i\in G_+}\lambda_i = \sum\limits_{k\in G_-}\mu_k \\ & \sum\limits_{i\in G_+}\lambda_i l_{ij} - \sum\limits_{k\in G_-}\mu_k u_{kj} \leq 0,\ j=1,\ldots,d \\ & \sum\limits_{k\in G_-}\mu_k l_{kj} - \sum\limits_{i\in G_+}\lambda_i u_{ij} \leq 0,\ j=1,\ldots,d \\ & \lambda_i,\mu_k \geq 0,\ \forall i\in G_+,\ \forall k\in G_- \end{array} \tag{2.20}$$

is feasible with optimal solution equal to 0.

Proof.
Given the training sample I, according to the classification rule (2.6), the two groups G_+ and G_- are linearly separable, that is, there exist ω, β, the parameters of a hyperplane separating the two groups, iff

$$\begin{array}{ll} \min\limits_{x\in X_i}(\omega^\top x+\beta) > 0, & \forall i\in G_+ \\ \max\limits_{x\in X_k}(\omega^\top x+\beta) < 0, & \forall k\in G_-. \end{array} \tag{2.21}$$

By homogeneity, expression (2.21) is equivalent to say that ω, β exist with

$$\begin{array}{ll} \min\limits_{x\in X_i}\omega^\top x+\beta \geq 1, & \forall i\in G_+ \\ \min\limits_{x\in X_k}(-\omega)^\top x-\beta \geq 1, & \forall k\in G_-. \end{array} \tag{2.22}$$

Now, by using expression (2.15) and with an analogous reasoning to that used in the proof of Theorem 2.1, we can rewrite (2.22) as

$$\begin{array}{l} \omega^\top x_i - \gamma_i^0(\omega)+\beta \geq 1,\ \forall i\in G_+ \\ -\omega^\top x_k - \gamma_k^0(\omega)-\beta \geq 1,\ \forall k\in G_-, \end{array} \tag{2.23}$$

and since our database is interval-valued, we consider γ_i the gauge defined in 2.2, we rewrite expression (2.23) and we obtain that the two groups G_+ and G_- are linearly separable iff the problem

$$\begin{array}{ll} \min\limits_{\sigma,\tau,\beta} & 0^\top\sigma + 0^\top\tau \\ \text{s.t.} & \sum\limits_{j=1}^{d}\sigma_j l_{ij} - \sum\limits_{j=1}^{d}\tau_j u_{ij} + \beta \geq 1,\ \forall i\in G_+ \\ & \sum\limits_{j=1}^{d}\tau_j l_{kj} - \sum\limits_{j=1}^{d}\sigma_j u_{kj} - \beta \geq 1,\ \forall k\in G_- \\ & \sigma_j,\tau_j \geq 0,\ j=1,\ldots,d,\ \beta \text{ s.r.} \end{array} \tag{2.24}$$

is feasible with optimal solution equal to 0.

And, by using duality properties of linear programming, one can state that the two groups are linearly separable iff the dual of problem (2.24) is feasible with optimal solution equal to 0. But the dual problem is just the one formulated in (2.20), and the result follows.

□

2.3 A multi-class classification experiment with interval data

2.3.1 The multi-class classification problem

In the previous sections, we have discussed the classification problem for two groups with interval data. However, in many real situations, classification problems arise with data belonging to more than two groups. Solving a multi-class classification problem (see [58, 98, 114]) is, in general, a more difficult task than solving a two-class classification problem. Different strategies have been proposed. Most of these suggest to transform the multi-class problem in a series of two-class problems to be solved (see e.g., [49, 56, 62, 93, 106, 117]).

In this section, we solve a classification problem with interval data belonging to four different groups. Our methodology for classification has been applied to the 'car' dataset, which is a database with 33 car models described by 8 interval variables (explaining the following characteristics of each car model: *price, engine capacity, top speed, acceleration, step, length, width and height*) and one nominal variable which represents one of the four following possible categories: *utilitarian* (U), *berlina* (B), *sportive* (S) or *luxury* (L) (see [43] for more details).

Then, we have a database with four different groups (multi-class classification problem), where the data are Cartesian products of intervals in dimension 8. We have performed several computational experiments, by considering three possible techniques for multi-class classification (1-v-r, 1-v-1 and DDAG).

In order to measure the probability of misclassification, we have used the leave-one-out (LOO) strategy (see e.g. [53, 66]), that is, in turns, we consider only one element in the test sample, we train the model with the remaining elements and we test this model with the unitary test sample. We repeat the process for every element of the database.

Before showing the results of the experiment, the multi-class classification techniques used in the experiments are explained.

One-versus-rest (1-v-r) : N classifiers are constructed, the l-th classifier is the result of solving the corresponding problem (2.17) where the elements of the l-th group have a label +1 associated and the rest of elements have a label -1 associated.

For every element X of the database, we consider the problem where this element is the only one belonging to the test sample and the rest belong to the training sample and we build the corresponding $N = 4$ classifiers (for the l-th problem, G_+ is formed by all the elements of the l-th group). Once ω_l, β_l (the l-th classifier) are obtained, we compute, for X in the test sample, the values $\min_{x\in X}(\omega_l^\top x+\beta_l)$ and $\max_{x\in X}(\omega_l^\top x + \beta_l)$, and we obtain the intensity of classifying X in the l-th group via (2.8)-(2.9).

We assign X to the group with the highest output of the intensity. In case of tie, X is assigned to the group l whose $\min_{x\in X}(\omega_l^\top x + \beta_l)$ is the highest value.

One-versus-one (1-v-1) : In this case, we build a classifier for every possible pair of groups l and h, with $l < h$. In total, we need to construct $N(N-1)/2$ classifiers.

Given the test sample with a single element X, we compute every classifier separating two groups l and h. Once ω_{lh}, β_{lh} are obtained, we give one vote to group l if the intensity is higher for l than for h (following the same reasoning as for the 1-v-r case), or equivalently, if $\max_{x\in X}(\omega_{lh}^\top x + \beta_{lh}) \geq -\min_{x\in X}(\omega_{lh}^\top x + \beta_{lh})$. And X is assigned to the group with the highest number of votes (following the Max Wins algorithm, [49]).

In case of tie between two groups l and h, we go back to the classifier separating those groups and we assign to the group with the highest output for the intensity. We only need to build $N(N-1)/2$ comparisons since, for two groups l and h, with $l \neq h$, one obtains that $\omega_{lh} = -\omega_{hl}$ and $\beta_{lh} = -\beta_{hl}$, that is, the classifier is the same (the parameters are opposite).

Decision Directed Acyclic Graphs (DDAG) : This method, proposed in [93], is a modification of the standard 1-v-1 method, which is introduced to reduce the number of evaluations to be performed (only $N-1$ evaluations must be computed instead of $N(N-1)/2$ in the 1-v-1 method). Furthermore, the advantage of choosing this method, instead of 1-v-1, when using leave-one-out is also that, since the test sample is composed by only one element, only $N-1$ classifiers must be built. We describe below the method when using leave-one-out.

A Directed Acyclic Graph, DAG for short, is a graph with oriented edges and without cycles. In each node of this DAG, one constructs the classifier for two groups l and h and, given an element of the dataset, we assign it to the group with the highest output for the intensity, after applying the classifier. If we assign the l-th group, we can eliminate the label h as a candidate group for this element,

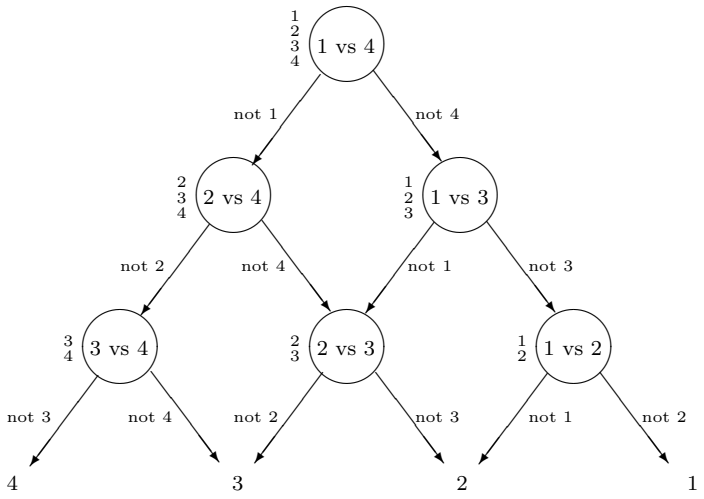

Figure 2.3: Directed Acyclic Graph

we pass to the next node to compare the group l against another one. The label at the end of this process (after $N-1$ comparisons) will be the assigned group.

An ordering must be imposed to decide which comparisons must be done in each step, but the selection of this ordering is arbitrary. In our case, we follow the DAG depicted in Figure 2.3 (several experiments done for another ordering yielded very similar results, and in [93] it is said that different orderings did not yield significant changes in terms of accuracy).

We start with the complete list of groups and we build the classifier for the groups 1 and 4. Depending on the assigned group, we take the left or right edge and consequently we eliminate one group. In each node, we build the classifier for the first and the last groups in the new list of groups. And we continue until the end of the graph, where the label is finally assigned.

2.3.2 Numerical results

All the computational experiments of this chapter have been implemented by using AMPL as the modeling language and have been solved via CPLEX or via LOQO, [112], (by using the NEOS server, [83]).

We have solved the classification problem (via leave-one-out) with the three multi-class techniques (1vr, 1v1, DDAG), for several values of the constants, $C_+, C_- \in \{0.001,\ 0.01,\ 0.1,\ 1,\ 10,\ 100,\ 1000\}$ and the results are given in the following tables. In Table 2.1, we show how many elements from the database of cars are misclassified, that is, how many are given a label different from its own category.

Likewise, we have studied the behaviour of the classifier via a resubstitution strategy

(see [35]), that is, when we use the complete set of instances as training sample to build the classifier (no test sample is used) and we apply it later to assign, in turn, a label to each instance. This way, we can study the quality of the classifier as a separator of the database. It is only done in the 1vr and 1v1 method (not in the DDAG method, since this is basically a 1v1 method where not all the comparisons between groups must be done).

In general, the best results for this dataset via leave-one-out and for different values of the constants have been obtained via the 1vr method. In fact, the worst number of misclassified elements in this case is equal to eight (which is quite good in comparison with the other two techniques). The lowest number of misclassified elements for leave-one-out and for resubstitution, through the three multi-class techniques, are shown in the tables in bold. For the 1vr method, there are different combinations of C_+ and C_- which give only four misclassified elements for leave-one-out and only two misclassified elements for resubstitution. For high values of C_+ and C_-, one obtains good values of well-classified elements, nevertheless, although biggest values of the constants were considered, the results could not be improved more.

On the other hand, although we obtain some high numbers of misclassified elements for the 1v1 method (it is especially remarkable the case when $C_+ = 0.001$ or $C_- = 0.001$), the lowest numbers of misclassified elements are obtained with this method. For $C_+ = C_- = 0.1$, we obtain only three misclassified elements via leave-one-out, and we got only one misclassified element in the training sample for several combinations of the constants. The results obtained via DDAG are quite similar to those obtained via 1v1.

In Tables 2.2-2.4, we show the detailed results for accuracy with each technique, for every combination of C_+ and C_-. The rows of each cell represent the original category of the car and the columns represent the label which has been assigned to that car model through leave-one-out. In bold, we show again the best results for the accuracy in the classification. We can observe that, in general, it is easier to distinguish the utilitarian and the sportive cars, and most of the difficulties arise while trying to discriminate between berlina and luxury cars.

	C_-	0.001		0.01		0.1		1		10		100		1000	
C_+		loo	rs	loo	rs	loo	rs	loo	rs	loo	rs	loo	rs	loo	rs
0.001	1vr	8	6	**4**	4	7	4	7	4	7	4	7	4	7	5
	1v1	6	5	17	17	17	17	17	17	17	17	17	17	17	18
	ddag	6	-	17	-	17	-	17	-	17	-	17	-	17	-
0.01	1vr	7	5	5	5	5	3	6	4	6	3	6	3	6	3
	1v1	13	11	7	4	7	5	7	5	7	5	7	5	7	5
	ddag	11	-	7	-	7	-	7	-	7	-	7	-	7	-
0.1	1vr	7	5	5	4	**4**	4	7	5	6	3	6	4	6	4
	1v1	11	9	7	6	**3**	3	4	2	4	2	4	2	4	2
	ddag	11	-	8	-	**4**	-	**4**	-	**4**	-	**4**	-	**4**	-
1	1vr	7	5	5	4	5	4	**4**	3	8	5	7	3	7	3
	1v1	11	9	8	6	5	4	5	2	5	2	5	2	5	2
	ddag	11	-	8	-	6	-	5	-	5	-	5	-	5	-
10	1vr	7	5	5	4	5	4	**4**	**2**	**4**	**2**	8	4	7	3
	1v1	11	9	8	6	6	4	7	4	6	2	6	**1**	6	**1**
	ddag	11	-	8	-	6	-	7	-	6	-	6	-	6	-
100	1vr	7	5	5	4	5	4	**4**	**2**	5	**2**	6	**2**	8	4
	1v1	11	9	8	6	6	4	7	4	8	2	6	**1**	6	**1**
	ddag	9	-	8	-	6	-	7	-	8	-	6	-	6	-
1000	1vr	7	5	5	3	5	4	**4**	**2**	5	**2**	7	4	6	**2**
	1v1	12	9	8	6	6	4	7	4	8	2	8	**1**	6	**1**
	ddag	11	-	8	-	6	-	7	-	8	-	8	-	6	-

Table 2.1: Misclassified elements for the 'car dataset' (loo: leave-one-out, rs: resubstitution)

1vr	C_-	0.001				0.01				0.1				1				10				100				1000			
C_+	Cars	U	B	L	S	U	B	L	S	U	B	L	S	U	B	L	S	U	B	L	S	U	B	L	S	U	B	L	S
0.001	U	7	3	0	0	**10**	**0**	**0**	**0**	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0
	B	1	6	1	0	**1**	**5**	**2**	**0**	1	6	1	0	1	6	1	0	1	6	1	0	1	6	1	0	1	6	1	0
	L	0	3	5	0	**0**	**0**	**8**	**0**	0	3	5	0	0	3	5	0	0	3	5	0	0	3	5	0	0	3	5	0
	S	0	0	0	7	**0**	**1**	**0**	**6**	0	1	0	6	0	1	0	6	0	1	0	6	0	1	0	6	0	1	0	6
0.01	U	9	1	0	0	10	0	0	0	10	0	0	0	10	0	0	0	10	0	0	0	10	0	0	0	10	0	0	0
	B	1	6	1	0	1	6	1	0	1	5	2	0	1	6	1	0	1	6	1	0	1	6	1	0	1	6	1	0
	L	0	3	5	0	0	3	5	0	0	1	7	0	0	3	5	0	0	3	5	0	0	3	5	0	0	3	5	0
	S	0	1	0	6	0	0	0	7	0	1	0	6	0	1	0	6	0	1	0	6	0	1	0	6	0	1	0	6
0.1	U	9	1	0	0	9	1	0	0	**10**	**0**	**0**	**0**	10	0	0	0	10	0	0	0	10	0	0	0	10	0	0	0
	B	1	6	1	0	1	7	0	0	**1**	**5**	**2**	**0**	1	5	2	0	1	6	1	0	1	6	1	0	1	6	1	0
	L	0	3	5	0	0	3	5	0	**0**	**1**	**7**	**0**	0	3	5	0	0	3	5	0	0	3	5	0	0	3	5	0
	S	0	1	0	6	0	0	0	7	**0**	**0**	**0**	**7**	0	0	1	6	0	1	0	6	0	1	0	6	0	1	0	6
1	U	9	1	0	0	9	1	0	0	10	0	0	0	**10**	**0**	**0**	**0**	10	0	0	0	10	0	0	0	10	0	0	0
	B	1	6	1	0	1	7	0	0	1	6	1	0	**1**	**5**	**2**	**0**	1	4	3	0	1	5	2	0	1	5	2	0
	L	0	3	5	0	0	3	5	0	0	3	5	0	**0**	**1**	**7**	**0**	0	3	5	0	0	3	5	0	0	3	5	0
	S	0	1	0	6	0	0	0	7	0	0	0	7	**0**	**0**	**0**	**7**	0	1	0	6	0	1	0	6	0	1	0	6
10	U	9	1	0	0	9	1	0	0	10	0	0	0	**10**	**0**	**0**	**0**	**10**	**0**	**0**	**0**	10	0	0	0	10	0	0	0
	B	1	6	1	0	1	7	0	0	1	6	1	0	**1**	**6**	**1**	**0**	**1**	**6**	**1**	**0**	1	4	3	0	1	5	2	0
	L	0	3	5	0	0	3	5	0	0	3	5	0	**0**	**2**	**6**	**0**	**0**	**2**	**6**	**0**	0	3	5	0	0	3	5	0
	S	0	1	0	6	0	0	0	7	0	0	0	7	**0**	**0**	**0**	**7**	**0**	**0**	**0**	**7**	0	1	0	6	0	1	0	6
100	U	9	1	0	0	9	1	0	0	10	0	0	0	**10**	**0**	**0**	**0**	9	1	0	0	10	0	0	0	10	0	0	0
	B	1	6	1	0	1	7	0	0	1	6	1	0	**1**	**6**	**1**	**0**	1	6	1	0	1	6	1	0	1	4	3	0
	L	0	3	5	0	0	3	5	0	0	3	5	0	**0**	**2**	**6**	**0**	0	2	6	0	0	4	4	0	0	3	5	0
	S	0	1	0	6	0	0	0	7	0	0	0	7	**0**	**0**	**0**	**7**	0	0	0	7	0	0	0	7	0	1	0	6
1000	U	9	1	0	0	9	1	0	0	10	0	0	0	**10**	**0**	**0**	**0**	9	1	0	0	9	1	0	0	10	0	0	0
	B	1	6	1	0	1	7	0	0	1	6	1	0	**1**	**6**	**1**	**0**	1	6	1	0	1	6	1	0	1	6	1	0
	L	0	3	5	0	0	3	5	0	0	3	5	0	**0**	**2**	**6**	**0**	0	2	6	0	0	4	4	0	0	4	4	0
	S	0	1	0	6	0	0	0	7	0	0	0	7	**0**	**0**	**0**	**7**	0	0	0	7	0	0	0	7	0	0	0	7

Table 2.2: Results for 1vr

1v1	C_-	0.001				0.01				0.1				1				10				100				1000			
C_+	Cars	U	B	L	S	U	B	L	S	U	B	L	S	U	B	L	S	U	B	L	S	U	B	L	S	U	B	L	S
0.001	U	9	1	0	0	1	8	1	0	1	8	1	0	1	8	1	0	1	8	1	0	1	8	1	0	1	8	1	0
	B	1	5	2	0	0	1	7	0	0	1	7	0	0	1	7	0	0	1	7	0	0	1	7	0	0	1	6	1
	L	0	2	6	0	0	0	7	1	0	0	7	1	0	0	7	1	0	0	7	1	0	0	7	1	0	0	7	1
	S	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7
0.01	U	9	1	0	0	9	1	0	0	8	2	0	0	8	2	0	0	8	2	0	0	8	2	0	0	8	2	0	0
	B	0	3	5	0	1	5	2	0	1	3	4	0	1	3	4	0	1	3	4	0	1	3	4	0	1	3	4	0
	L	0	5	3	0	0	3	5	0	0	0	8	0	0	0	8	0	0	0	8	0	0	0	8	0	0	0	8	0
	S	0	1	1	5	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7
0.1	U	9	1	0	0	10	0	0	0	**10**	**0**	**0**	**0**	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0
	B	3	5	0	0	1	7	0	0	**1**	**5**	**2**	**0**	1	5	2	0	1	5	2	0	1	5	2	0	1	5	2	0
	L	0	5	3	0	0	6	2	0	**0**	**0**	**8**	**0**	0	0	8	0	0	0	8	0	0	0	8	0	0	0	8	0
	S	0	1	1	5	0	0	0	7	**0**	**0**	**0**	**7**	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7
1	U	9	1	0	0	9	1	0	0	10	0	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0
	B	3	5	0	0	1	7	0	0	1	7	0	0	1	5	2	0	1	5	2	0	1	5	2	0	1	5	2	0
	L	0	5	3	0	0	6	2	0	0	4	4	0	0	1	7	0	0	1	7	0	0	1	7	0	0	1	7	0
	S	0	1	1	5	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7
10	U	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0
	B	3	5	0	0	1	7	0	0	1	7	0	0	1	6	1	0	1	5	2	0	1	5	2	0	1	5	2	0
	L	0	5	3	0	0	6	2	0	0	4	4	0	0	4	4	0	0	2	6	0	0	2	6	0	0	2	6	0
	S	0	1	1	5	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7
100	U	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	0	0	1	9	0	0	1
	B	3	5	0	0	1	7	0	0	1	7	0	0	1	6	1	0	1	5	2	0	1	5	2	0	1	5	2	0
	L	0	5	3	0	0	6	2	0	0	4	4	0	0	4	4	0	0	4	4	0	0	2	6	0	0	2	6	0
	S	0	1	1	5	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7
1000	U	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0
	B	3	5	0	0	1	7	0	0	1	7	0	0	1	6	1	0	1	5	2	0	1	5	2	0	1	5	2	0
	L	0	5	3	0	0	6	2	0	0	4	4	0	0	4	4	0	0	4	4	0	0	4	4	0	0	2	6	0
	S	0	1	2	4	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7

Table 2.3: Results for 1v1

DDAG	C_-	0.001				0.01				0.1				1				10				100				1000			
C_+	Cars	U	B	L	S	U	B	L	S	U	B	L	S	U	B	L	S	U	B	L	S	U	B	L	S	U	B	L	S
0.001	U	9	1	0	0	1	8	1	0	1	8	1	0	1	8	1	0	1	8	1	0	1	8	1	0	1	8	1	0
	B	1	5	2	0	0	1	7	0	0	1	7	0	0	1	7	0	0	1	7	0	0	1	7	0	0	1	7	0
	L	0	2	6	0	0	0	7	1	0	0	7	1	0	0	7	1	0	0	7	1	0	0	7	1	0	0	7	1
	S	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7
0.01	U	9	1	0	0	9	1	0	0	8	2	0	0	8	2	0	0	8	2	0	0	8	2	0	0	8	2	0	0
	B	3	5	0	0	1	5	2	0	1	3	4	0	1	3	4	0	1	3	4	0	1	3	4	0	1	3	4	0
	L	0	5	3	0	0	3	5	0	0	0	8	0	0	0	8	0	0	0	8	0	0	0	8	0	0	0	8	0
	S	0	1	1	5	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7
0.1	U	9	1	0	0	9	1	0	0	**9**	**1**	**0**	**0**	**9**	**1**	**0**	**0**	**9**	**1**	**0**	**0**	**9**	**1**	**0**	**0**	**9**	**1**	**0**	**0**
	B	3	5	0	0	1	7	0	0	**1**	**5**	**2**	**0**	**1**	**5**	**2**	**0**	**1**	**5**	**2**	**0**	**1**	**5**	**2**	**0**	**1**	**5**	**2**	**0**
	L	0	5	3	0	0	6	2	0	**0**	**0**	**8**	**0**	**0**	**0**	**8**	**0**	**0**	**0**	**8**	**0**	**0**	**0**	**8**	**0**	**0**	**0**	**8**	**0**
	S	0	1	1	5	0	0	0	7	**0**	**0**	**0**	**7**	**0**	**0**	**0**	**7**	**0**	**0**	**0**	**7**	**0**	**0**	**0**	**7**	**0**	**0**	**0**	**7**
1	U	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0
	B	3	5	0	0	1	7	0	0	1	7	0	0	1	5	2	0	1	5	2	0	1	5	2	0	1	5	2	0
	L	0	5	3	0	0	6	2	0	0	4	4	0	0	1	7	0	0	1	7	0	0	1	7	0	0	1	7	0
	S	0	1	1	5	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7
10	U	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0
	B	3	5	0	0	1	7	0	0	1	7	0	0	1	6	1	0	1	5	2	0	1	5	2	0	1	5	2	0
	L	0	5	3	0	0	6	2	0	0	4	4	0	0	4	4	0	0	2	6	0	0	2	6	0	0	2	6	0
	S	0	1	1	5	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7
100	U	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	0	1	0	9	0	1	0
	B	3	5	0	0	1	7	0	0	1	7	0	0	1	6	1	0	1	5	2	0	1	5	2	0	1	5	2	0
	L	0	4	4	0	0	6	2	0	0	4	4	0	0	4	4	0	0	4	4	0	0	2	6	0	0	2	6	0
	S	0	0	1	6	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7
1000	U	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	1	0	0	9	0	1	0
	B	3	5	0	0	1	7	0	0	1	7	0	0	1	6	1	0	1	5	2	0	1	5	2	0	1	5	2	0
	L	0	4	4	0	0	6	2	0	0	4	4	0	0	4	4	0	0	4	4	0	0	4	4	0	0	2	6	0
	S	0	0	3	4	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7	0	0	0	7

Table 2.4: Results for DDAG

	loo results				rs results			
original	U	B	L	S	U	B	L	S
U	10	0	0	0	10	0	0	0
B	3	3	2	0	2	4	2	0
L	0	2	6	0	0	1	7	0
S	0	2	0	5	0	0	0	7

Table 2.5: Best results of accuracy in [43]

Finally, we compare our results with those obtained in [43]. In that paper, several strategies are applied to classify the elements of this database, and the best results are obtained for a distributional approach and are shown in Table 2.5.

We can observe that the number of misclassified elements is equal to nine (with five misclassified elements on the training sample), while our lowest number of misclassified elements (obtained for 1v1) is equal to three (with three misclassified elements in the resubstitution process). In fact, for 1vr we had that for every combination of the constants, the number of misclassified elements was always smaller than or equal to eight. In [43], it is explained that their methods had tendency to overfit the data. In fact in some of their experiments, they obtained two misclassified elements on the training sample, but higher numbers for the test sample. Our method gets even better results in some cases for the training sample (in 1v1, we got only one mistake) and much better results for the test sample in most of the cases. Then, the results have been clearly improved and hence, one can say that our method is competitive.

2.4 Computational experiment with uncertain values

2.4.1 Computational experiment

We have applied our model to a database where the single instances have been transformed into intervals to represent uncertainty on the data. We have used the 'breast-cancer' dataset, which can be downloaded from the UCI Machine Learning Repository [4]. This dataset is composed of 699 instances, each one representing an individual affected by breast cancer, and 9 measurements (represented via a number between 1 and 10) have been taken from each individual. The instances are classified as benign (group G_+) or malignant (G_-). In the database, there are 16 instances with missing values (all of them for the sixth variable), which have been erased for the study (then, there are 444 instances in G_+ and 239 instances in G_-, 683 in total).

In order to construct the interval, we have computed the standard deviation σ_{x_j}

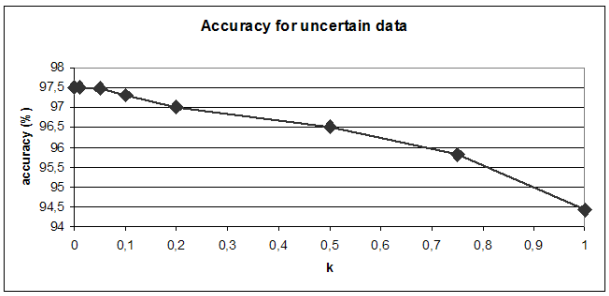

Figure 2.4: Results for uncertain data

in each coordinate j from 1 to d of each group of the dataset (G_+ and G_-). Then, each coordinate x_{ij} is replaced by the interval $[x_{ij} - k\sigma_{x_j}, x_{ij} + k\sigma_{x_j}]$, with σ_{x_j} the standard deviation of the corresponding group. The values for k are 0, 0.01, 0.05, 0.1, 0.2, 0.5, 0.75 and 1. The higher the value of k, the larger the uncertainty about the data. Observe that, when $k = 0$, we obtain the original database.

We have solved the corresponding classification problem with the interval data through 10-fold cross validation, that is, the instances of the database are grouped in 10 sets (these sets forming a partition) and, each one has been used in turn as test set against all 9 others taken together as training set, that is, the process is repeated ten times (see [66]).

The optimization problem (2.17) to compute the parameters of each classifier has been solved via AMPL+CPLEX, for different pairs of C_+ and C_-. And we have computed the classification accuracy when the fuzzy rule is considered.

2.4.2 Numerical results

In Tables 2.6-2.7, we present the results for the interval dataset, depending on the value of the parameter k. In Figure 2.4, the best results in the test sample for each value of k are depicted.

One can observe that, although in general the results get worse as the value of k is increasing, this process is quite smooth, and the results for the accuracy are quite similar in the first tables. In fact, the best results for accuracy continue being quite good when k increases (the accuracy is over 94% in each case).

We can also observe that the best values of accuracy are obtained for similar or almost similar values of C_+ and C_-. In fact, as k increases, the accuracy in the training and test sample for different values of C_+ and C_- is very bad (in some cases, we obtain degenerate solutions for a pair of values (C_+, C_-), with $C_+ \neq C_-$), while the accuracy in the diagonal of the table continues being very acceptable (around 95%).

		Train						Test					
k	$C_+ \backslash C_-$	0.001	0.01	0.1	1	10	100	1000	0.001	0.01	0.1	1	10	100	1000
0	0.001	96.88	97.49	95.40	92.61	91.93	91.80	91.80	96.64	**97.50**	95.30	92.26	91.81	91.81	91.81
	0.01	94.05	97.32	97.46	95.67	94.78	94.61	94.60	94.14	96.63	**97.50**	95.59	94.72	94.58	94.58
	0.1	82.37	94.16	97.25	97.43	95.75	95.12	94.92	81.56	93.40	96.78	97.36	95.74	94.87	94.72
	1	73.34	84.48	94.35	97.20	97.43	95.77	95.17	72.76	83.46	93.55	96.78	97.21	95.74	94.87
	10	72.51	78.93	85.15	94.35	97.20	97.41	95.77	72.47	78.04	84.20	93.55	96.78	97.21	95.74
	100	72.41	77.96	80.48	85.25	94.35	97.20	97.41	72.47	77.17	79.21	84.20	93.55	96.78	97.21
	1000	72.41	77.91	79.63	80.66	85.25	94.35	97.20	72.47	77.17	78.77	79.21	84.20	93.55	96.78
0.01	0.001	96.90	97.48	95.16	92.31	91.55	91.50	91.50	96.64	**97.50**	95.20	92.16	91.58	91.53	91.52
	0.01	94.03	97.32	97.44	95.55	94.55	94.31	94.26	94.11	96.61	**97.50**	95.52	94.35	94.02	94.04
	0.1	82.18	94.24	97.26	97.43	95.69	94.99	94.86	81.43	93.63	96.78	97.40	95.52	94.72	94.56
	1	73.00	84.42	94.30	97.24	97.43	95.71	95.03	72.76	83.19	93.66	96.78	97.31	95.53	94.71
	10	72.19	78.28	84.78	94.32	97.23	97.43	95.71	72.04	77.55	84.00	93.68	96.78	97.31	95.53
	100	72.13	77.47	79.70	84.85	94.32	97.23	97.43	72.04	77.04	79.08	84.02	93.68	96.78	97.31
	1000	72.12	77.39	78.87	79.85	84.86	94.32	97.23	72.05	77.01	78.49	79.16	84.02	93.68	96.78
0.05	0.001	96.85	97.42	94.32	90.53	89.88	89.80	89.80	96.60	**97.48**	93.93	90.29	89.35	89.28	89.28
	0.01	94.09	97.29	97.38	95.18	93.64	93.36	93.34	93.92	96.73	97.35	95.00	93.40	93.13	93.10
	0.1	81.05	94.43	97.29	97.36	95.28	94.06	93.80	80.47	93.89	96.67	97.30	95.01	93.58	93.44
	1	71.97	83.39	94.48	97.29	97.35	95.29	94.11	71.89	82.24	93.87	96.69	97.30	95.02	93.64
	10	71.35	75.84	83.71	94.49	97.29	97.35	95.29	71.32	75.86	82.58	93.87	96.69	97.30	95.02
	100	71.27	75.07	76.76	83.74	94.49	97.29	97.35	71.26	75.03	76.64	82.62	93.87	96.69	97.30
	1000	71.27	75.00	76.10	76.86	83.75	94.49	97.29	71.25	74.98	75.90	76.73	82.62	93.87	96.69
0.1	0.001	96.81	97.26	93.12	87.78	86.97	86.89	86.88	96.55	97.30	92.79	87.33	86.68	86.61	86.59
	0.01	94.12	97.26	97.28	94.42	92.25	91.95	91.91	93.77	96.77	**97.31**	94.01	91.70	91.46	91.42
	0.1	79.45	94.52	97.30	97.29	94.56	92.64	92.36	78.87	93.84	96.81	97.26	94.20	92.27	92.01
	1	70.70	81.87	94.57	97.30	97.28	94.58	92.68	70.41	80.97	93.85	96.81	97.26	94.22	92.34
	10	69.76	72.86	82.19	94.58	97.30	97.28	94.58	69.75	72.73	81.30	93.85	96.81	97.26	94.22
	100	69.69	72.35	73.34	82.22	94.58	97.30	97.28	69.68	72.35	73.18	81.33	93.85	96.81	97.26
	1000	69.69	72.30	72.84	73.40	82.22	94.58	97.30	69.67	72.29	72.89	73.24	81.34	93.85	96.81

Table 2.6: Results for uncertain data, with k =0, 0.01, 0.05, 0.1

		Train							Test					
k	$C_+ \backslash C_-$	0.001	0.01	0.1	1	10	100	1000	0.001	0.01	0.1	1	10	100	1000
0.2	0.001	96.83	96.98	88.59	74.72	72.08	71.75	71.72	96.63	**97.00**	88.27	74.22	71.57	71.28	71.25
	0.01	93.89	97.13	96.94	91.33	85.65	85.26	85.19	93.56	96.71	96.99	90.96	84.82	84.48	84.41
	0.1	75.63	94.33	97.20	96.93	91.62	86.68	86.39	75.08	93.72	96.73	96.94	91.20	85.81	85.57
	1	67.44	78.22	94.39	97.21	96.93	91.65	86.79	67.16	77.80	93.69	96.73	96.93	91.23	85.92
	10	66.65	68.83	78.52	94.39	97.21	96.93	91.65	66.47	68.62	78.14	93.69	96.73	96.93	91.23
	100	66.64	67.69	69.01	78.55	94.39	97.21	96.93	66.45	67.30	68.82	78.17	93.69	96.73	96.93
	1000	66.64	67.65	67.88	69.03	78.55	94.39	97.21	66.45	67.26	67.48	68.84	78.17	93.69	96.73
0.5	0.001	96.55	95.00	35.21	34.99	34.99	34.99	34.99	96.42	94.79	35.18	34.99	34.99	34.99	34.99
	0.01	91.42	96.67	95.19	36.32	34.99	34.99	34.99	91.04	**96.50**	94.99	36.31	34.99	34.99	34.99
	0.1	65.55	92.03	96.67	95.19	36.51	34.99	34.99	65.58	91.63	96.49	95.00	36.50	34.99	34.99
	1	65.01	65.86	92.10	96.67	95.19	36.53	34.99	65.01	65.84	91.68	96.49	95.00	36.52	34.99
	10	65.01	65.01	65.90	92.10	96.67	95.19	36.54	65.01	65.01	65.88	91.69	96.49	95.00	36.52
	100	65.01	65.01	65.01	65.90	92.10	96.67	95.19	65.01	65.01	65.01	65.88	91.69	96.49	95.00
	1000	65.01	65.01	65.01	65.01	65.90	92.10	96.67	65.01	65.01	65.01	65.01	65.88	91.69	96.49
0.75	0.001	95.86	77.29	34.99	34.99	34.99	34.99	34.99	95.72	77.19	34.99	34.99	34.99	34.99	34.99
	0.01	86.13	96.02	81.72	34.99	34.99	34.99	34.99	85.64	**95.81**	81.59	34.99	34.99	34.99	34.99
	0.1	65.01	87.23	96.03	82.13	34.99	34.99	34.99	65.01	86.80	95.81	82.00	34.99	34.99	34.99
	1	65.01	65.01	87.35	96.03	82.17	34.99	34.99	65.01	65.01	86.91	95.81	82.04	34.99	34.99
	10	65.01	65.01	65.01	87.36	96.03	82.18	34.99	65.01	65.01	65.01	86.92	95.81	82.04	34.99
	100	65.01	65.01	65.01	65.01	87.36	96.03	82.18	65.01	65.01	65.01	65.01	86.92	95.81	82.04
	1000	65.01	65.01	65.01	65.01	65.01	87.36	96.03	65.01	65.01	65.01	65.01	65.01	86.92	95.81
1	0.001	94.08	34.99	34.99	34.99	34.99	34.99	34.99	93.96	34.98	34.99	34.99	34.99	34.99	34.99
	0.01	77.70	94.55	34.99	34.99	34.99	34.99	34.99	77.27	94.37	34.98	34.99	34.99	34.99	34.99
	0.1	65.01	79.66	94.62	34.99	34.99	34.99	34.99	65.01	79.21	94.42	34.97	34.99	34.99	34.99
	1	65.01	65.01	79.87	94.62	34.99	34.99	34.99	65.01	65.01	79.41	94.43	34.97	34.99	34.99
	10	65.01	65.01	65.01	79.89	94.63	34.99	34.99	65.01	65.01	65.01	79.43	**94.43**	34.97	34.99
	100	65.01	65.01	65.01	65.01	79.89	94.63	34.99	65.01	65.01	65.01	65.01	79.43	**94.43**	34.97
	1000	65.01	65.01	65.01	65.01	65.01	79.89	94.63	65.01	65.01	65.01	65.01	65.01	79.43	**94.43**

Table 2.7: Results for uncertain data, with k =0.2, 0.5, 0.75, 1

2.5 Computational experiment with missing values

2.5.1 Imputation for missing values via intervals

In the task of editing survey data, the term 'missing data' is used to denote invalid blanks in an entry of any field of the survey (invalid in the sense that this value should appear in the dataset) or inconsistent entries.

One of the most widely-used strategies to deal with missing data is the imputation for given records, which means to replace the missing values of a database by other plausible values in such a way that the data must remain consistent.

Different methodologies for imputation have been studied in the literature (see [72, 73, 75] for a list of them), like using the mean (for quantitative variables) or the mode (for qualitative variables) of the non-missing values of the database (see e.g. [1]). One of the drawbacks of this method is that the standard deviation of the sample is ignored and it can contain relevant information which should be taken into account during the imputation process (see [97]).

Our idea to impute the missing values is to replace each blank by an interval (instead of a single value) built with the non-missing values of the dataset. This way, when a blank appears in the j-th variable of an observation, we use the non-missing values in the j-th variable to construct the interval.

Two different strategies will be followed to impute the missing values, by transforming our database into an interval-valued dataset. The first one is to build an interval based on the mean and deviation of the remaining values. This way, the standard deviation is taken into account for the imputation process. For a missing value in the j-th variable of an instance of the training sample, we fill it in by computing the mean $\overline{x}_j$ and the standard deviation σ_{x_j} for the values in this column of the remaining values, and afterwards, we substitute the blank by the interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$.

The second strategy is not based on the mean and the deviation, but on the quantiles. We consider the interval which is defined as $[Q_a, Q_{1-a}]$, where Q_a represents the a-th quantile, and thus the interval contains all but a fraction $2a$ of the all non-missing values.

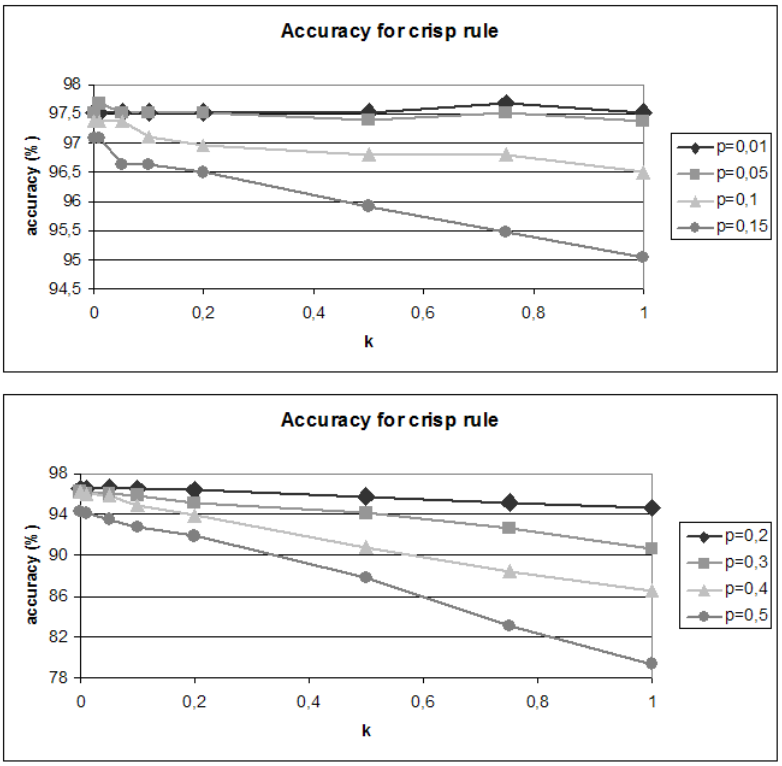

Figure 2.5: Accuracy for the crisp rule when using the interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$

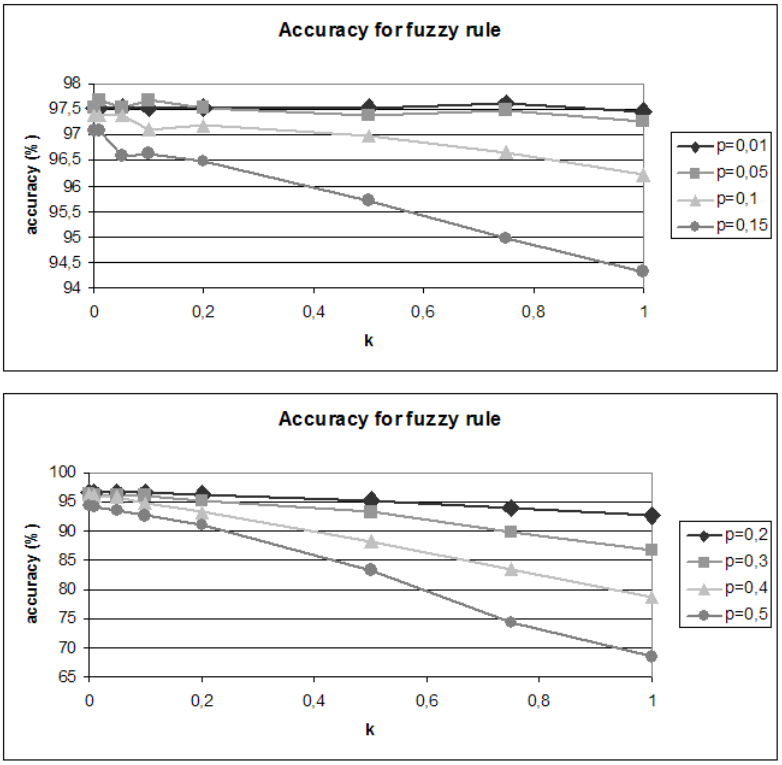

Figure 2.6: Accuracy for the fuzzy rule when using the interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$

2.5.2 Computational experiment with missing data completely at random

Our model for classification with interval data has been applied to a database for dealing with missing values. We have used the 'breast-cancer' dataset (UCI Machine Learning Repository [4]), with 683 instances in total.

The missing data have been generated in the database completely at random. A parameter p has been defined as the probability of replacing the value of a variable in the database by a missing value. The following values of p have been incorporated in the computational experiment: 0.01, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4 and 0.5. That is, we have performed numerical experiments where very few values have been erased ($p = 0.01, 0.05$) and where around half of the database is missing ($p = 0.5$).

With this modified database, for each value of p, we have solved the corresponding classification problem through 10-fold cross validation (see [66]).

Before solving the corresponding optimization problem (2.17), we have used the two different strategies explained before for imputation. In the first strategy, we replace the blank by the interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$, where $\overline{x}_j$ and σ_{x_j} have been computed with

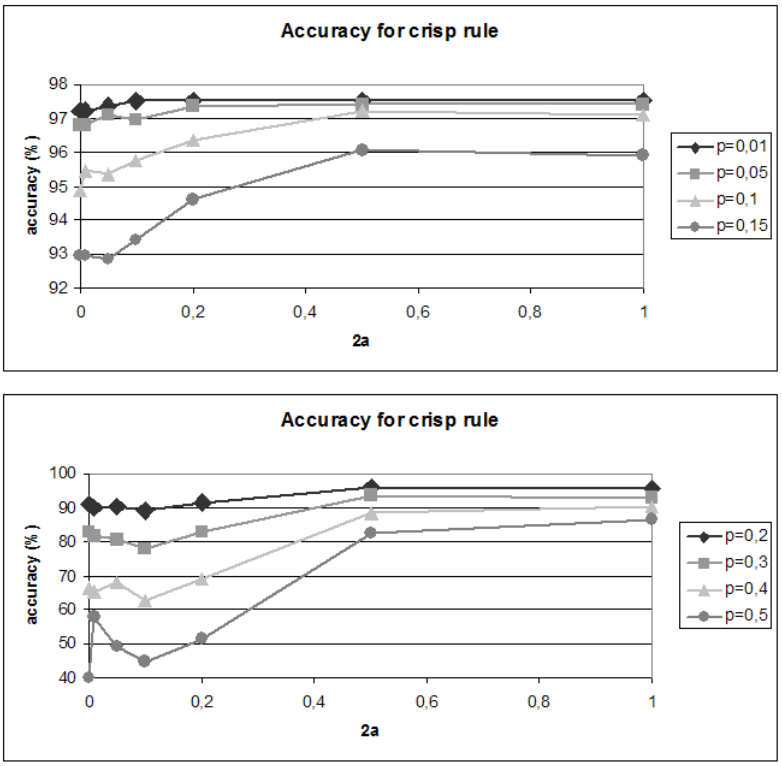

Figure 2.7: Accuracy for the crisp rule when using the interval $[Q_a, Q_{1-a}]$

the non-missing values in the j-th column of the instances of the corresponding group (G_+ or G_-). In the case of a missing value in any coordinate of an instance in the test sample, we compute the mean and the deviation of the corresponding variable for the remaining instances of the database (since the group which it belongs to is unknown). The values for k are 0, 0.01, 0.05, 0.1, 0.2, 0.5, 0.75 and 1. Observe that, when $k = 0$, the interval is a single point and we are considering the imputation to the mean.

In the second strategy, the blank is replaced by the interval $[Q_a, Q_{1-a}]$, Q_a being the a-th quantile. The values studied for $2a$ are 0, 0.01, 0.05, 0.1, 0.2, 0.5 and 1. Observe that, when $2a = 0$, the interval is the range of the variable, when $2a = 0.5$, we obtain the interquartile range, and when $2a = 1$, the interval is reduced to a singleton which is the median of the variable.

Furthermore, another standard interval based on quantiles is included in this analysis. It is the case of the interval defined by the inner fences (see [110]), which is an interval used in the analysis of outliers (every observation which is not contained between the inner fences is considered as an outlier). This interval is based on the quartiles of the sample,

$$IF = [Q_{0.25} - 1.5 \cdot (Q_{0.75} - Q_{0.25}), Q_{0.75} + 1.5 \cdot (Q_{0.75} - Q_{0.25})]. \tag{2.25}$$

All these previous modifications of the database have been performed with Matlab 6.5.

Then, we solve the optimization problem (2.17) to compute the parameters of the classifier with LOQO, [112], for different values of C, where $C_+ = C_- = C$ (previous experiments showed that the best results were obtained when the two constants were close to each other). The two classification rules (fuzzy and crisp) have been considered, and in Tables 2.8-2.23, we present the results of the accuracy in each case.

2.5.3 Numerical results

The accuracy, following the crisp or the fuzzy rule, for the different datasets obtained as a result of erasing values with a probability p and replacing those missing values by intervals depending on the parameter k or a, are presented in Tables 2.8-2.23. In Tables 2.8-2.15, we present the results when using the interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$, and in Tables 2.16-2.23, the results obtained for $[Q_a, Q_{1-a}]$. In each case, we show the accuracy in the training sample, in each group of the test sample, and the average in the test sample. The best results of accuracy for each combination of (p, k) or $(p, 2a)$ (including the IF-interval, defined in (2.25)), are shown in the tables in bold and are depicted in Figures 2.5-2.9.

In the case of replacing the missing values by the interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$ (Tables 2.8-2.15), one can observe that, when the percentage of introduced missing

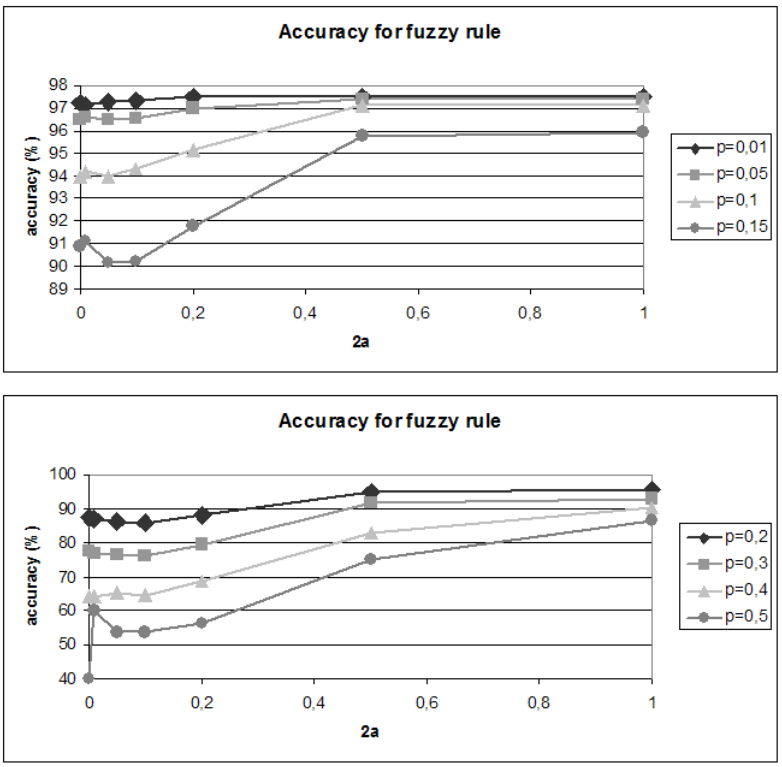

Figure 2.8: Accuracy for the fuzzy rule when using the interval $[Q_a, Q_{1-a}]$

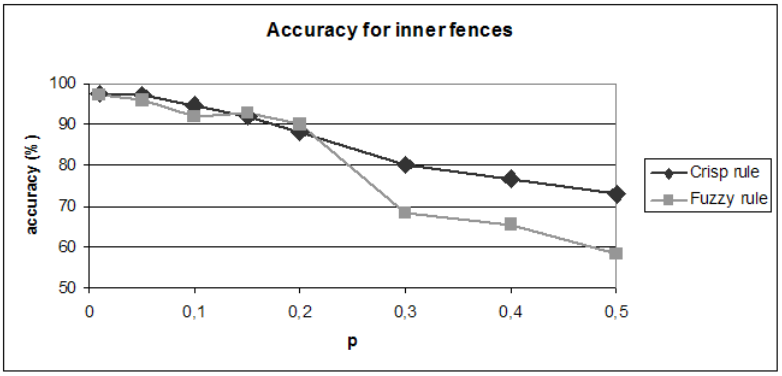

Figure 2.9: Accuracy for the two rules when using the inner fences

values is not too high (less than 30%), we obtain better results (or at least similar results) with interval imputation than with only mean imputation (case $k = 0$). In general, the best results are obtained for small intervals (k equal to 0.01 or 0.05). When the parameter p increases, the accuracy for high values of k decreases because, although the results for the training sample continue being very good, the accuracy in the test sample is worse, because the classifier overfits the parameters for the training sample.

In the case of intervals based on quantiles (the different values of $2a$ and the IF-interval), one can observe that, for the crisp classification rule (Tables 2.16-2.19), the interquartile range ($2a = 0.5$) seems to be a better imputation than the median ($2a = 1$), because the results are better or equal in every case for $p \leq 0.3$. This is not so clear when using the fuzzy rule (Tables 2.20-2.23). And when p increases, the results obtained for small values of a (bigger intervals) are not very good, due to the overfitting.

Likewise, we must say that, in general, the results obtained for the intervals based on the mean and the deviation are better than those obtained for intervals based on quantiles.

CRISP	k	0				0.01				0.05				0.1			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.01	1	97.49	97.31	97.50	97.38	97.49	97.31	97.50	97.38	97.49	97.31	97.50	97.38	97.49	97.31	97.08	97.23
	10	97.53	97.10	97.92	97.39	97.51	97.10	97.92	97.39	97.49	97.10	97.92	97.39	97.49	97.10	97.92	97.39
	10^2	97.61	97.31	97.92	**97.52**	97.62	97.31	97.92	**97.52**	97.59	97.31	97.92	**97.52**	97.62	97.31	97.92	**97.52**
	10^3	97.64	97.33	97.08	97.24	97.67	97.10	97.50	97.24	97.64	97.10	97.92	97.39	97.67	97.10	97.92	97.39
	10^4	97.67	97.33	97.08	97.24	97.64	97.10	97.50	97.24	97.66	97.54	96.67	97.23	97.66	97.10	97.92	97.39
	10^5	97.66	97.33	97.50	97.39	97.64	97.54	96.67	97.23	97.64	97.31	97.92	**97.52**	97.67	97.54	96.67	97.23
	10^6	97.71	97.33	96.67	97.10	97.71	97.33	96.67	97.10	97.62	97.54	97.08	97.38	97.61	97.33	97.08	97.24
0.05	1	97.61	97.10	96.25	96.80	97.63	97.10	96.25	96.80	97.63	97.10	96.25	96.80	97.64	97.10	96.25	96.80
	10	97.61	97.10	97.91	97.39	97.63	97.10	97.92	97.39	97.61	97.10	97.92	97.39	97.62	97.10	97.92	97.39
	10^2	97.72	97.33	97.50	97.39	97.67	97.31	97.92	97.52	97.72	97.10	97.92	97.39	97.69	97.10	97.92	97.39
	10^3	97.74	97.54	97.50	**97.52**	97.67	97.54	97.92	**97.67**	97.67	97.33	97.92	**97.53**	97.67	97.33	97.92	**97.53**
	10^4	97.71	97.54	97.50	**97.52**	97.71	97.54	97.50	97.52	97.74	97.54	97.50	97.52	97.77	97.33	97.92	**97.53**
	10^5	97.74	97.54	97.50	**97.52**	97.71	97.54	97.50	97.52	97.74	97.54	97.50	97.52	97.74	97.33	97.08	97.24
	10^6	97.72	97.54	97.50	**97.52**	97.76	97.54	97.08	97.38	97.72	97.55	97.08	97.38	97.74	97.54	93.74	96.21
0.10	1	97.87	97.10	95.83	96.66	97.87	97.10	95.83	96.66	97.85	97.10	95.42	96.51	97.89	97.10	95.42	96.51
	10	98.03	97.31	97.50	**97.38**	98.02	97.31	97.50	**97.38**	98.00	97.31	97.50	**97.38**	98.00	97.10	97.08	**97.10**
	10^2	98.11	97.33	97.08	97.24	98.15	97.08	97.08	97.08	98.10	97.08	97.08	97.08	98.13	97.31	96.25	96.94
	10^3	98.18	97.54	95.83	96.94	98.10	97.54	96.67	97.23	98.03	97.54	96.67	97.23	98,05	97.33	95.42	96.66
	10^4	98.13	97.54	95.83	96.94	98.11	97.54	96.25	97.09	98.16	97.54	96.25	97.09	98.13	97.33	95.83	96.81
	10^5	98.11	97.54	96.25	97.09	98.13	97.54	95.83	96.94	98.13	97.54	95.83	96.94	98.03	97.33	96.25	96.95
	10^6	98.13	97.54	95.83	96.94	98.13	97.54	96.25	97.09	98.08	97.54	95.83	96.94	98.10	97.33	96.25	96.95
0.15	1	98.10	97.31	94.57	96.35	98.08	97.31	94.57	96.35	98.06	97.31	94.57	96.35	98.00	97.54	94.57	96.50
	10	98.15	97.54	95.40	96.79	98.15	97.31	95.40	96.64	98.11	97.31	95.40	**96.64**	98.10	97.31	94.98	96.50
	10^2	98.29	97.54	96.23	**97.08**	98.34	97.54	96.23	**97.08**	98.32	97.31	94.98	96.50	98.29	97.31	95.40	**96.64**
	10^3	98.52	97.54	93.73	96.21	98.60	97.54	93.73	96.21	98.52	97.54	94.57	96.50	98.50	97.54	94.15	96.35
	10^4	98.62	97.54	93.32	96.06	98.65	97.54	93.73	96.21	98.62	97.54	92.48	95.77	98.57	97.54	93.32	96.06
	10^5	98.62	97.54	92.48	95.77	98.65	97.54	93.32	96.06	98.63	97.54	92.90	95.91	98.60	97.54	92.90	95.91
	10^6	98.62	97.54	92.90	95.91	98.54	97.54	92.90	95.92	98.45	97.54	94.57	96.50	98.57	97.54	92.90	95.91

Table 2.8: Crisp rule, p =0.01, 0.05, 0.1, 0.15 and k =0, 0.01, 0.05, 0.1

CRISP	k	0.2				0.5				0.75				1			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.01	1	97.49	97.31	97.08	97.23	97.51	97.31	97.08	97.23	97.51	97.31	97.08	97.23	97.48	97.31	97.08	97.23
	10	97.49	97.31	97.92	97.52	97.49	97.10	97.92	97.39	97.53	97.10	97.92	97.39	97.54	97.10	97.92	97.39
	10^2	97.59	97.31	97.92	97.52	97.58	97.31	97.50	97.38	97.62	97.54	97.92	**97.67**	97.58	97.31	97.92	**97.52**
	10^3	97.64	97.54	97.50	**97.52**	97.66	97.31	97.50	97.38	97.66	97.54	97.50	97.52	97.67	97.54	97.08	97.38
	10^4	97.66	97.54	97.50	**97.52**	97.72	97.54	97.50	**97.52**	97.72	97.54	97.08	97.38	97.66	97.54	96.67	97.23
	10^5	97.67	97.31	97.50	97.38	97.71	97.33	97.08	97.24	97.67	97.54	95.83	96.94	97.69	97.54	96.67	97.23
	10^6	97.74	97.33	97.08	97.24	97.66	97.54	97.08	97.38	97.72	97.54	96.25	97.09	97.67	97.54	96.25	97.09
0.05	1	97.64	97.10	96.25	96.80	97.54	97.31	95.83	96.79	97.45	97.31	95.83	96.79	97.32	97.31	96.25	96.94
	10	97.62	97.10	97.92	97.39	97.62	97.10	97.08	97.10	97.61	97.10	97.08	97.10	97.53	97.10	97.08	97.10
	10^2	97.72	97.10	97.50	97.24	97.72	97.10	97.50	97.24	97.72	97.31	97.92	**97.52**	97.69	97.54	97.08	**97.38**
	10^3	97.72	97.33	97.50	97.39	97.72	97.33	97.50	**97.39**	97.71	97.33	97.50	97.39	97.77	97.31	97.08	97.23
	10^4	97.72	97.33	97.92	**97.53**	97.79	97.33	97.08	97.24	97.75	97.10	96.67	96.95	97.77	97.31	96.67	97.09
	10^5	97.72	97.33	97.92	**97.53**	97.79	97.33	97.08	97.24	97.74	97.10	97.08	97.10	97.72	97.31	96.67	97.09
	10^6	97.75	97.33	97.08	97.24	97.77	97.33	97.50	**97.39**	97.79	97.54	97.08	97.38	97.79	97.31	96.67	97.09
0.10	1	97.87	97.10	95.00	96.37	97.64	97.31	95.42	96.65	97.64	97.31	95.42	96.65	97.67	97.10	93.73	95.92
	10	97.87	97.10	96.67	96.95	97.74	97.10	96.25	**96.80**	97.74	97.10	96.25	**96.80**	97.64	97.10	95.42	**96.51**
	10^2	97.93	96.87	97.08	96.95	97.76	96.87	95.00	96.22	97.76	96.87	95.00	96.22	97.62	96.65	94.17	95.78
	10^3	97.95	97.31	95.83	96.79	97.79	97.31	95.42	96.65	97.79	97.31	95.42	96.65	97.67	96.87	94.17	95.93
	10^4	97.82	97.33	96.25	**96.95**	97.82	97.31	95.83	96.79	97.82	97.31	95.83	96.79	97.67	97.08	93.73	95.91
	10^5	97.85	97.10	96.67	96.95	97.84	97.08	96.25	96.79	97.84	97.08	96.25	96.79	97.70	97.08	94.17	96.06
	10^6	97.89	97.08	96.67	96.94	97.82	96.86	96.25	96.64	97.82	96.86	96.25	96.64	97.71	97.08	94.17	96.06
0.15	1	98.02	97.54	93.73	96.21	97.84	97.31	92.90	95.77	97.56	97.31	91.23	95.18	97.22	97.31	90.82	**95.04**
	10	98.08	97.54	94.57	**96.50**	98.03	97.08	93.73	**95.91**	97.93	96.86	92.48	95.33	97.54	97.10	90.40	94.76
	10^2	98.26	97.31	94.57	96.35	98.08	97.08	93.32	95.76	97.93	96.86	92.07	95.18	97.85	96.86	90.40	94.60
	10^3	98.32	97.54	94.57	**96.50**	98.21	97.08	93.32	95.76	97.97	97.08	91.65	95.18	97.85	97.08	90.82	94.89
	10^4	98.54	97.54	94.15	96.35	98.39	97.54	92.48	95.77	97.97	97.08	92.48	**95.47**	97.85	96.86	90.82	94.74
	10^5	98.49	97.54	93.73	96.21	98.34	97.54	92.48	95.77	97.95	96.86	92.07	95.18	97.92	96.86	90.40	94.60
	10^6	98.55	97.54	93.32	96.06	98.29	97.31	92.48	95.62	97.98	96.86	92.48	95.33	97.89	97.08	90.40	94.74

Table 2.9: Crisp rule, p =0.01, 0.05, 0.1, 0.15 and k =0.2, 0.5, 0.75, 1

CRISP	k	0				0.01				0.05				0.1			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.2	1	98.24	97.54	94.57	**96.50**	98.24	97.54	94.57	**96.50**	98.24	97.77	94.57	**96.65**	98.24	97.54	94.57	**96.50**
	10	98.42	97.54	94.57	**96.50**	98.42	97.54	94.57	**96.50**	98.44	97.33	94.15	96.22	98.39	97.33	94.57	96.36
	10^2	98.73	97.54	92.90	95.91	98.70	97.54	92.07	95.62	98.70	97.54	93.32	96.06	98.68	97.54	93.32	96.06
	10^3	98.93	97.29	91.21	95.16	98.86	97.31	92.90	95.77	98.91	96.86	91.23	94.89	98.84	97.06	91.21	95.02
	10^4	99.01	97.52	90.80	95.17	98.99	97.52	91.63	95.46	99.02	97.75	90.80	95.31	98.94	97.52	90.80	95.17
	10^5	99.06	97.75	90.38	95.17	99.02	97.52	91.63	95.46	99.01	97.31	90.40	94.89	99.04	97.52	89.96	94.88
	10^6	99.02	97.75	92.46	95.90	99.02	97.75	91.63	95.61	99.02	97.54	91.23	95.33	99.01	97.29	90.80	95.02
0.3	1	98.98	97.33	93.73	96.07	98.98	97.33	93.73	**96.07**	98.98	97.33	93.73	**96.07**	98.98	97.33	93.32	**95.92**
	10	98.98	97.77	93.32	**96.21**	98.93	97.77	92.48	95.92	98.93	97.54	92.07	95.62	98.93	97.54	90.40	95.04
	10^2	99.46	97.97	89.98	95.18	99.43	97.75	90.82	95.32	99.35	97.75	90.82	95.32	99.32	97.75	91.23	95.47
	10^3	99.58	97.97	89.98	95.18	99.58	97.97	90.82	95.47	99.58	97.97	90.40	95.32	99.54	97.97	89.55	95.02
	10^4	99.58	97.97	89.11	94.87	99.58	97.97	89.13	94.88	99.58	98.20	89.11	95.02	99.58	98.20	87.45	94.44
	10^5	99.59	97.97	89.11	94.87	99.59	97.97	89.11	94.87	99.58	98.20	88.70	94.87	99.58	98.20	88.28	94.73
	10^6	99.59	97.97	89.11	94.87	99.59	98.20	88.70	94.87	99.56	97.97	88.70	94.73	99.56	97.97	87.86	94.44
0.4	1	99.17	97.77	93.32	**96.21**	99.17	97.77	92.90	**96.06**	99.17	97.77	92.46	**95.91**	99.17	97.77	89.51	**94.88**
	10	99.33	98.22	88.68	94.88	99.33	97.99	89.95	95.18	99.30	97.77	89.11	94.74	99.22	97.77	88.26	94.44
	10^2	99.50	98.20	88.26	94.72	99.48	97.99	89.11	94.88	99.48	97.54	87.86	94.15	99.50	97.97	85.78	93.71
	10^3	99.77	98.88	82.01	92.98	99.77	98.66	81.58	92.68	99.67	98.66	84.08	93.55	99.63	98.20	83.22	92.96
	10^4	99.76	98.66	82.01	92.83	99.77	98.66	80.72	92.38	99.71	98.66	81.14	92.53	99.66	98.66	81.99	92.82
	10^5	99.80	98.86	81.58	92.81	99.82	98.66	81.16	92.53	99.79	98.66	80.74	92.39	99.74	98.66	79.04	91.79
	10^6	99.84	98.86	81.99	92.96	99.82	98.86	81.58	92.81	99.79	98.66	79.89	92.09	99.69	98.66	81.12	92.52
0.5	1	99.40	97.77	87.88	**94.31**	99.40	97.77	87.45	**94.15**	99.38	97.77	85.76	**93.56**	99.38	97.54	84.08	**92.83**
	10	99.51	97.77	85.76	93.56	99.51	97.77	85.76	93.56	99.50	97.99	84.09	93.13	99.46	97.99	82.39	92.53
	10^2	99.85	97.80	80.36	91.70	99.85	98.01	79.93	91.68	99.85	97.80	79.11	91.26	99.85	97.58	77.83	90.66
	10^3	99.89	97.33	75.33	89.63	99.89	97.56	74.91	89.63	99.89	97.56	72.37	88.74	99.87	97.78	72.37	88.89
	10^4	100	97.54	73.66	89.18	100	97.54	72.83	88.89	100	97.33	71.56	88.31	100	97.78	69.86	88.01
	10^5	100	97.54	74.08	89.33	100	97.54	72.83	88.89	100	97.33	71.56	88.31	100	97.56	69.86	87.86
	10^6	100	97.54	73.66	89.18	100	97.54	72.83	88.89	100	97.33	71.56	88.31	100	97.56	69.86	87.86

Table 2.10: Crisp rule, p =0.2, 0.3, 0.4, 0.5 and k =0, 0.01, 0.05, 0.1

CRISP	k	0.2				0.5				0.75				1			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.2	1	98.24	97.54	93.73	96.21	98.02	97.54	91.65	95.48	97.77	97.54	89.96	94.89	97.30	97.54	87.84	94.15
	10	98.31	97.33	94.15	96.22	98.13	97.33	92.90	**95.78**	98.11	97.31	91.23	**95.18**	97.79	97.31	89.57	**94.60**
	10^2	98.62	97.54	94.15	**96.35**	98.39	97.10	91.65	95.19	98.18	97.08	89.96	94.59	97.97	96.86	87.88	93.72
	10^3	98.83	97.08	92.07	95.33	98.42	96.86	90.78	94.73	98.24	96.86	88.28	93.85	97.95	96.65	87.86	93.57
	10^4	98.80	97.54	90.38	95.03	98.55	97.31	88.70	94.30	98.32	96.65	88.30	93.73	97.98	96.65	87.03	93.28
	10^5	98.83	97.54	91.65	95.48	98.54	96.86	89.53	94.29	98.26	96.65	88.30	93.73	98.00	96.65	87.45	93.43
	10^6	98.81	97.54	90.80	95.18	98.52	97.31	88.70	94.30	98.32	96.65	87.88	93.58	98.03	96.42	85.78	92.70
0.3	1	98.91	97.33	91.23	**95.20**	98.67	97.33	87.90	94.03	98.42	97.33	82.01	91.97	97.75	97.33	78.21	90.64
	10	98.86	97.54	90.40	95.04	98.68	97.31	88.32	**94.16**	98.50	97.31	84.13	**92.70**	98.34	96.19	80.34	**90.65**
	10^2	99.11	97.75	89.98	95.03	98.89	96.63	86.63	93.13	98.47	96.63	82.41	91.65	98.28	96.17	79.06	90.18
	10^3	99.46	97.54	88.32	94.31	98.93	96.63	85.38	92.69	98.55	96.42	82.83	91.66	98.32	96.17	77.39	89.60
	10^4	99.56	97.97	87.03	94.14	98.96	96.86	85.78	92.98	98.54	96.40	81.58	91.21	98.23	95.95	78.22	89.75
	10^5	99.58	97.97	86.61	94.00	99.02	97.29	85.78	93.26	98.63	95.97	81.16	90.78	98.34	96.17	76.97	89.46
	10^6	99.56	97.97	86.20	93.85	98.96	97.08	84.95	92.84	98.49	95.97	81.99	91.08	98.36	96.19	78.64	90.05
0.4	1	99.17	97.77	86.59	**93.86**	99.11	97.56	77.83	90.65	98.81	97.33	71.96	**88.45**	97.95	97.54	66.11	**86.54**
	10	99.22	97.54	85.74	93.41	99.17	97.08	79.06	90.76	98.96	95.72	73.22	87.85	98.73	94.58	68.21	85.35
	10^2	99.45	97.75	84.93	93.26	99.22	96.87	79.47	**90.79**	99.04	94.83	74.87	87.85	98.67	92.99	69.44	84.75
	10^3	99.53	97.97	83.22	92.81	99.32	97.52	76.54	90.18	99.15	95.51	72.79	87.56	98.68	93.24	68.62	84.62
	10^4	99.63	98.20	81.97	92.52	99.43	97.75	75.29	89.89	99.14	95.95	72.37	87.70	98.75	93.24	69.04	84.77
	10^5	99.61	98.20	81.99	92.53	99.48	97.97	74.46	89.74	99.15	95.95	71.96	87.55	98.67	93.47	69.04	84.92
	10^6	99.66	98.45	80.33	92.11	99.46	97.75	75.31	89.89	99.20	96.19	72.79	88.00	98.70	93.24	69.04	84.77
0.5	1	99.30	97.77	79.89	91.51	99.27	97.08	69.00	87.26	99.11	96.63	57.30	82.87	98.39	95.49	49.37	**79.35**
	10	99.43	97.99	80.71	**91.94**	99.38	97.08	70.69	**87.85**	99.28	95.04	61.07	**83.15**	98.68	91.86	49.78	77.13
	10^2	99.84	97.35	75.71	89.78	99.50	96.44	67.34	86.26	99.35	94.39	58.59	81.86	98.85	91.44	50.20	77.01
	10^3	99.85	97.35	72.37	88.61	99.77	96.67	64.00	85.24	99.30	94.41	58.57	81.87	98.94	91.89	51.05	77.60
	10^4	99.95	98.01	66.47	86.97	99.84	97.35	61.92	84.95	99.48	95.55	58.57	82.61	98.91	91.91	51.90	77.91
	10^5	100	98.01	66.05	86.83	99.84	96.89	61.92	84.66	99.48	95.32	58.97	82.60	98.94	91.23	51.90	77.47
	10^6	100	98.01	66.05	86.83	99.85	97.12	61.09	84.51	99.59	95.55	59.40	82.90	98.93	92.10	51.07	77.74

Table 2.11: Crisp rule, p =0.2, 0.3, 0.4, 0.5 and k =0.2, 0.5, 0.75, 1

FUZZY	k	0				0.01				0.05				0.1			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.01	1	97.50	97.31	97.50	97.38	97.50	97.31	97.50	97.38	97.50	97.31	97.50	97.38	97.50	97.31	97.08	97.23
	10	97.53	97.10	97.92	97.39	97.51	97.10	97.92	97.39	97.50	97.10	97.92	97.39	97.50	97.10	97.92	97.39
	10^2	97.61	97.31	97.92	**97.52**	97.62	97.31	97.92	**97.52**	97.59	97.31	97.92	**97.52**	97.62	97.31	97.92	**97.52**
	10^3	97.64	97.33	97.08	97.24	97.67	97.10	97.50	97.24	97.64	97.10	97.92	97.39	97.67	97.10	97.92	97.39
	10^4	97.67	97.33	97.08	97.24	97.64	97.10	97.50	97.24	97.66	97.54	96.67	97.23	97.66	97.10	97.92	97.39
	10^5	97.66	97.33	97.50	97.39	97.64	97.54	96.67	97.23	97.64	97.31	97.92	**97.52**	97.67	97.54	96.67	97.23
	10^6	97.71	97.33	96.67	97.10	97.71	97.33	96.67	97.10	97.62	97.54	97.08	97.38	97.61	97.33	97.08	97.24
0.05	1	97.61	97.10	96.25	96.80	97.63	97.10	96.25	96.80	97.63	97.10	96.25	96.80	97.64	97.10	96.25	96.80
	10	97.61	97.10	97.92	97.39	97.63	97.10	97.92	97.39	97.61	97.10	97.92	97.39	97.63	97.10	97.92	97.39
	10^2	97.72	97.33	97.50	97.39	97.67	97.31	97.92	97.52	97.72	97.10	97.92	97.39	97.69	97.10	97.92	97.39
	10^3	97.74	97.54	97.50	**97.52**	97.67	97.54	97.92	**97.67**	97.67	97.33	97.92	**97.54**	97.67	97.33	97.92	97.54
	10^4	97.71	97.54	97.50	**97.52**	97.71	97.54	97.50	97.52	97.74	97.54	97.50	97.52	97.77	97.33	97.92	97.54
	10^5	97.74	97.54	97.50	**97.52**	97.71	97.54	97.50	97.52	97.74	97.54	97.50	97.52	97.74	97.33	97.08	97.24
	10^6	97.72	97.54	97.50	**97.52**	97.76	97.54	97.08	97.38	97.72	97.54	97.08	97.38	97.74	97.54	97.92	**97.67**
0.10	1	97.87	97.10	95.83	96.66	97.87	97.10	95.83	96.66	97.86	97.10	95.42	96.51	97.89	97.10	95.42	96.51
	10	98.03	97.31	97.50	**97.38**	98.02	97.31	97.50	**97.38**	98.00	97.31	97.50	**97.38**	98.00	97.10	97.08	**97.10**
	10^2	98.11	97.33	97.08	97.24	98.15	97.14	97.08	97.12	98.10	97.15	97.08	97.12	98.13	97.38	96.25	96.98
	10^3	98.18	97.54	95.83	96.94	98.10	97.54	96.67	97.23	98.03	97.54	96.67	97.23	98.05	97.33	95.42	96.66
	10^4	98.13	97.54	95.83	96.94	98.11	97.54	96.25	97.09	98.16	97.54	96.25	97.09	98.13	97.33	95.83	96.81
	10^5	98.11	97.54	96.25	97.09	98.13	97.54	95.83	96.94	98.13	97.54	95.83	96.94	98.03	97.33	96.25	96.95
	10^6	98.13	97.54	95.83	96.94	98.11	97.54	96.25	97.09	98.08	97.54	95.83	96.94	98.10	97.33	96.25	96.95
0.15	1	98.10	97.31	94.57	96.35	98.08	97.31	94.57	96.35	98.06	97.39	94.57	96.40	98.01	97.49	94.57	96.46
	10	98.15	97.54	95.40	96.79	98.15	97.31	95.40	96.64	98.11	97.23	95.40	**96.59**	98.10	97.42	94.98	96.57
	10^2	98.29	97.54	96.23	**97.08**	98.34	97.54	96.23	**97.08**	98.34	97.31	94.98	96.50	98.29	97.31	95.40	**96.64**
	10^3	98.52	97.54	93.73	96.21	98.59	97.54	93.73	96.21	98.53	97.54	94.57	96.50	98.50	97.50	94.15	96.33
	10^4	98.62	97.54	93.32	96.06	98.65	97.54	93.73	96.21	98.63	97.54	92.48	95.77	98.58	97.53	93.32	96.05
	10^5	98.62	97.54	92.48	95.77	98.65	97.54	93.32	96.06	98.63	97.54	92.90	95.91	98.61	97.54	92.90	95.91
	10^6	98.62	97.54	92.90	95.91	98.54	97.54	92.90	95.91	98.45	97.54	94.57	96.50	98.57	97.54	92.90	95.91

Table 2.12: Fuzzy rule, p =0.01, 0.05, 0.1, 0.15 and k =0, 0.01, 0.05, 0.1

FUZZY	k	0.2				0.5				0.75				1			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.01	1	97.50	97.31	97.08	97.23	97.51	97.31	97.08	97.23	97.51	97.31	97.08	97.23	97.48	97.31	97.08	97.23
	10	97.50	97.31	97.92	97.52	97.49	97.06	97.92	97.36	97.53	97.03	97.92	97.34	97.54	97.01	97.92	97.33
	10^2	97.59	97.30	97.92	97.51	97.58	97.29	97.50	97.37	97.63	97.44	97.92	**97.60**	97.58	97.21	97.92	**97.45**
	10^3	97.64	97.54	97.50	**97.52**	97.66	97.27	97.50	97.35	97.66	97.44	97.50	97.46	97.67	97.44	97.08	97.32
	10^4	97.66	97.52	97.50	97.51	97.72	97.52	97.50	**97.52**	97.72	97.49	97.08	97.35	97.66	97.47	96.67	97.19
	10^5	97.67	97.31	97.50	97.38	97.71	97.27	97.08	97.20	97.67	97.50	95.83	96.91	97.69	97.45	96.67	97.18
	10^6	97.74	97.33	97.08	97.24	97.66	97.52	97.08	97.37	97.72	97.50	96.25	97.06	97.67	97.45	96.25	97.03
0.05	1	97.64	97.10	96.25	96.80	97.54	97.31	95.83	96.79	97.45	97.29	95.83	96.78	97.32	97.27	96.25	96.91
	10	97.63	97.10	97.92	97.39	97.63	97.10	97.08	97.10	97.61	97.05	97.08	97.06	97.53	96.99	97.08	97.03
	10^2	97.72	97.10	97.50	97.24	97.72	97.10	97.50	97.24	97.72	97.22	97.92	**97.47**	97.69	97.36	97.08	**97.26**
	10^3	97.72	97.33	97.50	97.39	97.72	97.31	97.50	**97.38**	97.71	97.24	97.50	97.33	97.77	97.09	97.08	97.09
	10^4	97.72	97.33	97.92	**97.54**	97.79	97.31	97.08	97.23	97.76	96.95	96.67	96.85	97.77	97.11	96.67	96.96
	10^5	97.72	97.33	97.92	**97.54**	97.79	97.30	97.08	97.22	97.74	97.00	97.08	97.03	97.72	97.09	96.67	96.94
	10^6	97.76	97.33	97.08	97.24	97.77	97.31	97.50	97.38	97.79	97.45	97.08	97.32	97.79	97.10	96.67	96.95
0.10	1	97.91	97.10	95.42	96.51	97.89	97.09	95.00	96.36	97.65	97.16	95.42	96.55	97.65	96.79	93.73	95.72
	10	97.98	97.30	96.67	97.08	97.86	96.99	96.67	96.88	97.73	96.86	96.25	**96.65**	97.64	96.63	95.42	**96.21**
	10^2	98.02	97.18	96.25	96.85	97.93	96.94	97.08	**96.99**	97.75	96.86	95.00	96.21	97.62	96.39	94.17	95.61
	10^3	98.06	97.25	95.83	96.75	97.96	97.34	95.83	96.81	97.79	97.09	95.42	96.51	97.67	96.37	94.17	95.60
	10^4	98.03	97.45	96.67	**97.18**	97.82	97.16	96.25	96.84	97.82	96.98	95.83	96.58	97.67	96.51	93.73	95.54
	10^5	98.02	97.46	96.25	97.04	97.85	97.13	96.67	96.97	97.83	96.82	96.25	96.62	97.71	96.48	94.17	95.67
	10^6	98.02	97.44	95.83	96.88	97.89	97.13	96.67	96.97	97.82	96.58	96.25	96.47	97.70	96.46	94.17	95.65
0.15	1	98.01	97.44	93.73	96.14	97.85	97.22	92.90	95.71	97.57	96.75	91.23	94.82	97.21	96.23	90.82	**94.33**
	10	98.08	97.51	94.57	**96.48**	98.05	96.79	93.73	**95.72**	97.94	96.29	92.48	**94.96**	97.55	95.92	90.40	93.99
	10^2	98.24	97.26	94.57	96.32	98.12	96.91	93.32	95.65	97.95	95.76	92.07	94.47	97.90	94.91	90.40	93.33
	10^3	98.33	97.45	94.57	96.44	98.22	96.90	93.32	95.65	98.01	96.16	91.65	94.58	97.89	94.91	90.82	93.48
	10^4	98.51	97.53	94.15	96.35	98.37	97.14	92.48	95.51	98.03	95.97	92.48	94.75	97.87	94.48	90.82	93.20
	10^5	98.50	97.54	93.73	96.21	98.31	97.12	92.48	95.50	98.02	95.90	92.07	94.56	97.92	94.52	90.40	93.07
	10^6	98.54	97.52	93.32	96.05	98.29	96.70	92.48	95.22	98.02	95.80	92.48	94.64	97.90	94.74	90.40	93.22

Table 2.13: Fuzzy rule, p =0.01, 0.05, 0.1, 0.15 and k =0.2, 0.5, 0.75, 1

FUZZY	k	0				0.01				0.05				0.1			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.2	1	98.24	97.54	94.57	**96.50**	98.24	97.57	94.57	**96.52**	98.24	97.68	94.57	**96.59**	98.24	97.60	94.57	**96.54**
	10	98.42	97.54	94.57	**96.50**	98.43	97.54	94.57	96.50	98.43	97.33	94.15	96.22	98.39	97.33	94.57	96.36
	10^2	98.73	97.54	92.90	95.91	98.70	97.49	92.07	95.59	98.69	97.54	93.32	96.06	98.68	97.42	93.32	95.98
	10^3	98.93	97.29	91.21	95.16	98.86	97.31	92.90	95.77	98.91	97.11	91.23	95.05	98.85	97.15	91.21	95.07
	10^4	99.01	97.52	90.80	95.17	98.99	97.52	91.63	95.46	99.02	97.66	90.80	95.26	98.94	97.38	90.80	95.08
	10^5	99.06	97.75	90.38	95.17	99.02	97.52	91.63	95.46	99.01	97.31	90.40	94.89	99.02	97.48	89.96	94.85
	10^6	99.02	97.75	92.46	95.90	99.03	97.75	91.63	95.61	99.02	97.43	91.23	95.26	99.00	97.41	90.80	95.10
0.3	1	98.98	97.33	93.73	96.07	98.98	97.33	93.73	**96.07**	98.98	97.38	93.73	**96.10**	98.98	97.42	93.32	**95.98**
	10	98.98	97.77	93.32	**96.21**	98.93	97.77	92.48	95.92	98.92	97.54	92.07	95.62	98.94	97.59	90.40	95.07
	10^2	99.46	97.97	89.98	95.18	99.43	97.84	90.82	95.38	99.34	97.84	90.82	95.38	99.33	97.84	91.23	95.53
	10^3	99.58	97.97	89.98	95.18	99.58	97.97	90.82	95.47	99.58	97.97	90.40	95.32	99.55	97.86	89.55	94.95
	10^4	99.58	97.97	89.11	94.87	99.58	97.97	89.13	94.88	99.58	98.17	89.11	95.00	99.58	98.09	87.45	94.37
	10^5	99.59	97.97	89.11	94.87	99.59	97.97	89.11	94.87	99.58	98.14	88.70	94.83	99.58	98.12	88.28	94.68
	10^6	99.59	97.97	89.11	94.87	99.59	98.20	88.70	94.87	99.56	97.97	88.70	94.73	99.56	97.96	87.86	94.43
0.4	1	99.17	97.77	93.32	**96.21**	99.17	97.77	92.90	**96.06**	99.18	97.76	92.46	**95.91**	99.17	97.57	89.51	**94.75**
	10	99.33	98.22	88.68	94.88	99.33	97.96	89.95	95.15	99.28	97.85	89.11	94.79	99.21	97.86	88.26	94.50
	10^2	99.50	98.20	88.26	94.72	99.48	97.99	89.11	94.89	99.48	97.72	87.86	94.27	99.50	97.92	85.78	93.67
	10^3	99.77	98.88	82.01	92.98	99.77	98.72	81.58	92.72	99.67	98.66	84.08	93.55	99.63	98.25	83.22	92.99
	10^4	99.76	98.66	82.01	92.83	99.77	98.66	80.72	92.38	99.71	98.66	81.14	92.53	99.66	98.56	81.99	92.77
	10^5	99.80	98.86	81.58	92.81	99.82	98.66	81.16	92.53	99.79	98.66	80.74	92.39	99.74	98.66	79.04	91.80
	10^6	99.84	98.86	81.99	92.96	99.82	98.86	81.58	92.81	99.80	98.66	79.89	92.09	99.70	98.62	81.12	92.50
0.5	1	99.40	97.77	87.88	**94.31**	99.40	97.77	87.45	**94.15**	99.38	97.68	85.76	**93.51**	99.38	97.33	84.08	**92.69**
	10	99.51	97.77	85.76	93.56	99.51	97.77	85.76	93.56	99.49	97.96	84.09	93.11	99.46	97.71	82.39	92.35
	10^2	99.85	97.80	80.36	91.70	99.85	97.98	79.93	91.66	99.85	97.56	79.11	91.11	99.85	97.40	77.83	90.55
	10^3	99.89	97.33	75.33	89.63	99.89	97.56	74.91	89.63	99.89	97.77	72.37	88.88	99.87	97.54	72.37	88.74
	10^4	100	97.54	73.66	89.18	100	97.54	72.83	88.89	100	97.39	71.56	88.35	100	97.41	69.86	87.77
	10^5	100	97.54	74.08	89.33	100	97.54	72.83	88.89	100	97.39	71.56	88.35	100	97.14	69.86	87.59
	10^6	100	97.54	73.66	89.18	100	97.54	72.83	88.89	100	97.39	71.56	88.35	100	97.14	69.86	87.59

Table 2.14: Fuzzy rule, p =0.2, 0.3, 0.4, 0.5 and k =0, 0.01, 0.05, 0.1

FUZZY	k	0.2				0.5				0.75				1			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.2	1	98.24	97.55	93.73	96.22	98.03	96.98	91.65	**95.12**	97.80	96.07	89.96	**93.93**	97.32	95.03	87.84	**92.51**
	10	98.32	97.24	94.15	96.16	98.17	96.17	92.90	95.03	98.12	95.23	91.23	93.83	97.78	94.10	89.57	92.51
	10^2	98.63	97.39	94.15	**96.26**	98.37	96.03	91.65	94.50	98.22	94.63	89.96	93.00	98.02	93.25	87.88	91.37
	10^3	98.84	97.13	92.07	95.36	98.47	95.62	90.78	93.93	98.25	94.28	88.28	92.18	97.95	92.91	87.86	91.14
	10^4	98.80	97.39	90.38	94.94	98.55	95.98	88.70	93.43	98.30	94.04	88.30	92.03	97.97	92.94	87.03	90.87
	10^5	98.82	97.33	91.65	95.34	98.55	95.73	89.53	93.56	98.29	93.90	88.30	91.94	98.00	92.83	87.45	90.94
	10^6	98.82	97.27	90.80	95.00	98.52	95.98	88.70	93.43	98.31	94.09	87.88	91.91	98.04	92.60	85.78	90.21
0.3	1	98.91	97.29	91.23	**95.17**	98.64	96.01	87.90	**93.17**	98.32	94.03	82.01	**89.82**	97.67	91.47	78.21	**86.83**
	10	98.87	97.54	90.40	95.04	98.70	95.65	88.32	93.08	98.51	92.78	84.13	89.75	98.30	88.14	80.34	85.42
	10^2	99.14	97.61	89.98	94.94	98.90	95.02	86.63	92.09	98.54	92.03	82.41	88.66	98.26	88.20	79.06	85.00
	10^3	99.47	97.62	88.32	94.36	98.93	95.16	85.38	91.74	98.59	92.02	82.83	88.80	98.36	87.96	77.39	84.26
	10^4	99.56	98.03	87.03	94.18	98.97	95.35	85.78	92.00	98.59	91.75	81.58	88.19	98.28	88.14	78.22	84.67
	10^5	99.57	97.91	86.61	93.96	99.01	95.73	85.78	92.25	98.65	91.68	81.16	88.00	98.34	88.45	76.97	84.44
	10^6	99.55	97.84	86.20	93.77	98.97	95.53	84.95	91.83	98.54	91.79	81.99	88.36	98.40	87.99	78.64	84.72
0.4	1	99.17	96.85	86.59	93.26	99.12	93.66	77.83	88.12	98.82	89.56	71.96	**83.40**	97.91	85.64	66.11	**78.80**
	10	99.23	97.36	85.74	**93.29**	99.20	93.35	79.06	**88.35**	98.95	87.47	73.22	82.48	98.69	82.95	68.21	77.79
	10^2	99.45	97.21	84.93	92.91	99.21	93.04	79.47	88.29	99.05	87.12	74.87	82.84	98.61	81.68	69.44	77.39
	10^3	99.52	97.65	83.22	92.60	99.31	93.27	76.54	87.41	99.14	87.74	72.79	82.51	98.64	81.83	68.62	77.21
	10^4	99.61	97.82	81.97	92.27	99.43	93.94	75.29	87.41	99.17	88.21	72.37	82.67	98.68	81.75	69.04	77.30
	10^5	99.61	97.89	81.99	92.33	99.46	93.95	74.46	87.13	99.17	87.99	71.96	82.38	98.61	82.00	69.04	77.46
	10^6	99.66	98.28	80.33	92.00	99.45	93.90	75.31	87.39	99.18	88.23	72.79	82.82	98.65	81.78	69.04	77.32
0.5	1	99.30	96.64	79.89	90.78	99.27	89.68	69.00	82.44	99.07	83.50	57.30	**74.33**	98.33	78.66	49.37	**68.41**
	10	99.44	96.65	80.71	**91.07**	99.39	89.90	70.69	**83.18**	99.25	81.21	61.07	74.16	98.55	75.29	49.78	66.37
	10^2	99.82	96.44	75.71	89.18	99.48	89.24	67.34	81.58	99.34	81.79	58.59	73.67	98.73	74.78	50.20	66.18
	10^3	99.85	96.06	72.37	87.77	99.76	88.84	64.00	80.15	99.34	82.08	58.57	73.86	98.83	74.71	51.05	66.43
	10^4	99.95	96.26	66.47	85.84	99.83	88.91	61.92	79.47	99.55	81.39	58.57	73.41	98.83	74.81	51.90	66.79
	10^5	100	96.28	66.05	85.70	99.84	88.29	61.92	79.07	99.53	81.39	58.97	73.54	98.87	75.01	51.90	66.92
	10^6	100	96.28	66.05	85.70	99.85	88.57	61.09	78.95	99.59	81.36	59.40	73.68	98.84	74.99	51.07	66.62

Table 2.15: Fuzzy rule, p =0.2, 0.3, 0.4, 0.5 and k =0.2, 0.5, 0.75, 1

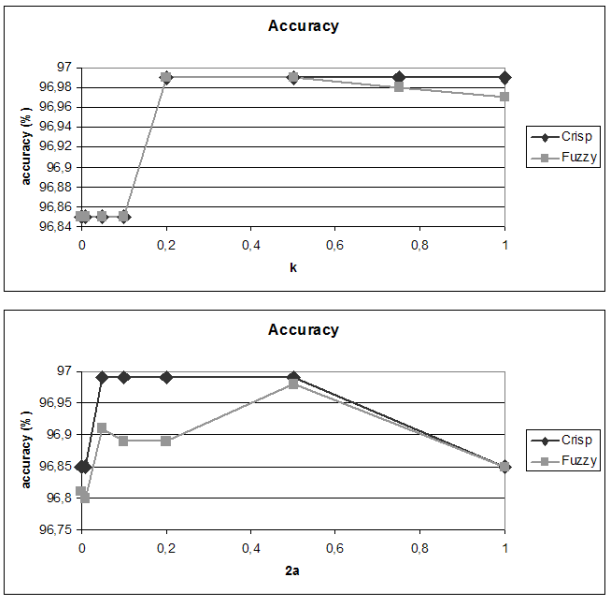

Figure 2.10: Accuracy for the database with its missing values. Up: interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$. Down: interval $[Q_a, Q_{1-a}]$

2.5.4 Computational experiment for the database with its missing values

In this experiment, we consider the complete 'breast-cancer' dataset (699 instances), including the 16 instances with missing values. All the missing values appear in the sixth variable.

To impute the missing values, we have built an interval following the two strategies explained for the previous experiments, i.e., the blank is replaced by the interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$ or by $[Q_a, Q_{1-a}]$, for the same values of k and $2a$ as those used in the other experiments.

The classification problem via 10-fold cross validation (see [66]) has been posed and each optimization problem (2.17), for different values of C, with $C_+ = C_- = C$, has been solved with LOQO, [112]. The accuracy following the crisp and the fuzzy rule are presented in the Tables 2.24-2.25, and the best results for each k and a are shown in bold and are depicted in Figure 2.10.

In this case, we can observe that, for the two types of intervals (and the two classification rules), the best results are obtained for non-degenerate intervals. In the case of imputing by $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$ (see Table 2.24), one observes that we obtain the best results in terms of accuracy for high values of k. It means that, in this case, it is better to take into account the value of the standard deviation when imputing the

CRISP	$2a$	Inner fences				0				0.01				0.05			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.01	1	97.40	97.31	97.08	97.23	97.19	97.54	96.25	97.09	97.17	97.54	96.67	**97.23**	97.36	97.10	97.08	97.10
	10	97.46	97.31	97.50	**97.38**	97.38	97.08	97.50	**97.23**	97.35	96.88	97.50	97.09	97.40	96.88	97.50	97.09
	10^2	97.54	97.31	97.08	97.23	97.46	97.08	97.50	**97.23**	97.51	97.08	97.08	97.08	97.53	97.08	97.92	**97.37**
	10^3	97.59	97.54	96.25	97.09	97.59	97.31	96.67	97.09	97.58	97.08	97.08	97.08	97.62	97.08	97.50	97.23
	10^4	97.76	97.54	95.83	96.94	97.58	97.08	96.67	96.94	97.56	97.08	96.67	96.94	97.71	97.31	97.08	97.23
	10^5	97.67	97.54	95.83	96.94	97.54	97.08	96.67	96.94	97.54	97.08	97.08	97.08	97.66	97.08	97.08	97.08
	10^6	97.69	97.54	96.25	97.09	97.46	97.08	97.08	97.08	97.62	97.08	95.83	96.65	97.66	97.08	97.08	97.08
0.05	1	97.06	97.08	95.83	96.65	96.84	97.54	95.42	**96.80**	96.86	97.31	95.83	96.79	97.07	96.88	95.42	96.36
	10	97.28	97.31	96.67	**97.09**	97.01	97.08	96.25	96.79	97.02	96.88	96.25	96.66	97.36	96.67	96.67	96.67
	10^2	97.54	97.31	96.25	96.94	97.10	97.08	96.25	96.79	97.17	96.88	96.67	**96.80**	97.49	96.65	96.67	96.65
	10^3	97.51	97.31	96.67	**97.09**	97.02	97.08	96.25	96.79	97.17	96.88	96.67	**96.80**	97.48	96.65	97.08	96.80
	10^4	97.59	97.08	96.67	96.94	97.06	97.08	96.25	96.79	97.22	96.88	95.83	96.51	97.56	97.10	96.25	96.80
	10^5	97.56	97.31	96.25	96.94	97.09	97.08	96.25	96.79	97.17	96.88	96.67	**96.80**	97.56	97.10	97.08	**97.10**
	10^6	97.63	97.31	96.67	**97.09**	97.06	97.08	96.25	96.79	97.19	96.88	95.42	96.36	97.61	97.10	97.08	**97.10**
0.10	1	94.94	96.88	88.30	93.87	95.45	97.08	90.80	**94.88**	95.62	96.40	92.05	94.88	96.62	95.76	94.17	95.20
	10	95.48	96.42	90.40	94.31	95.74	96.63	91.21	94.73	95.95	96.40	92.48	95.03	97.22	95.30	95.42	**95.34**
	10^2	95.64	96.65	90.82	**94.61**	95.92	96.63	91.63	94.88	96.03	96.40	92.90	95.18	97.20	94.62	95.83	95.05
	10^3	95.71	96.65	89.98	94.32	95.95	96.63	91.63	94.88	96.06	96.40	92.90	95.18	97.19	94.39	95.83	94.90
	10^4	95.67	96.65	89.98	94.32	95.95	96.63	91.63	94.88	96.08	96.63	93.32	**95.47**	97.23	95.08	95.83	95.34
	10^5	95.66	96.42	90.40	94.31	95.95	96.63	91.63	94.88	96.06	96.40	93.32	95.32	97.20	95.08	95.42	95.20
	10^6	95.64	96.65	90.82	**94.61**	95.95	96.63	91.63	94.88	96.05	96.63	92.90	95.32	97.22	95.30	95.42	**95.34**
0.15	1	92.31	96.88	81.14	91.37	93.51	97.08	85.36	**92.98**	93.51	97.08	85.36	**92.98**	95.87	94.38	89.98	**92.84**
	10	93.56	95.97	83.68	91.67	93.36	96.86	85.78	92.98	93.36	96.86	85.78	92.98	96.21	91.69	91.23	91.53
	10^2	93.61	95.51	85.36	**91.96**	93.38	97.08	85.36	**92.98**	93.38	97.08	85.36	**92.98**	96.19	90.55	91.23	90.79
	10^3	93.57	95.51	84.95	91.81	93.36	97.08	85.36	**92.98**	93.36	97.08	85.36	**92.98**	96.24	89.64	91.65	90.34
	10^4	93.57	95.28	85.36	91.81	93.36	97.08	85.36	**92.98**	93.36	97.08	85.36	**92.98**	96.21	89.64	91.65	90.34
	10^5	93.57	95.28	85.36	91.81	93.36	97.08	85.36	**92.98**	93.36	97.08	85.36	**92.98**	96.29	89.64	91.23	90.20
	10^6	93.57	95.28	85.36	91.81	93.36	97.08	85.36	**92.98**	93.36	97.08	85.36	**92.98**	96.27	89.64	92.07	90.49

Table 2.16: Crisp rule, p =0.01, 0.05, 0.1, 0.15 and $2a$ =0, 0.01, 0.05, inner fences

CRISP	$2a$	0.1				0.2				0.5				1			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.01	1	97.36	97.31	97.08	97.23	97.53	97.31	97.08	97.23	97.49	97.31	97.08	97.23	97.48	97.31	97.08	97.23
	10	97.49	96.88	97.50	97.09	97.53	97.31	97.92	**97.52**	97.53	97.31	97.92	**97.52**	97.48	97.10	97.92	97.39
	10^2	97.58	97.08	97.92	97.37	97.59	97.31	97.92	**97.52**	97.59	97.31	97.50	97.38	97.64	97.31	97.92	**97.52**
	10^3	97.69	97.54	97.50	**97.52**	97.72	97.54	97.08	97.38	97.66	97.54	97.08	97.38	97.64	97.33	97.50	97.39
	10^4	97.69	97.54	97.50	**97.52**	97.71	97.54	97.08	97.38	97.71	97.54	96.67	97.23	97.66	97.33	97.08	97.24
	10^5	97.67	97.54	97.50	**97.52**	97.72	97.54	96.67	97.23	97.72	97.54	97.08	97.38	97.67	97.10	97.50	97.24
	10^6	97.71	97.54	96.67	97.23	97.71	97.31	97.08	97.23	97.76	97.54	97.08	97.38	97.64	97.33	97.08	97.24
0.05	1	97.19	97.10	96.25	96.80	97.25	97.10	96.25	96.80	97.46	97.10	96.25	96.80	97.64	97.10	95.83	96.66
	10	97.56	96.88	97.08	96.95	97.51	96.88	97.08	96.95	97.59	97.31	97.08	97.23	97.61	97.10	97.92	97.39
	10^2	97.66	96.67	97.08	96.81	97.64	96.88	97.50	97.09	97.72	97.33	97.50	**97.39**	97.76	97.31	97.50	97.38
	10^3	97.74	97.10	96.25	96.80	97.71	97.33	97.08	97.24	97.72	97.10	97.08	97.10	97.79	97.33	97.08	97.24
	10^4	97.77	97.10	96.25	96.80	97.64	97.10	97.08	97.10	97.74	97.31	97.50	97.38	97.79	97.10	97.92	97.39
	10^5	97.61	97.10	96.67	**96.95**	97.67	97.31	97.08	97.23	97.76	97.10	97.08	97.10	97.80	97.10	97.08	97.10
	10^6	97.71	96.89	96.25	96.67	97.61	97.31	97.50	**97.38**	97.82	97.10	97.50	97.24	97.74	97.33	97.50	**97.39**
0.10	1	97.06	95.74	95.42	95.63	97.30	96.19	95.83	96.07	97.72	97.10	95.83	96.66	97.95	97.56	95.00	96.66
	10	97.28	95.30	96.67	**95.78**	97.48	95.74	96.25	95.92	97.80	97.31	97.08	**97.23**	98.10	97.33	96.67	**97.10**
	10^2	97.41	95.53	95.83	95.64	97.46	96.42	96.25	96.36	97.84	97.31	95.83	96.79	98.16	97.31	96.25	96.94
	10^3	97.51	95.30	95.83	95.49	97.49	96.19	95.83	96.07	97.82	97.10	97.08	97.10	98.13	97.54	96.25	97.09
	10^4	97.51	95.53	95.42	95.49	97.48	96.65	95.42	96.22	97.90	97.33	96.67	97.10	98.10	97.54	96.25	97.09
	10^5	97.51	94.85	95.42	95.05	97.54	96.21	95.83	96.08	97.84	97.10	95.83	96.66	98.15	97.54	95.83	96.94
	10^6	97.54	95.06	95.83	95.33	97.49	96.65	95.83	**96.36**	97.87	96.88	95.83	96.51	98.18	97.54	96.25	97.09
0.15	1	96.37	93.92	92.48	**93.42**	96.68	95.28	93.32	**94.60**	97.87	97.31	93.73	**96.06**	97.90	97.54	92.48	95.77
	10	96.80	91.69	92.07	91.82	97.43	94.60	92.07	93.71	98.08	96.86	93.32	95.62	98.26	97.54	92.90	**95.91**
	10^2	97.14	89.60	91.65	90.32	97.49	93.69	92.07	93.12	98.02	97.31	92.90	95.77	98.55	97.77	91.65	95.62
	10^3	97.19	90.06	92.07	90.76	97.51	94.62	91.23	93.44	97.97	97.08	92.48	95.47	98.93	97.99	89.57	95.04
	10^4	97.28	89.38	91.65	90.17	97.41	95.04	91.23	93.71	97.97	97.08	92.48	95.47	98.93	98.20	89.15	95.03
	10^5	97.15	89.83	91.65	90.47	97.43	94.13	91.65	93.26	98.08	97.31	92.90	95.77	98.89	98.20	89.57	95.18
	10^6	97.23	89.38	91.65	90.17	97.38	93.94	91.23	92.99	98.03	97.31	92.48	95.62	98.86	98.20	89.57	95.18

Table 2.17: Crisp rule, p =0.01, 0.05, 0.1, 0.15 and $2a$ =0.1, 0.2, 0.5, 1

CRISP	$2a$	Inner fences				0				0.01				0.05			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.2	1	90.26	95.51	74.46	**88.14**	91.77	96.40	80.74	**90.92**	92.35	95.06	80.31	**89.90**	95.61	91.67	87.46	**90.20**
	10	91.25	93.01	77.37	87.54	91.93	95.49	80.74	90.33	92.68	95.06	79.47	89.60	95.98	87.14	88.73	87.70
	10^2	91.43	92.56	77.37	87.24	91.90	95.27	80.33	90.04	92.83	95.06	79.89	89.75	96.05	85.78	89.57	87.10
	10^3	91.43	92.56	77.79	87.39	91.90	95.27	80.33	90.04	92.78	95.06	79.89	89.75	96.11	84.87	89.98	86.66
	10^4	91.41	92.35	77.79	87.25	91.90	95.27	80.33	90.04	92.78	95.06	79.89	89.75	96.14	85.09	89.15	86.51
	10^5	91.41	92.35	77.79	87.25	91.90	95.27	80.33	90.04	92.78	95.06	79.89	89.75	96.13	85.09	89.15	86.51
	10^6	91.41	92.35	77.79	87.25	91.90	95.27	80.33	90.04	92.78	95.06	79.89	89.75	96.13	84.87	89.57	86.51
0.3	1	84.02	93.90	54.78	**80.21**	86.07	93.45	63.61	**83.00**	86.11	92.33	62.30	81.82	94.21	81.70	79.09	**80.79**
	10	85.20	85.98	57.30	75.95	86.40	91.63	62.34	81.38	86.48	91.89	63.15	**81.84**	94.65	75.59	80.74	77.39
	10^2	85.29	84.17	57.74	74.92	86.45	91.40	61.94	81.09	86.45	91.44	63.15	81.54	94.71	73.33	82.03	76.38
	10^3	85.28	83.94	57.32	74.62	86.35	91.40	61.94	81.09	86.42	91.89	63.15	**81.84**	94.70	73.11	81.20	75.94
	10^4	85.26	83.71	57.32	74.48	86.35	91.40	61.94	81.09	86.42	91.89	63.15	**81.84**	94.70	73.11	81.63	76.09
	10^5	85.26	83.71	57.32	74.48	86.35	91.40	61.94	81.09	86.42	91.89	63.15	**81.84**	94.70	73.11	81.63	76.09
	10^6	85.26	83.71	57.32	74.48	86.35	91.40	61.94	81.09	86.42	91.89	63.15	**81.84**	94.70	73.11	81.63	76.09
0.4	1	80.59	95.25	42.68	**76.85**	77.00	83.69	34.28	**66.40**	78.62	79.92	38.50	**65.43**	89.72	70.83	62.37	**67.87**
	10	80.87	92.29	43.10	75.08	77.31	75.76	37.21	62.27	78.30	74.24	37.66	61.44	90.48	65.68	65.29	65.54
	10^2	80.92	90.93	43.10	74.19	77.14	72.37	37.21	60.06	78.30	73.79	37.25	61.00	90.74	64.55	65.29	64.81
	10^3	80.98	90.93	42.68	74.05	77.14	78.48	37.21	64.04	78.31	74.02	37.25	61.15	90.76	63.86	65.71	64.51
	10^4	80.98	90.93	42.68	74.05	77.14	78.03	37.21	63.75	78.36	73.56	37.25	60.85	90.78	63.86	65.71	64.51
	10^5	80.97	90.93	42.68	74.05	77.14	78.26	37.21	63.89	78.51	74.02	37.66	61.29	90.78	63.86	65.71	64.51
	10^6	80.97	90.93	42.68	74.05	77.14	78.03	37.21	63.75	78.31	74.02	37.25	61.15	90.78	63.86	65.71	64.51
0.5	1	80.74	99.32	23.84	**72.91**	39.31	5.21	93.48	36.10	77.86	55.21	38.06	49.21	84.94	51.04	46.01	**49.28**
	10	82.63	95.91	28.42	72.29	43.27	9.75	87.23	36.86	77.60	52.23	36.39	46.69	86.24	45.13	47.70	46.03
	10^2	82.98	95.45	28.41	71.99	43.27	9.75	87.23	36.86	77.60	52.03	36.39	46.56	86.30	43.56	48.95	45.45
	10^3	82.97	95.45	28.41	71.99	43.27	14.34	87.23	**39.84**	77.60	69.34	36.39	**57.81**	86.27	43.33	48.95	45.30
	10^4	82.98	95.45	28.41	71.99	43.27	14.34	87.23	**39.84**	77.60	63.47	36.39	53.99	86.27	43.33	48.95	45.30
	10^5	82.98	95.45	28.41	71.99	43.27	14.34	87.23	**39.84**	77.60	57.97	36.39	50.42	86.27	43.33	48.95	45.30
	10^6	82.98	95.45	28.41	71.99	43.27	12.25	87.23	38.49	77.60	54.55	36.39	48.19	86.27	43.33	48.95	45.30

Table 2.18: Crisp rule, p =0.2, 0.3, 0.4, 0.5 and $2a$ =0, 0.01, 0.05, inner fences

CRISP	$2a$	0.1				0.2				0.5				1			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.2	1	96.23	88.05	91.23	**89.16**	97.06	91.21	91.65	**91.36**	97.97	97.54	92.48	**95.77**	98.28	97.77	90.80	95.33
	10	97.01	84.87	92.07	87.39	97.38	90.30	92.07	90.92	98.18	97.08	92.48	95.47	98.45	97.77	91.21	**95.47**
	10^2	97.32	80.78	92.07	84.73	97.50	89.62	91.23	90.18	98.24	96.65	91.23	94.75	98.89	97.97	87.03	94.14
	10^3	97.32	81.69	92.48	85.46	97.46	90.74	91.65	91.06	98.41	96.19	90.80	94.31	99.09	97.95	84.95	93.40
	10^4	97.43	82.14	90.40	85.03	97.45	90.76	90.40	90.63	98.37	96.65	90.80	94.60	99.12	97.73	85.36	93.40
	10^5	97.33	81.00	91.23	84.58	97.43	90.53	90.40	90.48	98.36	96.65	89.55	94.16	99.15	97.95	82.43	92.52
	10^6	97.32	81.46	91.65	85.02	97.45	90.76	91.23	90.92	98.29	96.65	90.40	94.46	99.09	98.18	84.11	93.26
0.3	1	96.27	73.77	86.21	**78.12**	97.38	80.11	87.88	**82.83**	98.47	97.33	86.63	**93.59**	98.98	98.20	82.83	**92.82**
	10	97.01	67.90	87.46	74.74	97.76	75.13	87.88	79.59	98.41	96.65	87.46	93.43	99.15	98.20	81.99	92.53
	10^2	97.46	60.63	86.21	69.58	97.72	74.43	87.88	79.14	98.39	96.42	87.45	93.28	99.59	97.97	79.93	91.66
	10^3	97.49	61.31	86.21	70.02	97.74	75.11	87.46	79.44	98.57	96.42	86.20	92.84	99.61	98.66	78.22	91.51
	10^4	97.48	61.31	86.21	70.02	97.76	75.57	87.46	79.73	98.62	96.42	85.78	92.70	99.64	98.88	76.54	91.06
	10^5	97.53	61.76	86.21	70.32	97.72	74.89	87.05	79.14	98.62	96.65	85.80	92.85	99.63	98.88	76.99	91.22
	10^6	97.51	61.31	86.21	70.02	97.71	75.80	87.05	79.73	98.67	96.42	86.61	92.99	99.66	98.66	76.97	91.07
0.4	1	94.40	57.77	71.97	**62.74**	96.71	64.53	77.83	**69.18**	98.83	93.01	79.08	88.14	99.22	98.66	74.87	**90.33**
	10	96.34	47.82	76.16	57.74	97.51	52.75	80.34	62.40	98.89	91.19	79.08	86.95	99.48	98.66	72.79	89.60
	10^2	96.66	43.98	74.51	54.66	97.53	50.74	78.68	60.52	99.01	93.71	78.24	**88.30**	99.63	98.88	69.00	88.43
	10^3	96.68	43.98	73.68	54.37	97.46	49.39	77.84	59.35	99.07	92.80	74.08	86.25	99.80	98.88	64.84	86.97
	10^4	96.67	43.75	73.26	54.08	97.48	49.62	77.01	59.21	99.12	93.26	73.66	86.40	99.87	98.88	63.59	86.53
	10^5	96.67	43.52	73.68	54.07	97.49	50.28	78.26	60.07	99.07	92.78	74.06	86.23	99.87	98.88	63.59	86.53
	10^6	96.67	43.52	73.68	54.07	97.50	49.38	77.84	59.34	99.01	92.12	74.08	85.81	99.89	98.88	63.17	86.39
0.5	1	91.04	37.39	58.19	**44.67**	96.08	44.26	65.24	**51.60**	99.07	88.07	69.82	81.68	99.43	99.11	63.15	**86.53**
	10	92.83	27.73	62.37	39.85	97.69	28.94	68.19	42.67	99.28	89.03	70.25	**82.46**	99.66	99.11	60.24	85.51
	10^2	93.56	23.20	62.79	37.05	97.87	31.86	63.15	42.81	99.40	87.48	66.92	80.29	99.85	99.36	56.05	84.20
	10^3	93.66	23.24	61.94	36.78	97.95	32.80	65.65	44.30	99.37	83.39	64.82	76.89	99.89	99.34	53.13	83.17
	10^4	93.70	23.01	61.52	36.49	98.10	34.60	62.30	44.29	99.48	84.24	64.82	77.45	100	99.11	52.72	82.88
	10^5	93.70	22.78	61.52	36.34	97.90	34.15	64.38	44.73	99.53	84.24	65.24	77.59	100	99.11	53.13	83.02
	10^6	93.70	22.78	61.52	36.34	97.89	34.13	64.40	44.72	99.54	84.24	65.24	77.59	100	99.11	53.97	83.31

Table 2.19: Crisp rule, p =0.2, 0.3, 0.4, 0.5 and $2a$ =0.1, 0.2, 0.5, 1

FUZZY	$2a$	Inner fences				0				0.01				0.05			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.01	1	97.39	96.89	97.08	96.96	97.13	97.31	96.25	96.94	97.13	97.31	96.67	**97.08**	97.36	96.83	97.08	96.92
	10	97.43	96.94	97.50	**97.14**	97.30	97.00	97.50	97.18	97.28	96.79	97.50	97.04	97.39	96.77	97.50	97.02
	10^2	97.51	96.92	97.08	96.97	97.38	97.05	97.50	**97.21**	97.43	97.01	97.08	97.03	97.52	96.96	97.92	**97.30**
	10^3	97.56	97.18	96.25	96.85	97.52	97.21	96.67	97.02	97.51	97.05	97.08	97.06	97.62	97.01	97.50	97.18
	10^4	97.72	97.17	95.83	96.71	97.50	96.98	96.67	96.87	97.50	97.02	96.67	96.89	97.70	97.03	97.08	97.05
	10^5	97.64	97.15	95.83	96.69	97.47	97.03	96.67	96.91	97.48	97.01	97.08	97.04	97.65	96.99	97.08	97.02
	10^6	97.65	97.20	96.25	96.87	97.39	97.01	97.08	97.03	97.56	97.00	95.83	96.59	97.65	96.96	97.08	97.00
0.05	1	97.05	95.61	95.83	95.69	96.73	97.08	95.42	**96.50**	96.80	97.02	95.83	**96.60**	97.07	96.55	95.42	96.15
	10	97.27	95.81	96.67	**96.11**	96.86	96.59	96.25	96.47	96.88	96.43	96.25	96.36	97.36	96.20	96.67	96.37
	10^2	97.52	95.64	96.25	95.85	96.95	96.62	96.25	96.49	97.03	96.36	96.67	96.47	97.49	96.12	96.67	96.31
	10^3	97.49	95.71	96.67	96.05	96.90	96.50	96.25	96.41	97.05	96.33	96.67	96.45	97.48	95.99	97.08	96.37
	10^4	97.58	95.51	96.67	95.91	96.88	96.51	96.25	96.42	97.11	96.24	95.83	96.10	97.56	96.11	96.25	96.16
	10^5	97.54	95.61	96.25	95.83	96.92	96.53	96.25	96.43	97.05	96.29	96.67	96.42	97.56	96.16	97.08	**96.48**
	10^6	97.61	95.74	96.67	96.06	96.92	96.55	96.25	96.45	97.06	96.29	95.42	95.99	97.61	96.10	97.08	96.44
0.10	1	94.91	93.53	88.30	91.70	95.04	95.64	90.80	**93.95**	95.36	95.38	92.05	**94.21**	96.63	93.83	94.17	93.95
	10	95.42	92.20	90.40	91.57	95.24	95.20	91.21	93.81	95.62	94.80	92.48	93.99	97.20	93.20	95.42	**93.97**
	10^2	95.58	92.30	90.82	**91.78**	95.39	95.10	91.63	93.88	95.68	94.67	92.90	94.05	97.19	92.87	95.83	93.90
	10^3	95.65	92.26	89.98	91.46	95.42	95.07	91.63	93.87	95.70	94.65	92.90	94.04	97.19	92.65	95.83	93.77
	10^4	95.62	92.22	89.98	91.44	95.42	95.07	91.63	93.87	95.73	94.68	93.32	94.20	97.22	92.75	95.83	93.83
	10^5	95.60	91.88	90.40	91.36	95.42	95.07	91.63	93.87	95.70	94.64	93.32	94.18	97.18	92.81	95.42	93.72
	10^6	95.58	92.22	90.82	91.73	95.42	95.07	91.63	93.87	95.71	94.65	92.90	94.04	97.21	92.99	95.42	93.84
0.15	1	92.29	86.31	83.68	85.39	92.67	93.69	85.36	90.77	93.61	93.37	87.01	**91.14**	95.90	90.32	89.98	**90.20**
	10	93.49	85.81	85.36	85.65	92.51	93.67	85.78	**90.91**	93.66	92.99	87.03	90.91	96.19	88.29	91.23	89.32
	10^2	93.53	85.78	84.95	85.49	92.52	93.61	85.36	90.72	93.70	92.96	86.61	90.74	96.16	86.85	91.23	88.38
	10^3	93.51	85.72	85.36	85.60	92.51	93.61	85.36	90.72	93.72	92.94	86.61	90.73	96.22	86.39	91.65	88.23
	10^4	93.51	85.72	85.36	85.60	92.51	93.61	85.36	90.72	93.72	92.94	86.61	90.73	96.17	86.40	91.65	88.24
	10^5	93.51	85.72	85.36	85.60	92.51	93.61	85.36	90.72	93.72	92.94	86.61	90.73	96.26	86.05	91.23	87.86
	10^6	93.51	97.08	85.36	**92.98**	92.51	93.61	85.36	90.72	93.72	92.94	86.61	90.73	96.23	86.34	92.07	88.35

Table 2.20: Fuzzy rule, p =0.01, 0.05, 0.1, 0.15 and $2a$ =0, 0.01, 0.05, inner fences

FUZZY	$2a$	0.1				0.2				0.5				1			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.01	1	97.36	97.06	97.08	97.07	97.53	97.12	97.08	97.10	97.49	97.31	97.08	97.23	97.48	97.31	97.08	97.23
	10	97.49	96.77	97.50	97.03	97.53	97.08	97.92	97.37	97.53	97.26	97.92	**97.49**	97.48	97.10	97.92	97.39
	10^2	97.58	96.98	97.92	97.31	97.59	97.27	97.92	**97.50**	97.59	97.29	97.50	97.36	97.64	97.31	97.92	**97.52**
	10^3	97.69	97.25	97.50	97.34	97.72	97.30	97.08	97.22	97.66	97.50	97.08	97.35	97.64	97.33	97.50	97.39
	10^4	97.69	97.26	97.50	**97.34**	97.71	97.32	97.08	97.24	97.71	97.51	96.67	97.22	97.66	97.33	97.08	97.24
	10^5	97.67	97.25	97.50	97.34	97.72	97.28	96.67	97.07	97.72	97.53	97.08	97.37	97.67	97.10	97.50	97.24
	10^6	97.71	97.23	96.67	97.03	97.71	97.06	97.08	97.07	97.76	97.52	97.08	97.37	97.64	97.33	97.08	97.24
0.05	1	97.19	96.58	96.25	96.46	97.25	96.58	96.25	96.47	97.46	97.10	96.25	96.80	97.64	97.10	95.83	96.66
	10	97.56	96.29	97.08	**96.57**	97.51	96.48	97.08	96.69	97.59	97.31	97.08	97.23	97.61	97.10	97.92	97.39
	10^2	97.66	96.07	97.08	96.43	97.64	96.31	97.50	96.73	97.72	97.33	97.50	**97.39**	97.76	97.31	97.50	97.38
	10^3	97.74	96.37	96.25	96.33	97.71	96.46	97.08	96.68	97.72	97.10	97.08	97.09	97.79	97.33	97.08	97.24
	10^4	97.77	96.28	96.25	96.27	97.64	96.54	97.08	96.73	97.74	97.31	97.50	97.38	97.79	97.10	97.92	97.39
	10^5	97.61	96.17	96.67	96.34	97.67	96.68	97.08	96.82	97.76	97.10	97.08	97.09	97.80	97.10	97.08	97.10
	10^6	97.71	96.06	96.25	96.13	97.61	96.74	97.50	**97.01**	97.82	97.10	97.50	97.24	97.74	97.33	97.50	**97.39**
0.10	1	97.06	93.66	95.42	94.27	97.30	94.46	95.83	94.94	97.72	97.00	95.83	96.59	97.95	97.56	95.00	96.66
	10	97.28	93.02	96.67	**94.30**	97.48	94.08	96.25	94.84	97.80	97.13	97.08	**97.11**	98.10	97.33	96.67	**97.10**
	10^2	97.42	93.45	95.83	94.29	97.45	94.59	96.25	**95.17**	97.84	97.25	95.83	96.75	98.16	97.31	96.25	96.94
	10^3	97.51	92.89	95.83	93.92	97.49	94.76	95.83	95.14	97.82	97.08	97.08	97.08	98.13	97.54	96.25	97.09
	10^4	97.52	92.91	95.42	93.79	97.48	94.75	95.42	94.98	97.90	97.12	96.67	96.96	98.10	97.54	96.25	97.09
	10^5	97.50	92.83	95.42	93.74	97.54	94.61	95.83	95.04	97.84	97.04	95.83	96.62	98.15	97.54	95.83	96.94
	10^6	97.53	92.79	95.83	93.86	97.50	94.79	95.83	95.15	97.87	96.78	95.83	96.45	98.18	97.54	96.25	97.09
0.15	1	96.37	89.03	92.48	**90.23**	96.68	90.98	93.32	**91.80**	97.87	96.87	93.73	**95.77**	97.90	97.54	92.48	95.77
	10	96.79	87.11	92.07	88.85	97.43	90.21	92.07	90.86	98.08	96.67	93.32	95.49	98.26	97.54	92.90	**95.91**
	10^2	97.13	85.71	91.65	87.79	97.51	89.29	92.07	90.26	98.07	96.86	92.90	95.47	98.55	97.77	91.65	95.62
	10^3	97.17	85.56	92.07	87.84	97.52	89.80	91.23	90.30	97.99	96.61	92.48	95.17	98.93	97.99	89.57	95.04
	10^4	97.29	85.56	91.65	87.69	97.46	89.98	91.23	90.42	97.98	96.68	92.48	95.21	98.93	98.20	89.15	95.03
	10^5	97.15	85.97	91.65	87.96	97.49	89.66	91.65	90.35	98.10	96.74	92.90	95.39	98.89	98.20	89.57	95.18
	10^6	97.30	85.49	91.65	87.65	97.43	89.70	91.23	90.23	98.05	96.72	92.48	95.24	98.86	98.20	89.57	95.18

Table 2.21: Fuzzy rule, p =0.01, 0.05, 0.1, 0.15 and $2a$ =0.1, 0.2, 0.5, 1

FUZZY	$2a$	Inner fences				0				0.01				0.05			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.2	1	90.22	78.99	77.37	78.42	89.87	90.71	80.74	**87.22**	91.22	90.32	80.31	**86.82**	95.63	85.72	87.46	**86.33**
	10	91.19	78.55	77.37	78.14	89.85	90.01	80.74	86.76	91.39	89.67	79.47	86.10	95.88	83.04	88.73	85.03
	10^2	91.34	78.54	77.79	78.28	89.84	90.05	80.33	86.65	91.53	89.61	79.89	86.21	95.91	81.99	89.57	84.64
	10^3	91.33	78.51	77.79	78.26	89.84	90.08	80.33	86.67	91.48	89.60	79.89	86.20	95.98	80.92	89.98	84.09
	10^4	91.31	78.50	77.79	78.25	89.84	90.08	80.33	86.67	91.48	89.60	79.89	86.20	96.01	81.11	89.15	83.92
	10^5	91.31	78.50	77.79	78.25	89.84	90.08	80.33	86.67	91.48	89.60	79.89	86.20	96.00	80.96	89.15	83.83
	10^6	91.31	95.27	80.33	**90.04**	89.84	90.08	80.33	86.67	91.48	89.60	79.89	86.20	95.99	80.86	89.57	83.91
0.3	1	84.02	75.44	54.78	**68.21**	82.29	84.99	63.61	**77.51**	83.61	84.70	62.30	**76.86**	94.17	75.01	79.09	**76.44**
	10	85.20	70.67	57.30	65.99	82.52	84.46	62.34	76.72	83.80	84.05	63.15	76.74	94.53	71.34	80.74	74.63
	10^2	85.29	70.06	57.74	65.75	82.57	84.39	61.94	76.53	83.77	84.05	63.15	76.73	94.53	69.99	82.03	74.20
	10^3	85.28	70.02	57.32	65.57	82.48	84.43	61.94	76.56	83.75	84.05	63.15	76.74	94.50	69.73	81.20	73.74
	10^4	85.26	69.95	57.32	65.53	82.48	84.43	61.94	76.56	83.75	84.05	63.15	76.74	94.52	69.72	81.63	73.89
	10^5	85.26	69.95	57.32	65.53	82.48	84.43	61.94	76.56	83.75	84.05	63.15	76.74	94.52	69.72	81.63	73.89
	10^6	85.26	69.95	57.32	65.53	82.48	84.43	61.94	76.56	83.75	84.05	63.15	76.74	94.52	69.72	81.63	73.89
0.4	1	80.60	77.92	42.68	**65.59**	70.88	80.61	34.28	**64.40**	72.89	78.15	38.50	**64.27**	89.26	66.83	62.37	**65.27**
	10	80.87	76.00	43.10	64.49	68.63	78.11	37.21	63.80	71.80	77.44	37.66	63.52	89.89	63.75	65.29	64.29
	10^2	80.91	75.82	43.10	64.37	68.37	78.05	37.21	63.76	71.47	77.16	37.25	63.19	90.11	62.94	65.29	63.76
	10^3	80.98	75.94	42.68	64.30	68.37	78.05	37.21	63.76	71.46	77.12	37.25	63.17	90.13	62.74	65.71	63.78
	10^4	80.98	75.94	42.68	64.30	68.37	78.05	37.21	63.76	71.50	77.12	37.25	63.17	90.15	62.73	65.71	63.77
	10^5	80.96	75.94	42.68	64.30	68.37	78.05	37.21	63.76	71.63	76.95	37.66	63.20	90.15	62.72	65.71	63.77
	10^6	80.96	75.94	42.68	64.30	68.37	78.05	37.21	63.76	71.48	77.13	37.25	63.17	90.15	62.72	65.71	63.77
0.5	1	80.74	77.14	23.84	**58.49**	39.31	7.50	93.48	37.59	69.99	71.83	38.06	60.01	84.22	58.17	46.01	**53.92**
	10	82.63	74.09	28.42	58.11	42.46	14.66	87.23	40.05	71.62	72.95	36.39	60.16	85.28	55.34	47.70	52.67
	10^2	82.98	73.66	28.41	57.83	42.46	14.66	87.23	40.05	71.62	72.95	36.39	60.16	85.35	54.96	48.95	52.86
	10^3	82.97	73.67	28.41	57.83	42.46	14.66	87.23	**40.05**	71.62	72.95	36.39	**60.16**	85.32	55.02	48.95	52.89
	10^4	82.98	73.67	28.41	57.83	42.46	14.66	87.23	40.05	71.62	72.95	36.39	60.16	85.32	55.01	48.95	52.89
	10^5	82.98	73.67	28.41	57.83	42.46	14.66	87.23	40.05	71.62	72.95	36.39	60.16	85.32	55.01	48.95	52.89
	10^6	82.98	73.67	28.41	57.83	42.46	14.66	87.23	**40.05**	71.62	72.95	36.39	60.16	85.32	55.01	48.95	52.89

Table 2.22: Fuzzy rule, p =0.2, 0.3, 0.4, 0.5 and $2a$ =0, 0.01, 0.05, inner fences

FUZZY	$2a$	0.1				0.2				0.5				1			
		Train	Test			Train	Test			Train	Test			Train	Test		
p	C	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
0.2	1	96.24	83.15	91.23	**85.98**	97.04	86.19	91.65	**88.10**	97.99	96.15	92.48	**94.87**	98.28	97.77	90.80	95.33
	10	96.98	80.75	92.07	84.71	97.36	85.00	92.07	87.47	98.17	95.28	92.48	94.30	98.45	97.77	91.21	**95.47**
	10^2	97.35	77.70	92.07	82.73	97.51	84.38	91.23	86.78	98.25	94.88	91.23	93.61	98.89	97.97	87.03	94.14
	10^3	97.36	78.19	92.48	83.19	97.50	84.82	91.65	87.21	98.36	94.47	90.80	93.18	99.09	97.95	84.95	93.40
	10^4	97.49	79.19	90.40	83.12	97.53	85.34	90.40	87.11	98.37	94.68	90.80	93.32	99.12	97.73	85.36	93.40
	10^5	97.37	78.26	91.23	82.80	97.50	85.58	90.40	87.26	98.37	94.52	89.55	92.78	99.15	97.95	82.43	92.52
	10^6	97.37	78.09	91.65	82.83	97.50	85.43	91.23	87.46	98.29	94.42	90.40	93.01	99.09	98.18	84.11	93.26
0.3	1	96.15	70.98	86.21	**76.31**	97.34	75.19	87.88	**79.63**	98.49	94.38	86.63	**91.67**	98.98	98.20	82.83	**92.82**
	10	96.87	67.27	87.46	74.34	97.72	71.24	87.88	77.06	98.39	92.58	87.46	90.79	99.15	98.20	81.99	92.53
	10^2	97.39	64.01	86.21	71.78	97.72	71.37	87.88	77.15	98.42	93.39	87.45	91.31	99.59	97.97	79.93	91.66
	10^3	97.42	64.20	86.21	71.90	97.75	71.59	87.46	77.14	98.58	93.42	86.20	90.89	99.61	98.66	78.22	91.51
	10^4	97.42	64.10	86.21	71.84	97.76	71.86	87.46	77.32	98.65	93.58	85.78	90.85	99.64	98.88	76.54	91.06
	10^5	97.45	64.45	86.21	72.07	97.71	71.53	87.05	76.96	98.67	93.69	85.80	90.93	99.63	98.88	76.99	91.22
	10^6	97.43	64.22	86.21	71.91	97.75	71.76	87.05	77.11	98.68	93.60	86.61	91.16	99.66	98.66	76.97	91.07
0.4	1	94.38	60.74	71.97	**64.67**	96.69	63.86	77.83	**68.75**	98.81	84.94	79.08	**82.89**	99.22	98.66	74.87	**90.33**
	10	96.20	54.92	76.16	62.36	97.43	57.79	80.34	65.68	98.87	83.99	79.08	82.27	99.48	98.66	72.79	89.60
	10^2	96.41	53.30	74.51	60.72	97.49	58.58	78.68	65.61	98.99	85.37	78.24	82.88	99.63	98.88	69.00	88.43
	10^3	96.48	53.28	73.68	60.42	97.45	58.46	77.84	65.24	99.08	84.97	74.08	81.16	99.80	98.88	64.84	86.97
	10^4	96.48	53.42	73.26	60.36	97.46	58.37	77.01	64.89	99.09	85.23	73.66	81.18	99.87	98.88	63.59	86.53
	10^5	96.47	53.36	73.68	60.47	97.45	58.25	78.26	65.25	99.07	85.25	74.06	81.33	99.87	98.88	63.59	86.53
	10^6	96.47	53.36	73.68	60.47	97.49	58.32	77.84	65.16	99.02	84.33	74.08	80.74	99.89	98.88	63.17	86.39
0.5	1	84.22	58.17	46.01	**53.92**	96.06	51.86	65.24	**56.54**	99.05	75.91	69.82	73.78	99.43	99.11	63.15	**86.53**
	10	85.28	55.34	47.70	52.67	97.60	43.95	68.19	52.43	99.28	77.54	70.25	**74.99**	99.66	99.11	60.24	85.51
	10^2	85.35	54.96	48.95	52.86	97.79	44.76	63.15	51.19	99.38	76.77	66.92	73.32	99.85	99.36	56.05	84.20
	10^3	85.32	55.02	48.95	52.89	97.85	45.35	65.65	52.45	99.38	75.40	64.82	71.70	99.89	99.34	53.13	83.17
	10^4	85.32	55.01	48.95	52.89	97.99	46.82	62.30	52.23	99.49	76.46	64.82	72.39	100	99.11	52.72	82.88
	10^5	85.32	55.01	48.95	52.89	97.84	46.33	64.38	52.65	99.55	76.77	65.24	72.73	100	99.11	53.13	83.02
	10^6	85.32	55.01	48.95	52.89	97.81	45.95	64.40	52.40	99.53	76.64	65.24	72.65	100	99.11	53.97	83.31

Table 2.23: Fuzzy rule, p =0.2, 0.3, 0.4, 0.5 and $2a$ =0.1, 0.2, 0.5, 1

missing value than only using the mean.

Likewise, in the case of imputing by $[Q_a, Q_{1-a}]$ (see Table 2.25), we obtain better results for the accuracy when using intermediate values of $2a$ than when using $2a = 1$ (the case of imputing by the median). In particular, as happened in the previous experiment, the interquartile range seems to yield a better imputation than only considering the median.

Then, we conclude that imputing via intervals seems to be a good strategy when dealing with missing values.

2.6 Conclusions and extensions

In this work, a classification problem based on Support Vector Machines has been described, where the elements to be classified are sets with certain geometrical properties. A fuzzy and a crisp classification rule have been defined in terms of a separating hyperplane which is found via solving a margin maximization problem.

Our model generalizes the formulation given in [108, 109] for data affected by some kind of perturbations, which are supposed to be unknown but bounded for a given norm. Likewise, our problem is applied to the case of interval-valued data, where a convex quadratic problem is obtained.

Several experiments have been performed. In particular, our model gets to improve the results published in [43] for the 'car' dataset, in a multi-class problem. This tool also shows that imputation based on intervals (by using the mean and deviation of the data or by using the interquartile range) obtains good results for the case of having missing data in a classification problem.

An exhaustive study of the accuracy for the different values of the parameters C_+ and C_- (the regularization constants in SVMs), and the parameters k and a (which determine the length of the intervals), has been performed, with the aim of showing the sensitivity of the results with respect to the parameters. For an optimal choice of the meta-parameters of the modes, several strategies can be followed, such as setting C_+ and C_- equal to the range of values of the training sample (see e.g. [80]).

The introduction of different kernels in the model is another topic which deserves further studies. The extension of this model to regression is the topic to consider in the following chapter.

		Train	Test			Train	Test			Train	Test			Train	Test		
		Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
	$C\backslash k$	0				0.01				0.05				0.1			
Crisp	0.01	95.17	98.02	89.20	94.98	95.17	98.02	89.20	94.98	95.18	98.02	88.78	94.84	95.17	98.02	89.20	94.98
	0.1	96.36	97.59	93.77	96.27	96.36	97.59	93.77	96.27	96.36	97.59	93.77	96.27	96.36	97.59	93.77	96.27
	1	97.14	96.94	96.68	**96.85**	97.14	96.94	96.68	**96.85**	97.14	96.94	96.68	**96.85**	97.14	96.94	96.68	**96.85**
	10	97.25	96.72	96.68	96.71	97.23	96.94	96.68	**96.85**	97.22	96.94	96.68	**96.85**	97.22	96.72	96.68	96.71
	100	97.31	96.72	96.27	96.56	97.23	96.50	96.68	96.56	97.30	96.94	95.43	96.42	97.28	96.72	96.27	96.56
	1000	97.33	96.72	95.43	96.28	97.35	96.72	96.68	96.71	97.25	96.94	96.27	96.71	97.35	96.94	95.85	96.56
Fuzzy	0.01	95.17	98.02	89.20	94.98	95.17	98.02	89.20	94.98	95.18	98.02	88.78	94.84	95.17	98.02	89.20	94.98
	0.1	96.36	97.59	93.77	96.27	96.36	97.59	93.77	96.27	96.36	97.59	93.77	96.27	96.36	97.59	93.77	96.27
	1	97.14	96.94	96.68	**96.85**	97.14	96.94	96.68	**96.85**	97.14	96.94	96.68	**96.85**	97.14	96.94	96.68	**96.85**
	10	97.25	96.72	96.68	96.71	97.23	96.94	96.68	**96.85**	97.22	96.94	96.68	**96.85**	97.22	96.72	96.68	96.71
	100	97.31	96.72	96.27	96.56	97.23	96.50	96.68	96.56	97.30	96.94	95.43	96.42	97.28	96.72	96.27	96.56
	1000	97.33	96.72	95.43	96.28	97.35	96.72	96.68	96.71	97.25	96.94	96.27	96.71	97.35	96.94	95.85	96.56
	$C\backslash k$	0.2				0.5				0.75				1			
Crisp	0.01	95.22	98.02	88.78	94.84	95.18	98.02	89.20	94.98	95.17	98.02	88.78	94.84	95.07	98.02	89.20	94.98
	0.1	96.36	97.59	93.77	96.27	96.36	97.59	93.35	96.13	96.34	97.59	93.35	96.13	96.41	97.59	93.35	96.13
	1	97.14	96.94	97.10	**96.99**	97.15	96.94	97.10	**96.99**	97.14	96.94	97.10	**96.99**	97.14	97.15	96.68	**96.99**
	10	97.25	96.72	96.68	96.71	97.20	96.94	96.68	96.85	97.20	96.72	97.10	96.85	97.19	96.94	97.10	**96.99**
	100	97.23	96.50	96.68	96.56	97.27	96.50	96.27	96.42	97.30	96.72	95.85	96.42	97.23	96.94	95.43	96.42
	1000	97.33	96.50	95.85	96.28	97.31	96.72	95.43	96.28	97.33	96.94	95.85	96.56	97.33	96.94	95.02	96.27
Fuzzy	0.01	95.22	98.02	88.78	94.84	95.18	98.02	89.20	94.98	95.17	98.02	88.78	94.84	95.07	98.02	89.20	94.98
	0.1	96.36	97.59	93.77	96.27	96.36	97.59	93.35	96.13	96.34	97.59	93.35	96.13	96.41	97.59	93.35	96.13
	1	97.14	96.94	97.10	**96.99**	97.15	96.94	97.10	**96.99**	97.14	96.91	97.10	**96.98**	97.14	97.11	96.68	96.96
	10	97.25	96.72	96.68	96.71	97.20	96.94	96.68	96.85	97.20	96.70	97.10	96.84	97.19	96.90	97.10	**96.97**
	100	97.23	96.50	96.68	96.56	97.27	96.50	96.27	96.42	97.30	96.72	95.85	96.42	97.23	96.92	95.43	96.40
	1000	97.33	96.50	95.85	96.28	97.31	96.72	95.43	96.28	97.33	96.94	95.85	96.56	97.33	96.93	95.02	96.27

Table 2.24: Results for the database with its missing values when using $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$

		Train	Test			Train	Test			Train	Test			Train	Test		
		Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av	Av	G_+	G_-	Av
	$C\backslash 2a$	Inner fences				0				0.01				0.05			
Crisp	0.01	95.06	98.02	88.37	94.69	95.04	97.81	89.20	94.84	95.07	97.81	89.20	94.84	95.02	98.02	88.78	94.84
	0.1	96.31	97.59	92.93	95.98	96.26	97.59	93.77	96.27	96.26	97.59	93.77	96.27	96.28	97.59	93.77	96.27
	1	97.04	97.15	96.68	**96.99**	97.06	97.15	96.27	**96.85**	97.08	97.15	96.27	**96.85**	97.14	97.15	96.68	**96.99**
	10	97.11	96.50	96.68	96.56	97.00	96.72	97.10	**96.85**	96.96	96.72	96.68	96.71	97.11	96.50	97.10	96.71
	100	97.23	96.94	96.27	96.71	97.19	96.72	95.85	96.42	97.17	96.72	96.68	96.71	97.28	96.72	95.85	96.42
	1000	97.27	96.72	95.43	96.28	97.22	97.15	95.85	96.70	97.28	96.94	96.27	96.71	97.28	96.72	96.27	96.56
Fuzzy	0.01	95.06	98.06	88.37	94.72	95.04	97.81	89.20	94.84	95.07	97.81	89.20	94.84	95.02	98.02	88.78	94.84
	0.1	96.31	97.36	92.93	95.83	96.23	97.54	93.77	96.24	96.23	97.54	93.77	96.24	96.28	97.53	93.77	96.23
	1	97.04	96.78	96.68	**96.74**	97.02	97.10	96.27	**96.81**	97.04	97.09	96.27	**96.80**	97.14	97.02	96.68	**96.91**
	10	97.11	96.22	96.68	96.38	96.97	96.65	97.10	**96.81**	96.94	96.65	96.68	96.66	97.11	96.42	97.10	96.65
	100	97.23	96.68	96.27	96.53	97.17	96.68	95.85	96.39	97.16	96.68	96.68	96.68	97.28	96.60	95.85	96.34
	1000	97.27	96.43	95.43	96.08	97.21	97.15	95.85	96.70	97.27	96.88	96.27	96.67	97.28	96.60	96.27	96.48
	$C\backslash 2a$	0.1				0.2				0.5				1			
Crisp	0.01	95.06	98.02	88.78	94.84	95.06	98.02	89.20	94.98	95.12	98.02	89.20	94.98	95.17	98.02	89.20	94.98
	0.1	96.30	97.59	93.77	96.27	96.36	97.59	93.35	96.13	96.41	97.59	93.35	96.13	96.36	97.59	93.35	96.13
	1	97.14	97.15	96.68	**96.99**	97.14	97.15	96.68	**96.99**	97.17	96.94	97.10	**96.99**	97.14	96.94	96.68	**96.85**
	10	97.11	96.50	96.68	96.56	97.15	96.72	96.68	96.71	97.27	96.72	96.68	96.71	97.27	96.94	96.68	**96.85**
	100	97.27	96.72	97.10	96.85	97.30	96.72	96.27	96.56	97.30	96.50	95.85	96.28	97.30	96.72	96.68	96.71
	1000	97.31	96.94	95.43	96.42	97.33	96.72	96.68	96.71	97.38	96.72	95.43	96.28	97.36	96.72	95.43	96.28
Fuzzy	0.01	95.06	98.02	88.78	94.84	95.06	98.02	89.20	94.98	95.12	98.02	89.20	94.98	95.17	98.02	89.20	94.98
	0.1	96.30	97.53	93.77	96.23	96.36	97.53	93.35	96.09	96.41	97.59	93.35	96.13	96.36	97.59	93.35	96.13
	1	97.14	97.00	96.68	**96.89**	97.14	96.99	96.68	**96.89**	97.17	96.92	97.10	**96.98**	97.14	96.94	96.68	**96.85**
	10	97.11	96.41	96.68	96.50	97.15	96.59	96.68	96.62	97.27	96.70	96.68	96.69	97.27	96.94	96.68	**96.85**
	100	97.27	96.55	97.10	96.74	97.30	96.57	96.27	96.46	97.30	96.50	95.85	96.28	97.30	96.72	96.68	96.71
	1000	97.31	96.78	95.43	96.32	97.33	96.54	96.68	96.59	97.38	96.72	95.43	96.28	97.36	96.72	95.43	96.28

Table 2.25: Results for the database with its missing values when using $[Q_a, Q_{1-a}]$

Chapter 3

Support Vector Regression with imprecise data

Contents

3.1 Introduction

In Chapter 2, we have considered the classification problem where the elements are sets in $\mathbb{R}^d$, affected by some kind of imprecision. In this chapter, we extend that problem to the regression case. Then, we consider a regression problem with imprecise data, that is, the elements of the dataset are affected by uncertainty. We propose two formulations based on standard ϵ-Support Vector Regression, by using two different distances (maximum and Hausdorff distances) for measuring the error between predicted and real intervals. The formulation is applied to the case of interval data, where our model has been tested on real databases. The case of data affected by some kind of noise is also handled, and it will be seen that our model generalizes the formulation proposed in [107, 108]. The technique described in this chapter is also useful for modeling the case in which there exist missing values.

The structure of the chapter is the following. In Section 3.2, the extension of ϵ-Support Vector Regression to the case of non-single elements is described. A general optimization problem is given, from which two different formulations will be derived according to the distance used as a measurement of the error between the predicted interval and the observed one for each element of the dataset. The formulation with the maximum distance will be explained in depth in Section 3.3, whereas the formulation with the Hausdorff distance is given in Section 3.4. In each formulation, the general model is particularized to the case of interval data and perturbed data. In Section 3.5, a computational experiment with a cardiological database is performed. Section 3.6 includes an experiment with the methodology for imputation of missing values where the blanks are filled in by intervals built with the remaining values of the corresponding variable in the dataset. Finally, Section 3.7 contains some discussion and concluding remarks.

3.2 Modeling the problem

Whereas in the standard ϵ-SVR approach (see Subsection 1.2.1), each instance in the database is of the form $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, now we consider a database $\Omega \subset \mathbb{R}^d \times \mathbb{R}$ with elements $i = (X_i, Y_i) \in \Omega$, where Y_i is an interval in $\mathbb{R}$, $Y_i = [\tilde{l}_i, \tilde{u}_i]$, with $\tilde{l}_i \leq \tilde{u}_i$, and X_i is of the form $X_i = x_i + B_i$, with $x_i \in \mathbb{R}^d$ and with B_i being the unit ball of a symmetric gauge γ_i (as defined in (2.1)), that is, X_i is the same kind of ball as defined in the classification problem of Chapter 2 (see Subsection 2.2.1).

We also consider in this chapter the two gauges given in expressions (2.2) and (2.4), whose corresponding unit balls B_i are given in (2.3) and (2.5).

When γ_i is of the form (2.2), taking x_i such that $x_{ij} = \frac{l_{ij}+u_{ij}}{2}$, $j = 1, \ldots, d$, one has

that $X_i = x_i + B_i$ is a Cartesian product of intervals, that is, $X_i = \prod_{j=1}^{d}[l_{ij}, u_{ij}]$.

When γ_i is of the form (2.4), we can model the case of noisy data, taking x_i as the original value of the instance and with B_i being a ball representing the perturbation, unknown and bounded in p-norm by a value r_i. That is, $x \in X_i$ iff $x = x_i + s$, with $\gamma_i(s) \leq 1$, or equivalently, $\|s\|_p \leq r_i$, for each $i \in \Omega$. The regression problem with this kind of data was first tackled in [107, 108], by using ϵ-SVR as well.

In case of dealing with balls, the concept of ϵ-SVR must be modified and one has that our goal will be to compute the parameters ω and β of a hyperplane such that a given distance from (X_i, Y_i) to that hyperplane is at most ϵ, for every i in the database. Two distances will be considered: the maximum distance d_{max} and the Hausdorff distance d_H, defined on intervals $[\underline{a}, \overline{a}]$, $[\underline{b}, \overline{b}]$ as

$$\begin{aligned} d_{max}([\underline{a}, \overline{a}], [\underline{b}, \overline{b}]) &= \max\{|a-b| : \ a \in [\underline{a}, \overline{a}], \ b \in [\underline{b}, \overline{b}]\} \qquad (3.1)\\ &= \max\{|\underline{a} - \overline{b}|, |\overline{a} - \underline{b}|\} \\ d_H([\underline{a}, \overline{a}], [\underline{b}, \overline{b}]) &= \max\{\max_{a\in[\underline{a},\overline{a}]} \min_{b\in[\underline{b},\overline{b}]} |a-b|, \min_{a\in[\underline{a},\overline{a}]} \max_{b\in[\underline{b},\overline{b}]} |a-b|\} \qquad (3.2)\\ &= \max\{|\underline{a} - \underline{b}|, |\overline{a} - \overline{b}|\}. \end{aligned}$$

Then, our aim will be to seek ω and β such that

$$dist([\min_{x\in X_i}(\omega^\top x + \beta), \max_{x\in X_i}(\omega^\top x + \beta)], [\tilde{l}_i, \tilde{u}_i]) \leq \epsilon, \ \forall i \in \Omega, \qquad (3.3)$$

where $dist$ is a distance (such as $d_{\max}$ or d_H) in the space of intervals.

Given a training sample $I = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)\}$, extracted from Ω, we must formulate the optimization problem to solve to obtain the parameters ω and β for the regressor.

Different solutions to the problem can be obtained. We are interested in finding the solution with minimum norm of ω, as done in the standard Support Vector Regression case (see Subsection 1.2.1).

Then, our problem can be formulated as

$$\begin{array}{ll} \min\limits_{\omega,\beta} & \dfrac{1}{2}\displaystyle\sum_{j=1}^{d} \omega_j^2 \\ \text{s.t.} & dist([\min\limits_{x\in X_i}(\omega^\top x + \beta), \max\limits_{x\in X_i}(\omega^\top x + \beta)], [\tilde{l}_i, \tilde{u}_i]) \leq \epsilon, \ \forall i \in I. \end{array} \qquad (3.4)$$

In the next two sections, formulations for the problems with the maximum and the Hausdorff distances will be given.

3.3 Formulation based on the maximum distance

3.3.1 Formulation of the problem

Figure 3.1 gives a graphical explanation of the model for the maximum distance in the interval data case. A hyperplane is sought to fit the boxes, and penalties appear when the maximum distance from the box to the corresponding vertical projection on the hyperplane is larger than ϵ, that is, when not every point of the box is inside the ϵ-insensitive tube. Variable ξ is used for the points above the tube and ξ^* for the points below the tube.

Given the training sample $I \subseteq \Omega$, considering the maximum distance d_{max} between the predicted and the real interval, constraint (3.3) can be written as

$$\max_{(x,y)\in(X_i,Y_i)} |\omega^\top x + \beta - y| \leq \epsilon, \ \forall i \in I. \tag{3.5}$$

Constraint (3.5) can be divided into the following pair of constraints,

$$\begin{aligned} \max_{x\in X_i} \ \max_{y\in Y_i} \ (y - \omega^\top x - \beta) \leq \epsilon, \quad \forall i \in I \\ \max_{x\in X_i} \ \max_{y\in Y_i} \ (\omega^\top x + \beta - y) \leq \epsilon, \quad \forall i \in I, \end{aligned}$$

and by taking into account that Y_i is an interval $[\tilde{l}_i, \tilde{u}_i]$, we can rewrite them as

$$\begin{aligned} \max_{x\in X_i} \ (\tilde{u}_i - \omega^\top x - \beta) \leq \epsilon, \quad \forall i \in I \\ \max_{x\in X_i} \ (\omega^\top x + \beta - \tilde{l}_i) \leq \epsilon, \quad \forall i \in I. \end{aligned}$$

Then, the optimization problem (3.4) is the following

$$\begin{aligned} \min_{\omega,\beta} \quad & \frac{1}{2}\sum_{j=1}^{d} \omega_j^2 \\ \text{s.t.} \quad & \max_{x\in X_i} \ (\tilde{u}_i - \omega^\top x - \beta) \leq \epsilon, \ \forall i \in I \\ & \max_{x\in X_i} \ (\omega^\top x + \beta - \tilde{l}_i) \leq \epsilon, \ \forall i \in I. \end{aligned} \tag{3.6}$$

In order to obtain a Soft-Margin version, one can introduce some slack variables ξ, ξ^* in the constraints and one must add a penalty term in the objective function,

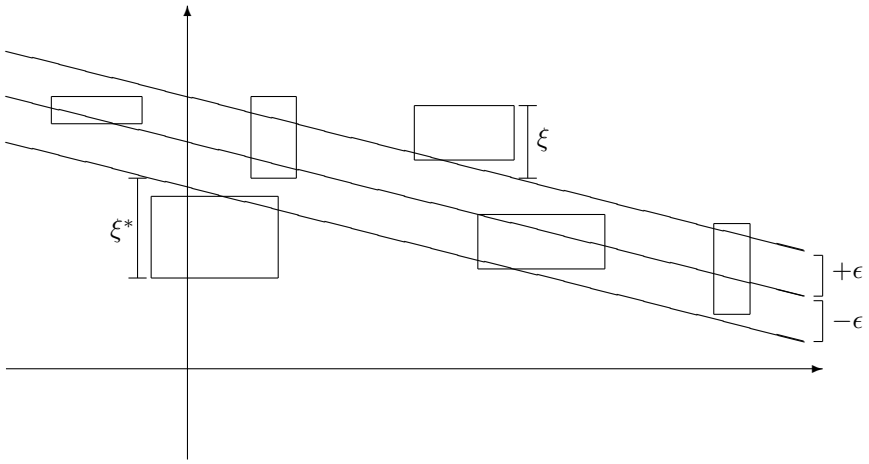

Figure 3.1: Formulation based on the maximum distance

similar to (1.30),

$$\begin{aligned}
\min_{\omega,\beta,\xi,\xi^*} \quad & \frac{1}{2}\sum_{j=1}^{d}\omega_j^2 + C\sum_{i\in I}(\xi_i+\xi_i^*) \\
\text{s.t.} \quad & \tilde{u}_i - \min_{x\in X_i}\omega^\top x - \beta \le \epsilon + \xi_i,\ \forall i\in I \qquad (3.7)\\
& \max_{x\in X_i}\omega^\top x + \beta - \tilde{l}_i \le \epsilon + \xi_i^*,\ \forall i\in I \qquad (3.8)\\
& \xi_i,\xi_i^* \ge 0,\ \forall i\in I.
\end{aligned}$$

3.3.2 An equivalent formulation

The following result gives an equivalent and more tractable formulation of our problem by using duality for the constraints (3.7)-(3.8). Recall that the dual gauge γ_i^0 of γ_i in ω is defined by $\gamma_i^0(\omega) = \max_{\gamma_i(u)\le 1}(\omega^\top u)$.

Theorem 3.1 *Problem with constraints (3.7)-(3.8) admits the following equivalent formulation as a convex quadratic minimization problem with convex nonlinear constraints*

$$\begin{aligned}
\min_{\omega,\beta,\xi,\xi^*} \quad & \frac{1}{2}\sum_{j=1}^{d}\omega_j^2 + C\sum_{i\in I}(\xi_i+\xi_i^*) \\
s.t. \quad & \tilde{u}_i - \omega^\top x_i + \gamma_i^0(\omega) - \beta \le \epsilon + \xi_i,\ \forall i\in I \\
& \omega^\top x_i + \gamma_i^0(\omega) + \beta - \tilde{l}_i \le \epsilon + \xi_i^*,\ \forall i\in I \\
& \xi_i,\xi_i^* \ge 0,\ \forall i\in I,
\end{aligned} \qquad (3.9)$$

where γ_i is the gauge associated to the element $i\in I$ defining the ball B_i used in $X_i = x_i + B_i$ and γ_i^0 is its dual gauge.

Proof.
The proof is analogous to that used in Theorem 2.1. We change constraints (3.7)-(3.8) by using that $X_i = x_i + B_i$, B_i being the unit ball induced by the gauge γ_i for each X_i.

In the proof of Theorem 2.1, by using the definition of the dual gauge γ_i^0 (which is symmetric), we obtained

$$\min_{x \in x_i + B_i} \omega^\top x = \omega^\top x_i - \gamma_i^0(\omega). \tag{3.10}$$

Analogously, one has that

$$\max_{x \in x_i + B_i} \omega^\top x = \omega^\top x_i + \max_{\gamma_i(u) \leq 1} (\omega^\top u) = \omega^\top x_i + \gamma_i^0(\omega). \tag{3.11}$$

Then, by using (3.10), the set of constraints (3.7) can be rewritten as

$$\tilde{u}_i - \omega^\top x_i + \gamma_i^0(\omega) - \beta \leq \epsilon + \xi_i, \ \forall i \in I, \tag{3.12}$$

and by using (3.11), the set of constraints (3.8) remains as follows,

$$\omega^\top x_i + \gamma_i^0(\omega) + \beta - \tilde{l}_i \leq \epsilon + \xi_i^*, \ \forall i \in I. \tag{3.13}$$

□

Below, we consider the two cases of interest for the definitions of γ_i given in (2.2) and (2.4). The first one is the case in which the elements of the database are boxes in dimension d, that is, $X_i = \prod_{j=1}^d [l_{ij}, u_{ij}]$, for every $i \in I$.

Corollary 3.1 *Let γ_i be the gauge defined in (2.2). Then, Problem (3.9) admits the following equivalent formulation as a convex quadratic problem with linear constraints*

$$\begin{array}{ll} \min\limits_{\sigma,\tau,\beta,\xi,\xi^*} & \dfrac{1}{2}\sum\limits_{j=1}^d (\sigma_j - \tau_j)^2 + C \sum\limits_{i \in I} (\xi_i + \xi_i^*) \\ s.t. & \tilde{u}_i + \sum\limits_{j=1}^d \tau_j u_{ij} - \sum\limits_{j=1}^d \sigma_j l_{ij} - \beta \leq \epsilon + \xi_i, \ \forall i \in I \\ & \sum\limits_{j=1}^d \sigma_j u_{ij} - \sum\limits_{j=1}^d \tau_j l_{ij} + \beta - \tilde{l}_i \leq \epsilon + \xi_i^*, \ \forall i \in I \\ & \xi_i, \xi_i^*, \sigma_j, \tau_j \geq 0, \ \forall i \in I, \ j = 1, \ldots, d. \end{array} \tag{3.14}$$

Proof.
For this proof, we need to observe that, for $s \in \mathbb{R}^d$, if $\gamma_i(s) = \max\limits_{j=1,\ldots,d} \dfrac{2|s_j|}{u_{ij} - l_{ij}}$ (for an element i of the training sample), then its dual gauge is

$$\gamma_i^0(s) = \sum_{j=1}^d \frac{u_{ij} - l_{ij}}{2} |s_j|. \tag{3.15}$$

Let us start with the set of constraints (3.12). If we replace $x_{ij} = \dfrac{l_{ij}+u_{ij}}{2}$, $j = 1,\ldots,d$ and $\gamma_i^0(\omega) = \displaystyle\sum_{j=1}^{d} |\omega_j| \frac{u_{ij}-l_{ij}}{2}$, one obtains the following constraints,

$$\tilde{u}_i - \sum_{j=1}^{d} \omega_j \left(\frac{l_{ij}+u_{ij}}{2}\right) + \sum_{j=1}^{d} |\omega_j| \left(\frac{u_{ij}-l_{ij}}{2}\right) - \beta \leq \epsilon + \xi_i, \ \forall i \in I.$$

Let us define $\sigma_j = \max\{0, \omega_j\}$ and $\tau_j = \max\{0, -\omega_j\}$, for $j = 1,\ldots,d$. One has that $\omega_j = \sigma_j - \tau_j$ and $|\omega_j| = \sigma_j + \tau_j$, and the constraints can be written as

$$\tilde{u}_i - \sum_{j=1}^{d} (\sigma_j - \tau_j) \left(\frac{l_{ij}+u_{ij}}{2}\right) + \sum_{j=1}^{d} (\sigma_j + \tau_j) \left(\frac{u_{ij}-l_{ij}}{2}\right) - \beta \leq \epsilon + \xi_i, \ \forall i \in I.$$

which yields

$$\tilde{u}_i + \sum_{j=1}^{d} \tau_j u_{ij} - \sum_{j=1}^{d} \sigma_j l_{ij} - \beta \leq \epsilon + \xi_i, \ \forall i \in I. \tag{3.16}$$

We proceed analogously with the set of constraints (3.13): we replace the values of x_i and $\gamma_i^0(\omega)$ and we obtain

$$\sum_{j=1}^{d} \omega_j \left(\frac{l_{ij}+u_{ij}}{2}\right) + \sum_{j=1}^{d} |\omega_j| \left(\frac{u_{ij}-l_{ij}}{2}\right) + \beta - \tilde{l}_i \leq \epsilon + \xi_i^*, \ \forall i \in I.$$

Afterwards, we introduce the variables σ_j and τ_j, and after some computations, one obtains

$$\sum_{j=1}^{d} \sigma_j u_{ij} - \sum_{j=1}^{d} \tau_j l_{ij} + \beta - \tilde{l}_i \leq \epsilon + \xi_i^*, \ \forall i \in I. \tag{3.17}$$

Joining constraints (3.16) and (3.17), we can rewrite our problem and we derive formulation (3.14). □

Remark 3.1 *When γ_i was defined in (2.2), we assumed that $l_{ij} < u_{ij}$, $\forall j = 1,\ldots,d$. In the case of degenerated boxes (that is, when $l_{ij} = u_{ij}$ for some coordinates), denote by J_F the set of indexes with $l_{ij} = u_{ij}$ and denote by J_V the set of indexes with $l_{ij} < u_{ij}$. Let us define γ_i as*

$$\gamma_i(s_1,\ldots,s_d) = \begin{cases} \max\limits_{j \in J_V} \dfrac{2|s_j|}{u_{ij}-l_{ij}}, & \text{if } s_j = 0, \ \forall j \in J_F \\ +\infty, & \text{otherwise.} \end{cases} \tag{3.18}$$

One has that $\gamma_i^0(s)$ has the same form as (3.15) and then, formulation (3.14) remains valid.

Remark 3.2 *When uncertainty only affects to the dependent variable Y_i, and then the predictor variables are single-valued, that is, $l_{ij} = u_{ij} = x_{ij}$, $\forall i \in I$, $\forall j = 1, \ldots, d$, Problem (3.14) can be rewritten as*

$$\begin{array}{ll} \min\limits_{\omega,\beta,\xi,\xi^*} & \frac{1}{2}\sum\limits_{j=1}^{d}\omega_j^2 + C\sum\limits_{i\in I}(\xi_i + \xi_i^*) \\ s.t. & \tilde{u}_i - \sum\limits_{j=1}^{d}\omega_j x_{ij} - \beta \leq \epsilon + \xi_i, \ \forall i \in I \\ & \sum\limits_{j=1}^{d}\omega_j x_{ij} + \beta - \tilde{l}_i \leq \epsilon + \xi_i^*, \ \forall i \in I \\ & \xi_i, \xi_i^* \geq 0, \ \forall i \in I, \ j = 1, \ldots, d. \end{array} \tag{3.19}$$

Likewise, if uncertainty only affects to the predictor variables and the dependent variable is single-valued, that is, $\tilde{l}_i = \tilde{u}_i = y_i$, $\forall i \in I$, the problem to solve is

$$\begin{array}{ll} \min\limits_{\sigma,\tau,\beta,\xi,\xi^*} & \frac{1}{2}\sum\limits_{j=1}^{d}(\sigma_j - \tau_j)^2 + C\sum\limits_{i\in I}(\xi_i + \xi_i^*) \\ s.t. & y_i + \sum\limits_{j=1}^{d}\tau_j u_{ij} - \sum\limits_{j=1}^{d}\sigma_j l_{ij} - \beta \leq \epsilon + \xi_i, \ \forall i \in I \\ & \sum\limits_{j=1}^{d}\sigma_j u_{ij} - \sum\limits_{j=1}^{d}\tau_j l_{ij} + \beta - y_i \leq \epsilon + \xi_i^*, \ \forall i \in I \\ & \xi_i, \xi_i^*, \sigma_j, \tau_j \geq 0, \ \forall i \in I, \ j = 1, \ldots, d. \end{array} \tag{3.20}$$

When we consider γ_i as defined in (2.4), we obtain our second case, which is of interest to model the case with data affected by some kind of perturbations. Then, by observing that the dual gauge of $\gamma_i = \frac{1}{r_i}\|\cdot\|_p$ is $\gamma_i^0 = r_i\|\cdot\|_q$ (with $\|\cdot\|_q$ the dual norm of $\|\cdot\|_p$, p and q satisfying that $\frac{1}{p} + \frac{1}{q} = 1$), we obtain the following result, previously derived by [107, 108], as a straightforward consequence of our Theorem 3.1.

Corollary 3.2 *Let γ_i be the gauge defined in (2.4). Then, Problem (3.9) admits the following equivalent formulation,*

$$\begin{array}{ll} \min\limits_{\omega,\beta,\xi,\xi^*} & \frac{1}{2}\sum\limits_{j=1}^{d}\omega_j^2 + C\sum\limits_{i\in I}(\xi_i + \xi_i^*) \\ s.t. & \tilde{u}_i - \omega^\top x_i + r_i\|\omega\|_q - \beta \leq \epsilon + \xi_i, \ \forall i \in I \\ & \omega^\top x_i + r_i\|\omega\|_q + \beta - \tilde{l}_i \leq \epsilon + \xi_i^*, \ \forall i \in I \\ & \xi_i, \xi_i^* \geq 0, \ \forall i \in I, \end{array} \tag{3.21}$$

where $\|\cdot\|_q$ is the dual norm of $\|\cdot\|_p$, i.e., $\frac{1}{p} + \frac{1}{q} = 1$.

This formulation (3.21), for $\tilde{l}_i = \tilde{u}_i = y_i$ (when the output is crisp) is equivalent to that given in [107, 108]. In these two papers, the authors formulate this problem by building the robust counterpart of the problem (by using robust optimization methods, [6, 7]) and they solve the problem in the Euclidean norm case, that is, for $p = q = 2$. Our formulation (3.9) for any kind of gauge γ_i is thus much more general than that obtained for Support Vector Regression with noisy data.

3.4 Formulation based on the Hausdorff distance

3.4.1 Formulation of the problem

Figure 3.2 explains graphically the model with the Hausdorff distance d_H. In this case, ξ and ξ^* penalize the case when the distance from $\tilde{u}_i$ to the highest value of the interval obtained projecting the box on the hyperplane is bigger than ϵ (ξ for points above the tube, ξ^* for points below the tube). Analogously, η and η^* are penalties for the distances between $\tilde{l}_i$ and the lowest value of the projection on the hyperplane.

If we use the distance d_H in (3.2) as a measurement between the predicted and the real interval-valued output, constraint (3.3) can be written as

$$\max\left\{|\tilde{u}_i - \max_{x\in X_i}(\omega^\top x+\beta)|, |\tilde{l}_i - \min_{x\in X_i}(\omega^\top x+\beta)|\right\} \leq \epsilon,\ \forall i \in I.$$

This is equivalent to say that

$$|\tilde{u}_i - \max_{x\in X_i}(\omega^\top x+\beta)| \leq \epsilon, \quad \forall i \in I \tag{3.22}$$

$$|\tilde{l}_i - \min_{x\in X_i}(\omega^\top x+\beta)| \leq \epsilon, \quad \forall i \in I. \tag{3.23}$$

Constraints (3.22)-(3.23) can be divided into

$$\begin{aligned}
\tilde{u}_i - \max_{x\in X_i}\omega^\top x - \beta \leq \epsilon, &\quad \forall i \in I\\
\max_{x\in X_i}\omega^\top x + \beta - \tilde{u}_i \leq \epsilon, &\quad \forall i \in I\\
\tilde{l}_i - \min_{x\in X_i}\omega^\top x - \beta \leq \epsilon, &\quad \forall i \in I\\
\min_{x\in X_i}\omega^\top x + \beta - \tilde{l}_i \leq \epsilon, &\quad \forall i \in I.
\end{aligned}$$

Then, when using Hausdorff distance d_H in the constraints, Problem (3.4) can be

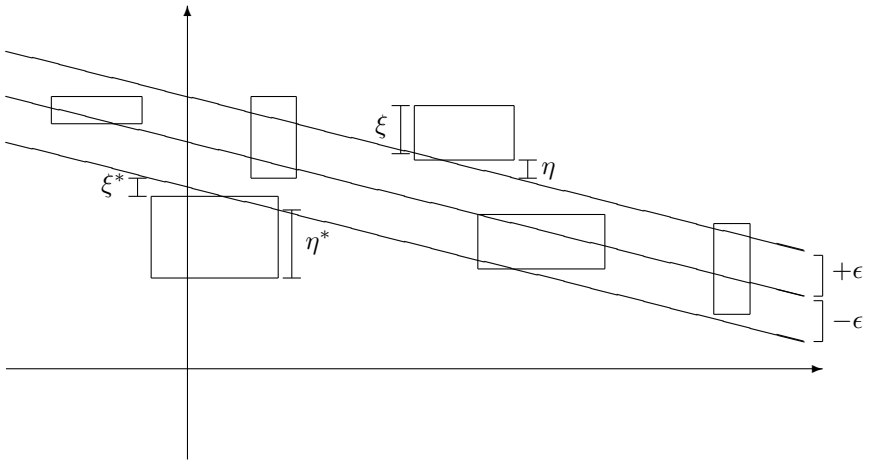

Figure 3.2: Formulation based on Hausdorff distance

written as follows,

$$\begin{aligned}
\min_{\omega,\beta} \quad & \frac{1}{2}\sum_{j=1}^{d}\omega_j^2 \\
\text{s.t.} \quad & \tilde{u}_i - \max_{x\in X_i}\omega^\top x - \beta \le \epsilon,\ \forall i \in I \\
& \max_{x\in X_i}\omega^\top x + \beta - \tilde{u}_i \le \epsilon,\ \forall i \in I \\
& \tilde{l}_i - \min_{x\in X_i}\omega^\top x - \beta \le \epsilon,\ \forall i \in I \\
& \min_{x\in X_i}\omega^\top x + \beta - \tilde{l}_i \le \epsilon,\ \forall i \in I.
\end{aligned} \tag{3.24}$$

As we did in Section 3.3, a Soft-Margin version, feasible even when (3.24) is unfeasible, is obtained here by adding slack variables ξ, ξ^*, η, η^* as follows,

$$\begin{aligned}
\min_{\omega,\beta,\xi,\xi^*,\eta,\eta^*} \quad & \frac{1}{2}\sum_{j=1}^{d}\omega_j^2 + C\sum_{i\in I}(\xi_i + \xi_i^* + \eta_i + \eta_i^*) \\
\text{s.t.} \quad & \tilde{u}_i - \max_{x\in X_i}\omega^\top x - \beta \le \epsilon + \xi_i,\ \forall i \in I && (3.25)\\
& \max_{x\in X_i}\omega^\top x + \beta - \tilde{u}_i \le \epsilon + \xi_i^*,\ \forall i \in I && (3.26)\\
& \tilde{l}_i - \min_{x\in X_i}\omega^\top x - \beta \le \epsilon + \eta_i,\ \forall i \in I && (3.27)\\
& \min_{x\in X_i}\omega^\top x + \beta - \tilde{l}_i \le \epsilon + \eta_i^*,\ \forall i \in I && (3.28)\\
& \xi_i, \xi_i^*, \eta_i, \eta_i^* \ge 0,\ \forall i \in I.
\end{aligned}$$

3.4.2 An equivalent formulation

By observing that $X_i = x_i + Bi$ (with B_i the unit ball induced by the gauge γ_i) and by using expressions (3.10)-(3.11) in constraints (3.25)-(3.28), then, following an analogous reasoning to that used in proof of Theorem 3.1, we obtain the following equivalent formulation.

Theorem 3.2 *Problem with constraints (3.25)-(3.26) admits the following equivalent formulation,*

$$\begin{array}{ll} \min\limits_{\omega,\beta,\xi,\xi^*,\eta,\eta^*} & \frac{1}{2}\sum\limits_{j=1}^{d}\omega_j^2 + C\sum\limits_{i\in I}(\xi_i+\xi_i^*+\eta_i+\eta_i^*) \\ s.t. & \tilde{u}_i - \omega^\top x_i - \gamma_i^0(\omega) - \beta \le \epsilon + \xi_i,\ \forall i\in I \\ & \omega^\top x_i + \gamma_i^0(\omega) + \beta - \tilde{u}_i \le \epsilon + \xi_i^*,\ \forall i\in I \\ & \tilde{l}_i - \omega^\top x_i + \gamma_i^0(\omega) - \beta \le \epsilon + \eta_i,\ \forall i\in I \\ & \omega^\top x_i - \gamma_i^0(\omega) + \beta - \tilde{l}_i \le \epsilon + \eta_i^*,\ \forall i\in I \\ & \xi_i,\xi_i^*,\eta_i,\eta_i^* \ge 0,\ \forall i\in I, \end{array} \tag{3.29}$$

where γ_i is the gauge associated to the element $i \in I$ defining the ball B_i used in $X_i = x_i + B_i$ and γ_i^0 is its dual gauge.

We consider now the cases for the definitions of γ_i given in (2.2) and (2.4). In the first case the elements are boxes in dimension d, that is, $X_i = \prod_{j=1}^{d}[l_{ij}, u_{ij}]$, for every $i \in I$.

Corollary 3.3 *Let γ_i be the gauge defined in (2.2). Then, Problem (3.29) admits the following equivalent formulation as a convex quadratic problem with linear and equilibrium constraints,*

$$\begin{array}{ll} \min\limits_{\sigma,\tau,\beta,\xi,\xi^*,\eta,\eta^*} & \frac{1}{2}\sum\limits_{j=1}^{d}(\sigma_j-\tau_j)^2 + C\sum\limits_{i\in I}(\xi_i+\xi_i^*+\eta_i+\eta_i^*) \\ s.t. & \tilde{u}_i - \sum\limits_{j=1}^{d}\sigma_j u_{ij} + \sum\limits_{j=1}^{d}\tau_j l_{ij} - \beta \le \epsilon + \xi_i,\ \forall i\in I \\ & \sum\limits_{j=1}^{d}\sigma_j u_{ij} - \sum\limits_{j=1}^{d}\tau_j l_{ij} + \beta - \tilde{u}_i \le \epsilon + \xi_i^*,\ \forall i\in I \\ & \tilde{l}_i - \sum\limits_{j=1}^{d}\sigma_j l_{ij} + \sum\limits_{j=1}^{d}\tau_j u_{ij} - \beta \le \epsilon + \eta_i,\ \forall i\in I \\ & \sum\limits_{j=1}^{d}\sigma_j l_{ij} - \sum\limits_{j=1}^{d}\tau_j u_{ij} + \beta - \tilde{l}_i \le \epsilon + \eta_i^*,\ \forall i\in I \\ & \sigma_j\cdot\tau_j = 0,\ j=1,\dots,d \\ & \xi_i,\xi_i^*,\eta_i,\eta_i^*,\sigma_j,\tau_j \ge 0,\ \forall i\in I,\ j=1,\dots,d. \end{array} \tag{3.30}$$

Remark 3.3 *Since we define* $\sigma_j = \max\{0, \omega_j\}$ *and* $\tau_j = \max\{0, -\omega_j\}$, *for* $j = 1, \ldots, d$, *we have imposed the following constraint*

$$\sigma_j \cdot \tau_j = 0, \ j = 1, \ldots, d.$$

In principle, equilibrium constraints should have also been added to Problem (3.14). However, they are redundant due to the convexity of the problem.

Remark 3.4 *When* γ_i *was defined in (2.2), we assumed that* $l_{ij} < u_{ij}$, $\forall j = 1, \ldots, d$. *In the case of degenerated boxes (that is, when* $l_{ij} = u_{ij}$ *for some coordinates),* γ_i *can be defined as in (3.18), and* $\gamma_i^0(s)$ *has the same form as (3.15). Then, formulation (3.30) remains valid.*

Remark 3.5 *If uncertainty only affects to* Y_i, *and* $l_{ij} = u_{ij} = x_{ij}$, $\forall i \in I$, $\forall j = 1, \ldots, d$, *the problem to solve is the following convex quadratic problem with linear constraints*

$$\begin{aligned}
\min_{\omega,\beta,\xi,\xi^*,\eta,\eta^*} \quad & \frac{1}{2}\sum_{j=1}^{d}\omega_j^2 + C\sum_{i\in I}(\xi_i + \xi_i^* + \eta_i + \eta_i^*) \\
s.t. \quad & \tilde{u}_i - \sum_{j=1}^{d}\omega_j x_{ij} - \beta \leq \epsilon + \xi_i, \ \forall i \in I \\
& \sum_{j=1}^{d}\omega_j x_{ij} + \beta - \tilde{u}_i \leq \epsilon + \xi_i^*, \ \forall i \in I \\
& \tilde{l}_i - \sum_{j=1}^{d}\omega_j x_{ij} - \beta \leq \epsilon + \eta_i, \ \forall i \in I \\
& \sum_{j=1}^{d}\omega_j x_{ij} + \beta - \tilde{l}_i \leq \epsilon + \eta_i^*, \ \forall i \in I \\
& \xi_i, \xi_i^*, \eta_i, \eta_i^* \geq 0, \ \forall i \in I, \ j = 1, \ldots, d.
\end{aligned} \tag{3.31}$$

Likewise, if uncertainty only affects to the predictor variables and $\tilde{l}_i = \tilde{u}_i = y_i$, $\forall i \in I$, *Problem (3.30) can be written as the following convex quadratic problem with*

linear and equilibrium constraints

$$
\begin{array}{ll}
\min\limits_{\sigma,\tau,\beta,\xi,\xi^*,\eta,\eta^*} & \frac{1}{2}\sum\limits_{j=1}^{d}(\sigma_j-\tau_j)^2 + C\sum\limits_{i\in I}(\xi_i+\xi_i^*+\eta_i+\eta_i^*) \\
s.t. & y_i - \sum\limits_{j=1}^{d}\sigma_j u_{ij} + \sum\limits_{j=1}^{d}\tau_j l_{ij} - \beta \le \epsilon + \xi_i,\ \forall i\in I \\
& \sum\limits_{j=1}^{d}\sigma_j u_{ij} - \sum\limits_{j=1}^{d}\tau_j l_{ij} + \beta - y_i \le \epsilon + \xi_i^*,\ \forall i\in I \\
& y_i - \sum\limits_{j=1}^{d}\sigma_j l_{ij} + \sum\limits_{j=1}^{d}\tau_j u_{ij} - \beta \le \epsilon + \eta_i,\ \forall i\in I \\
& \sum\limits_{j=1}^{d}\sigma_j l_{ij} - \sum\limits_{j=1}^{d}\tau_j u_{ij} + \beta - y_i \le \epsilon + \eta_i^*,\ \forall i\in I \\
& \sigma_j\cdot\tau_j = 0,\ j=1,\ldots,d \\
& \xi_i,\xi_i^*,\eta_i,\eta_i^*,\sigma_j,\tau_j \ge 0,\ \forall i\in I,\ j=1,\ldots,d.
\end{array}
\tag{3.32}
$$

Corollary 3.4 *Let γ_i be the gauge defined in (2.4). Then, Problem (3.29) admits the following equivalent formulation,*

$$
\begin{array}{ll}
\min\limits_{\omega,\beta,\xi,\xi^*,\eta,\eta^*} & \frac{1}{2}\sum\limits_{j=1}^{d}\omega_j^2 + C\sum\limits_{i\in I}(\xi_i+\xi_i^*+\eta_i+\eta_i^*) \\
s.t. & \tilde{u}_i - \omega^\top x_i - r_i\|\omega\|_q - \beta \le \epsilon + \xi_i,\ \forall i\in I \\
& \omega^\top x_i + r_i\|\omega\|_q + \beta - \tilde{u}_i \le \epsilon + \xi_i^*,\ \forall i\in I \\
& \tilde{l}_i - \omega^\top x_i + r_i\|\omega\|_q - \beta \le \epsilon + \eta_i,\ \forall i\in I \\
& \omega^\top x_i - r_i\|\omega\|_q + \beta - \tilde{l}_i \le \epsilon + \eta_i^*,\ \forall i\in I \\
& \xi_i,\xi_i^*,\eta_i,\eta_i^* \ge 0,\ \forall i\in I,
\end{array}
\tag{3.33}
$$

where $\|\cdot\|_q$ is the dual norm of $\|\cdot\|_p$, i.e., $\frac{1}{p}+\frac{1}{q}=1$.

3.5 Computational experiment with interval data

3.5.1 Error measures

For the numerical experiments, different measurements of the fitness of the model will be considered in each case (the standard measurements used in the literature of regression with interval data in the framework of Symbolic Data Analysis, [34, 70, 71]). They are obtained from the observed intervals $Y_i = [\tilde{l}_i, \tilde{u}_i]$ and the corresponding predicted intervals $\hat{Y}_i = [\hat{l}_i, \hat{u}_i]$, $i \in \Omega$. The measurements are the *lower bound root*

Pulse rate	Systolic blood pressure	Diastolic blood pressure
[44, 68]	[90, 100]	[50, 70]
[60, 72]	[90, 130]	[70, 90]
[56, 90]	[140, 180]	[90, 100]
[70, 112]	[110, 142]	[80, 108]
[54, 72]	[90, 100]	[50, 70]
[70, 100]	[130, 160]	[80, 110]
[72, 100]	[130, 160]	[76, 90]
[76, 98]	[110, 190]	[70, 110]
[86, 96]	[138, 180]	[90, 110]
[86, 100]	[110, 150]	[78, 100]
[63, 75]	[60, 100]	[140, 150]

Table 3.1: Cardiological interval-valued database

mean-squared error ($RMSE_l$) and the *upper bound root mean-squared error* ($RMSE_u$), which are defined as follows,

$$RMSE_l = \sqrt{\frac{1}{n}\sum_{i\in\Omega}(\tilde{l}_i - \hat{l}_i)^2} \tag{3.34}$$

$$RMSE_u = \sqrt{\frac{1}{n}\sum_{i\in\Omega}(\tilde{u}_i - \hat{u}_i)^2}, \tag{3.35}$$

with n the cardinal of Ω.

Another measurement which we introduce to compute the fitness is the *mean Hausdorff distance* ($\overline{d}_H$), between the observed and predicted intervals, for the elements of the database, defined as

$$\overline{d}_H = \frac{1}{n}\sum_{i\in\Omega} d_H([\tilde{l}_i, \tilde{u}_i], [\hat{l}_i, \hat{u}_i]) = \frac{1}{n}\sum_{i\in\Omega} \max\{|\tilde{l}_i - \hat{l}_i|, |\tilde{u}_i - \hat{u}_i|\}. \tag{3.36}$$

3.5.2 Results for resubstitution

We apply our methodology to solve the regression problem with interval data in a cardiological database. The first results for the regression analysis with this dataset were published in [9]. This dataset shows the records of the pulse rate, the systolic blood pressure and the diastolic blood pressure (these records being intervals) for eleven patients (see Table 3.1). The aim of the problem is to predict an interval for the 'pulse' variable, given the interval values of the 'systolic' and 'diastolic pressure' variables.

First of all, we compute the predicted interval for the 'pulse' variable via a resubstitution strategy (see [35]), that is, the complete set of instances will be our training sample, the regressor will be computed and we will assign the predicted interval to each patient of the training sample.

	Resubstitution		Leave-one-out	
Method\ Measure	$RMSE_l$	$RMSE_u$	$RMSE_l$	$RMSE_u$
CM	11.09	10.41	24.78	28.41
MinMax	10.43	9.71	14.82	22.25
CRM	9.81	8.94	12.81	20.37
Interval ϵ-SVR	11.03	10.31	12.71	11.89

Table 3.2: Results via resubstitution (left) and leave-one-out (right) for the cardiological interval-valued database

In Table 3.2 (left), we show the results obtained for different methods in the literature. CM stands for the center method explained in [9]. In that work, a linear regression model on the midpoint of the intervals was applied. MinMax [10] is a methodology where two models are fitted independently for the lower and the upper bounds. CRM stands for the center and range method in [34, 70], two linear independent models were used to predict the center and the range of the interval outputs and, this way, to build the predictions of the lower and upper bounds. We also present the best results obtained via our methodology (interval ϵ-SVR).

From these four methods, the best performance is obtained with CRM. Although our results are worse (for resubstitution) than those obtained with CRM, they are comparable in general with those obtained via the classical regression model. From this, one can think that a methodology based on ϵ-SVR can be competitive for this problem.

Table 3.3 shows the real interval values and the predicted outputs for the 'pulse' variable for these four methods.

For our methodology, the formulations based on the maximum distance and on the Hausdorff distance (in the interval case) were used, but the results corresponds to the best result (which was for Hausdorff-based formulation). Since the problems for the maximum distance were quadratic convex, they were solved by using AMPL+CPLEX. For the programs with the Hausdorff distance we had to use, however, AMPL+MINOS.

3.5.3 Results for leave-one-out

The next experiment shows the performance of the regressor built via our methodology when using a leave-one-out (LOO) strategy (see e.g. [53, 66]). We compute the error between the real output and the predicted output and we repeat the process for every element of the database. The fitness of the model will be studied via the measurements (3.34)-(3.36). The LOO strategy is more interesting than the resubstitution situation because it gives an idea of the behaviour of the regressor for new possible observations.

Real value	CM	MinMax	CRM	Interval ϵ-SVR
[44, 68]	[59, 66]	[56, 72]	[52, 74]	[57, 65]
[60, 72]	[63, 79]	[60, 84]	[60, 82]	[62, 80]
[56, 90]	[83, 97]	[77, 100]	[80, 100]	[83, 99]
[70, 112]	[71, 86]	[67, 89]	[67, 90]	[71, 88]
[54, 72]	[59, 66]	[56, 72]	[52, 74]	[57, 65]
[70, 100]	[78, 92]	[73, 95]	[73, 97]	[78, 95]
[72, 100]	[77, 89]	[72, 93]	[73, 93]	[77, 90]
[76, 98]	[69, 102]	[65, 104]	[73, 99]	[69, 105]
[86, 96]	[82, 99]	[77, 101]	[79, 101]	[83, 102]
[86, 100]	[71, 87]	[67, 91]	[68, 90]	[70, 89]
[63, 75]	[65, 80]	[66, 81]	[62, 82]	[66, 82]

Table 3.3: Predicted values of 'pulse' variable

The regression problem has been solved for several combinations of the parameters C and ϵ, namely, for every pair (C, ϵ), with $C = 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$, and $\epsilon = 0.0001, 0.001, 0.01, 0.1, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 5, 7, 10$. We have considered the two choices of $dist$, but the results that we present belong to d_H, because they are systematically better than those obtained with d_{max}.

Table 3.4 displays the results for the measurements $RMSE_l$ and $RMSE_u$ (expressions (3.34)-(3.35)) for the different combinations of the parameters. One can observe that the results obtained in the case of $C = 0.001$ and $C = 0.01$ are better than the rest (especially, the former). The best results for these measurements have been marked in bold in the table.

Table 3.5 shows the results obtained when we use the mean Hausdorff distance (3.36) to measure the error between the predicted interval and the real one to study the fitness of our model to the data. Good values can be found again for $C = 0.001, 0.01$ and the lowest value of the distance is in bold.

Finally, in Table 3.2 (right), a comparison for the measurements (3.34)-(3.35) obtained for the cardiological dataset via different methods is given. In particular, we present the results obtained for CM ([9]), MinMax ([10]), CRM ([34, 70]) and our methodology.

One can observe that the results obtained with our method are better than with the other models. In fact, attending to the $RMSE_l$ and $RMSE_u$ measurements, any result for $C = 0.001$ or $C = 0.01$ would be better than those obtained so far in the literature. The other methods were good in general with the training sample, but the error is bigger in the test sample, due to a problem of overfitting. The improvement with respect to CRM, which was the best result so far, is quite remarkable (especially in $RMSE_u$), and thus, one can conclude that our method is competitive to deal with regression with interval data.

C	0.00001		0.0001		0.001		0.01		0.1	
$\epsilon \backslash RMSE$	l	u	l	u	l	u	l	u	l	u
0.0001	15.17	21.66	15.24	20.33	13.08	12.77	15.66	15.46	21.48	20.33
0.001	15.17	21.66	15.35	20.35	13.08	12.77	15.66	15.46	21.48	20.33
0.01	15.15	21.60	15.34	20.35	13.08	12.76	15.66	15.46	21.48	20.33
0.1	15.15	21.59	15.31	20.32	13.04	12.76	15.66	15.46	21.49	20.38
0.5	15.08	21.43	15.12	20.04	13.04	12.61	15.86	15.66	21.51	20.56
1	15.28	21.29	15.43	19.94	13.02	12.29	15.84	15.84	21.57	20.65
1.5	15.40	21.31	15.70	19.68	13.02	12.08	15.48	15.28	21.84	20.87
2	15.63	20.98	15.87	19.53	12.89	11.90	14.51	14.15	22.04	20.87
2.5	16.00	20.88	15.96	19.63	**12.71**	**11.89**	14.25	13.86	20.68	19.67
3	16.40	20.76	15.68	19.43	13.87	12.17	14.40	13.92	19.30	18.54
3.5	16.66	20.44	15.96	19.14	12.94	12.17	14.76	14.16	19.19	18.39
5	17.10	19.40	16.55	18.14	12.79	12.61	13.59	13.21	21.79	20.70
7	17.45	17.90	16.85	16.76	13.42	13.12	14.72	13.29	21.16	20.16
10	18.38	17.24	17.74	16.61	14.28	13.11	14.25	12.65	19.67	18.57

Table 3.4: $RMSE_l$ and $RMSE_u$ for the cardiological database via leave-one-out

$\epsilon \backslash C$	0.00001	0.0001	0.001	0.01	0.1
0.0001	23.74	23.03	14.92	16.11	18.47
0.001	23.74	23.08	14.92	16.11	18.47
0.01	23.69	23.08	14.91	16.11	18.47
0.1	23.71	23.04	14.89	16.10	18.50
0.5	23.68	22.76	14.84	16.15	18.60
1	23.82	22.89	14.65	16.10	18.57
1.5	23.93	22.91	14.49	15.82	18.62
2	23.89	22.94	14.31	15.24	18.52
2.5	24.04	23.07	14.26	14.94	17.76
3	24.18	22.71	14.98	15.01	17.11
3.5	24.13	22.69	14.68	15.13	16.94
5	23.70	22.40	15.05	**14.15**	17.58
7	22.95	21.67	15.90	14.78	17.47
10	22.84	22.09	16.07	14.69	17.03

Table 3.5: Mean Hausdorff distance ($\overline{d}_H$) between the predicted interval and the real interval (leave-one-out)

3.6 Computational experiment with missing data

3.6.1 Imputation for missing values via intervals

As explained in Subsection 2.5.1, several strategies can be adopted when handling missing data, such as the imputation for given records, which means to replace the missing values of a dataset by other plausible values in such a way that the data must remain consistent.

The methodology that we propose for imputation consists in replacing each blank by an interval (instead of a single value) constructed with the non-missing values of the dataset. That is, if a blank appears in the j-th variable of an observation, the non-missing values in the j-th variable of the rest of observations are used to build the interval. Two different strategies, as done in the experiment of Section 2.5.1, will be followed to construct these intervals.

The first one is, for a missing value in the j-th variable, to build the interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$, where $\overline{x}_j$ and σ_{x_j} are, respectively, the mean and the standard deviation for the values in the j-th column of the remaining observations, and with k a parameter to be tuned.

The second strategy is based on the quantiles. We consider the interval which is defined as $[Q_a, Q_{1-a}]$, where Q_a represents the a-th quantile, and thus the interval contains all but a fraction $2a$ of all non-missing values.

3.6.2 Description of the experiment

Our formulation for regression with interval data has been applied to a real database, obtained from the UCI Machine Learning Repository [4], for dealing with missing values. The 'automobile' database contains 205 observations. Each record describes different characteristics of a determined automobile, with nominal and numerical variables. However, the nominal variables have been discarded for our experiment and thus, each observation is represented by 16 numerical variables: 15 of them will be the predictor variables and the last one, which is the price of the car, will be the dependent variable to be approximated via regression. There are several missing values, in some predictor variables (variables 2, 9, 10, 12 and 13) and in the dependent variable as well, which will be imputed via an interval.

The regression problem has been solved through 10-fold cross validation (see [66]). We have used the formulation using d_H. Before solving the corresponding optimization problem (3.30), the two different strategies explained before have been used for imputing the missing values. In the first strategy, we replace the blank by the interval

$[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$, with $\overline{x}_j$ and σ_{x_j} computed with the non-missing values in the j-th column of the elements of the training sample. The values studied for k are 0, 0.01, 0.05, 0.1, 0.2, 0.5, 0.75 and 1. Observe that the case $k = 0$ corresponds to considering the imputation to the mean.

In the second strategy, the blank is replaced by the interval $[Q_a, Q_{1-a}]$, Q_a being the a-th quantile. The values chosen for $2a$ are 0, 0.01, 0.05, 0.1, 0.2, 0.5 and 1. Observe that, when $2a = 0$, we obtain the range of the variable, when $2a = 0.5$, we obtain the interquartile range, and when $2a = 1$, the interval is reduced to a singleton, which is the median of the variable.

For each record of the database, we obtain a predicted interval $\hat{Y}_i = [\hat{l}_i, \hat{u}_i]$. Since the values of the dependent variable (the 'price' of the car) are punctual, we compute $\hat{y}_i = \frac{\hat{l}_i + \hat{u}_i}{2}$, the midpoint of the bounds of the interval, and we use it to compare the predicted and the real values for the dependent variable.

Two measurements have been chosen to compute the fitness of our model in this database: the *mean absolute error* (MAE) and the *root mean-squared error* (RMSE), defined as

$$MAE = \frac{1}{n} \sum_{i \in \Omega} |y_i - \hat{y}_i| \tag{3.37}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i \in \Omega} (y_i - \hat{y}_i)^2}, \tag{3.38}$$

between the predicted value $\hat{y}_i$ and the real value y_i of the variable 'price' in the database.

There are four records in the database with a missing value in this variable. These missing values have been transformed into an interval for the experiment to build the hyperplanes, but they are not taken into account to measure the fitness.

The imputation process and all the modifications of the database have been performed with Matlab 6.5. The optimization problems have been implemented with AMPL and solved with LOQO [112] (by using the NEOS server, [83]). Different combinations of the parameters C and ϵ have been considered.

3.6.3 Numerical results

The results for MAE for the different intervals are displayed in Tables 3.6-3.9 and for RMSE in Tables 3.10-3.13. The best results for each k and a are shown in bold and are depicted in Figures 3.3-3.4.

One can observe that the best results are obtained, in both imputation strategies, for non-degenerate intervals with medium-size intervals. When imputing by $[\overline{x}_j -$

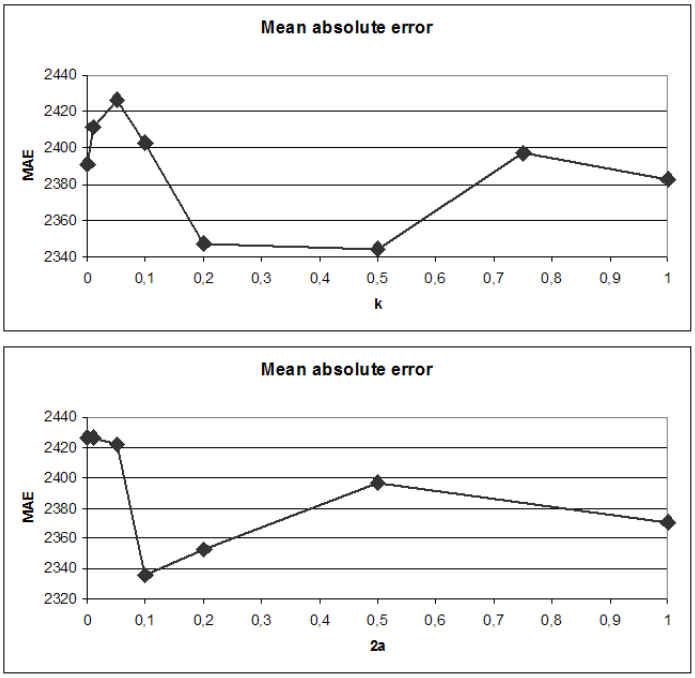

Figure 3.3: Best results for the mean absolute error. Up: interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$. Down: interval $[Q_a, Q_{1-a}]$

$k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$, the best results for the two measurements (MAE and RMSE) are obtained for $k = 0.2$ and $k = 0.5$, and we improve the results obtained when imputing to the mean (case $k = 0$). It means that, in this case, it is better to use the value of the standard deviation for imputing the missing value than only using the mean.

The situation is quite similar when imputing by $[Q_a, Q_{1-a}]$. Better results are obtained for $2a = 0.1$ and $2a = 0.2$ than for $2a = 1$ (imputation to the median).

Concerning the values of the parameters C and ϵ, one can assert that big values of C (around 1000 or 10000) seem to be more suitable for this dataset. However, the variation is bigger in the case of the parameter ϵ.

Then, we conclude that imputation via intervals seems to be a good strategy when dealing with missing values in regression problems.

k	$\epsilon\backslash C$	0.1	1	10	100	1000	10000	100000
	0.001	7084.05	2597.63	2870.03	2437.70	2508.48	2551.69	3245.35
	0.01	2619.93	2563.47	3574.43	2445.85	2487.04	2481.80	3199.14
	0.1	2960.54	2831.10	2636.85	2394.53	2523.59	2433.77	3249.50
	1	2953.87	2639.04	2483.76	2544.29	2457.99	2445.48	3388.20
	2	4223.66	2643.36	2454.52	2476.50	2676.40	2433.98	3355.55
0	5	3082.13	3682.27	2504.87	2466.14	2774.77	3684.02	3377.33
	10	2890.56	2726.73	2624.86	2636.46	2932.34	2503.47	3259.52
	50	2960.48	3092.09	3159.90	4052.34	**2390.83**	2502.70	3385.84
	100	4636.50	2992.28	2533.54	3198.47	2580.09	2424.05	3389.89
	500	2718.73	2512.29	2463.92	2697.66	2572.34	2681.71	3371.90
	1000	2765.21	2846.93	3305.12	2499.69	2988.98	2646.16	3353.96
	1500	3305.86	2929.59	2497.11	2954.70	2767.32	2951.69	3320.76
	0.001	3607.65	6784.85	6821.07	6805.25	6998.63	2487.02	4237.73
	0.01	5540.41	6808.26	6832.51	6804.68	6647.92	2598.80	4375.45
	0.1	7261.81	6807.43	6827.32	6805.62	6630.29	4863.78	4297.54
	1	7233.58	6804.80	6821.61	7400.09	6600.58	2723.24	4383.17
	2	7130.19	6791.17	6826.25	6784.95	6592.02	2519.69	4399.95
0.01	5	6561.17	6800.90	6826.71	7341.40	6548.18	2561.43	4358.44
	10	6716.50	6798.11	6807.13	6777.38	6519.67	2501.01	4386.30
	50	3564.61	2556.67	2689.38	2561.91	2481.90	2451.19	4442.67
	100	2455.32	2716.08	2641.32	2555.88	2510.63	**2411.16**	4377.38
	500	2883.64	2823.38	4171.47	2522.09	2497.71	2642.20	4471.80
	1000	3173.87	2746.49	10541.40	6774.40	2474.22	2959.28	4421.30
	1500	2634.10	2676.56	6912.14	2467.96	2445.17	2466.92	4373.35
	0.001	2640.62	2580.88	2775.28	2485.06	2472.95	2517.13	3597.68
	0.01	3079.12	2574.54	2517.89	2785.32	2493.22	2553.24	3601.02
	0.1	2726.56	4329.75	2570.95	2426.76	2604.81	3608.90	3593.12
	1	2817.53	4008.66	2828.27	4628.60	2473.74	2548.19	3633.81
	2	2772.31	2571.82	2745.44	2637.02	2995.04	2503.41	3620.94
0.05	5	3851.89	2575.74	2541.39	2457.55	2467.97	2464.02	3035.87
	10	2615.64	2499.67	2718.85	2439.87	2685.50	2859.88	2481.40
	50	2482.93	4104.34	3117.36	3068.55	3072.74	3498.52	2689.93
	100	2741.33	2922.98	2947.70	4000.05	**2426.08**	3586.14	6873.31
	500	3942.21	2693.75	2604.93	3303.11	2458.49	3563.21	4754.05
	1000	3138.34	2493.04	2481.85	2433.61	2502.43	3538.68	4437.42
	1500	3281.92	2637.69	2900.00	2464.44	2479.78	3507.60	4627.46
	0.001	2487.27	4287.86	4272.01	4140.04	2512.39	2614.55	3886.54
	0.01	2708.81	4243.74	4285.99	2609.96	2468.04	2418.56	3863.26
	0.1	4724.23	4200.64	4264.98	2518.12	2668.89	2556.64	3972.21
	1	4107.99	4148.51	4315.94	2482.56	2607.79	2470.73	4002.59
	2	4056.07	4166.55	4212.34	2465.01	2800.15	2525.05	3862.22
0.1	5	4051.51	4191.97	4061.04	2648.82	2453.27	3981.42	4070.27
	10	4054.58	4254.53	4133.81	2643.40	2480.42	4025.49	4185.52
	50	2500.49	2564.10	2580.00	3119.76	2470.50	4041.87	3990.38
	100	2544.58	2562.04	2519.60	2495.86	2513.04	3816.27	3852.54
	500	2839.62	3011.66	2458.90	2465.17	2413.05	4136.40	4111.86
	1000	3327.81	2799.12	2643.84	**2402.48**	2669.99	4321.36	4092.51
	1500	2516.18	2751.96	3136.47	2527.16	2580.17	4231.19	3902.97

Table 3.6: MAE. Interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$

k	$\epsilon \backslash C$	0.1	1	10	100	1000	10000	100000
	0.001	3266.65	2487.20	3445.20	2515.65	2865.73	2536.32	4856.38
	0.01	2494.97	2491.87	2511.66	2684.68	3242.01	2522.41	4813.76
	0.1	2717.68	2408.15	2720.75	3131.61	2660.02	2547.16	4788.83
	1	2593.12	2451.42	2410.59	2452.80	**2347.55**	2908.88	4812.79
	2	3277.46	2691.19	2400.75	2439.05	3822.34	2993.63	4813.51
0.2	5	3147.73	2642.14	2494.03	2535.89	3469.24	3369.44	4796.80
	10	2723.01	3594.18	2722.38	2825.97	2515.16	2898.94	4791.76
	50	4836.95	2974.49	3491.70	2611.63	2538.02	4914.68	4808.73
	100	3177.15	2486.29	2666.38	2493.73	2491.49	4944.50	4805.36
	500	2767.91	4651.08	2347.60	2451.43	2411.56	4909.12	4828.31
	1000	3137.61	2479.42	2664.66	2459.33	2467.53	4838.63	4598.14
	1500	2597.40	2750.55	2553.94	2725.11	2454.62	4731.55	4534.28
	0.001	2687.74	2857.61	2927.70	2443.09	2520.80	2530.71	**2344.20**
	0.01	2457.83	2576.90	2578.53	2579.90	2438.63	2546.31	3353.26
	0.1	2525.85	2704.67	2423.09	2440.30	2483.32	2614.82	3272.27
	1	2698.14	2475.53	2461.57	2515.08	3074.58	2454.19	2918.04
	2	2936.45	3152.54	2460.78	2770.73	2857.64	2810.02	3490.04
0.5	5	3045.15	2863.01	2990.63	3180.89	2377.73	2775.91	3684.54
	10	2526.67	2780.05	2924.33	2391.30	2475.90	2714.20	3593.76
	50	3027.80	2405.48	3456.79	2582.61	2483.41	3453.92	3867.74
	100	2789.75	2423.10	2528.53	2548.84	2478.88	2737.95	3921.96
	500	2515.75	2504.36	2568.48	2644.91	2527.35	2989.54	3694.38
	1000	2477.65	2954.94	3576.56	2729.56	2470.91	2521.73	3763.21
	1500	3261.32	2469.13	2771.71	2486.26	2568.64	3522.15	3658.83
	0.001	4053.53	2604.81	4455.91	2436.62	2435.64	2520.19	3475.13
	0.01	3186.41	2628.14	3047.64	2440.83	2441.36	2501.16	3658.09
	0.1	3632.12	2899.63	2596.82	2463.25	2552.02	2542.94	3231.59
	1	2825.60	3755.32	2527.77	2553.73	4088.33	2873.78	2825.54
	2	2995.87	2786.43	2584.52	2759.69	3916.15	2523.66	2532.38
0.75	5	2862.27	2538.60	2431.11	2640.30	3369.97	2441.96	2471.89
	10	2707.33	2524.26	2467.86	3150.26	**2397.07**	2694.33	3133.98
	50	3717.33	2778.60	2452.70	2505.23	2438.64	2559.73	3150.64
	100	2577.34	2679.39	5498.55	2611.74	2437.13	2963.45	2585.39
	500	3187.86	3222.31	4806.54	2486.58	2403.41	2401.88	3702.54
	1000	2765.24	2659.92	2726.10	2499.20	2461.56	2769.43	2451.55
	1500	3850.97	3079.70	2691.85	2687.26	2767.25	2540.09	3004.25
	0.001	3048.78	2968.94	3617.10	2437.22	2460.91	2537.11	3231.23
	0.01	2873.22	2597.23	2441.06	**2382.82**	2447.18	2460.87	3350.50
	0.1	3156.44	2754.81	2450.10	2403.00	2431.00	2510.49	3376.16
	1	2847.69	2445.55	2783.80	2408.15	4545.61	2536.21	3343.22
	2	2602.67	2816.98	2554.30	2547.31	4510.02	2481.35	3358.79
1	5	3725.82	2868.65	2610.87	2484.54	4556.55	2617.10	3371.79
	10	3602.39	2717.06	2462.77	2454.40	4581.36	3755.46	3381.21
	50	4891.44	5154.38	5123.45	5112.75	5151.09	2991.89	3425.14
	100	4886.65	5123.78	5109.16	5132.04	5160.88	3686.20	3278.78
	500	4849.02	5218.67	5180.17	5081.83	5092.26	3280.42	3420.74
	1000	4713.15	5030.80	5125.62	5174.21	5238.59	3446.51	3372.58
	1500	4683.77	5236.12	5122.57	5341.90	5147.82	3600.50	3347.99

Table 3.7: MAE. Interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$

$2a$	$\epsilon \backslash C$	0.1	1	10	100	1000	10000	100000
	0.001	4291.69	2737.43	2505.15	2672.89	**2426.22**	2801.65	4485.76
	0.01	4275.44	3355.66	2540.66	2498.26	2943.94	2456.37	4431.00
	0.1	4309.66	6200.28	3959.74	2481.90	2471.10	4225.71	4478.33
	1	4166.17	3400.28	2566.21	4247.22	3736.12	2521.28	4507.87
	2	3721.11	2474.54	2463.66	3120.38	4889.16	2533.68	4514.86
0	5	3781.38	5091.83	2530.84	2780.67	9073.02	3810.04	4430.49
	10	3700.08	2831.72	2569.97	2596.62	6988.85	4448.89	4467.91
	50	4275.74	3948.21	4910.17	8588.59	7703.47	4822.49	4956.24
	100	4036.68	4129.18	6783.78	7997.96	7713.47	5217.57	4521.87
	500	4033.36	4599.32	7028.35	7781.45	8128.09	4496.62	4502.25
	1000	4292.28	4463.94	7019.17	6792.90	7292.11	4436.09	4508.77
	1500	4169.41	4396.28	7842.06	7304.79	8621.70	5043.53	4402.42
	0.001	5311.40	5079.97	6540.28	6424.83	6263.68	2646.48	5071.51
	0.01	5889.63	5172.37	5673.55	6495.11	6329.34	2651.91	5087.50
	0.1	5570.80	5420.45	6062.06	6695.22	5623.35	**2426.38**	5141.43
	1	4815.58	5990.27	6534.49	6153.96	6697.46	2548.35	5135.84
	2	5204.30	5526.95	6440.05	6742.79	7087.13	2621.72	5101.77
0.01	5	5126.71	5639.25	5784.98	7383.08	6789.23	4135.82	5128.00
	10	5495.88	7119.94	6168.64	6170.35	6275.96	5344.65	5153.16
	50	4275.74	3948.21	4910.17	8588.59	7703.47	5307.97	5100.92
	100	4036.68	4129.18	6783.78	7997.96	7713.47	5196.32	5318.96
	500	4033.36	4599.32	7028.35	7781.45	8128.09	5092.20	5336.05
	1000	4292.28	4463.94	7019.17	6792.90	7292.11	5043.82	5215.09
	1500	4169.41	4396.28	7842.06	7304.79	8621.70	5046.74	5114.58
	0.001	4394.94	4283.89	4204.72	4173.29	4180.29	2822.52	7113.29
	0.01	4379.66	4242.87	4197.99	4193.68	4021.62	2442.97	6726.00
	0.1	4355.82	4242.08	4188.42	4166.00	4006.45	**2422.46**	6614.03
	1	4337.90	4238.24	4186.07	4156.91	4019.94	4061.94	6656.69
	2	4301.24	4233.77	4173.84	4172.52	4101.51	2959.08	6561.90
0.05	5	4339.47	4243.06	4177.87	4199.46	4122.69	5070.31	6561.98
	10	4324.31	4235.61	4176.33	4152.86	4246.53	10942.30	6480.23
	50	2872.30	3826.83	2982.67	2485.20	2520.76	7959.29	6650.94
	100	2740.72	2587.04	2784.63	2823.22	2485.34	7401.63	6958.09
	500	2954.76	2847.77	2764.50	2534.77	3702.37	7357.32	6901.62
	1000	2909.84	3022.14	2640.74	2457.48	5057.82	7264.25	6886.16
	1500	4304.01	3020.11	3295.84	2834.52	5720.23	7175.78	6835.20
	0.001	2730.15	2808.35	2640.26	2643.20	5949.74	2686.89	5168.97
	0.01	2602.98	2535.97	2477.90	3687.32	5985.68	2459.33	5120.17
	0.1	2526.36	2676.62	2498.03	6466.95	5956.40	3002.50	5129.39
	1	2614.56	3001.23	2432.44	6228.66	5967.36	2688.10	5475.37
	2	2861.44	2476.09	3082.08	5892.53	6069.97	2427.51	5125.53
0.1	5	3007.66	2460.56	2504.33	5973.97	5961.31	5196.28	5262.28
	10	2717.68	2484.63	2501.98	5959.15	5969.95	5302.96	5065.44
	50	4582.01	2665.71	2556.45	2697.51	**2335.63**	5435.45	3310.36
	100	5229.03	4246.36	5500.82	3367.34	2850.65	5377.14	2878.69
	500	6055.45	2910.80	10261.70	2570.64	2577.44	5409.49	2970.82
	1000	4149.11	3249.08	3504.15	3092.69	2830.00	5539.88	3327.95
	1500	4374.19	2606.48	3885.12	3922.38	2545.78	5277.26	2569.53

Table 3.8: MAE. Interval $[Q_a, Q_{1-a}]$

$2a$	$\epsilon \backslash C$	0.1	1	10	100	1000	10000	100000
	0.001	3713.64	4507.69	2739.16	2631.08	2485.23	2503.60	5096.85
	0.01	3470.88	3140.55	2490.55	2620.89	2451.64	2541.10	5466.23
	0.1	2720.40	3511.33	2582.64	2558.39	2413.21	2468.90	5120.19
	1	3297.89	2517.90	3283.58	2475.44	2542.20	2449.83	5387.51
	2	3023.56	2685.01	2787.95	2469.61	3650.91	2512.33	5283.59
0.2	5	4213.48	2598.65	3504.22	2441.16	3784.45	3690.87	5259.67
	10	7344.67	3162.90	3073.63	2477.49	2776.21	2504.30	5258.30
	50	3742.55	3620.95	2536.11	2539.38	2565.72	2881.43	5281.44
	100	3137.27	4717.39	2985.77	2410.96	2459.75	3722.22	5096.50
	500	2845.11	3410.62	3147.78	2490.84	2464.36	2536.91	5114.92
	1000	3016.49	2582.90	2479.67	3174.91	**2352.88**	4414.38	5179.80
	1500	3112.88	2580.87	2569.94	4780.88	2560.05	4336.05	5201.37
	0.001	5582.25	2686.18	2523.61	2443.64	2455.10	2551.12	4652.50
	0.01	2858.91	2704.00	2700.13	2517.27	2434.09	2526.01	4667.21
	0.1	2462.98	2566.49	3997.70	2440.53	2459.68	2499.97	4664.88
	1	2922.46	2602.72	3114.66	2458.81	2451.84	2477.96	4664.77
	2	3196.45	3023.06	2611.94	2516.18	2547.76	**2396.60**	4620.23
0.5	5	2827.81	3079.18	2635.87	2433.38	2415.74	2518.03	4690.50
	10	2482.14	2555.84	2482.86	2442.57	2817.84	4960.84	4671.36
	50	2813.10	2592.58	2446.82	2493.37	2469.03	4710.72	4690.14
	100	3124.95	2413.04	2462.09	2628.51	2455.13	4764.77	4709.35
	500	3641.64	2581.96	2455.13	2670.04	2430.39	4714.70	4706.35
	1000	2960.57	2681.66	2469.85	2412.67	2838.71	4707.30	4692.92
	1500	3278.53	7794.27	2425.08	2526.81	2544.17	4624.87	4686.43
	0.001	2936.66	2513.41	3493.27	2542.79	2583.91	2473.06	5549.70
	0.01	3498.00	2591.50	2887.49	2506.64	4155.10	2446.67	3180.51
	0.1	3206.63	2551.58	3224.95	2500.39	2547.80	3768.62	**2370.66**
	1	3124.48	2616.88	2539.67	3672.65	2526.31	2517.42	2814.52
	2	2651.65	2485.84	2517.61	3136.46	2533.92	2395.90	2539.67
1	5	2618.35	2460.57	2676.35	3332.55	2454.69	2791.44	2580.10
	10	2747.55	4577.51	2478.98	2462.48	2505.84	3470.90	2580.50
	50	2615.08	2471.55	2476.49	2511.89	2431.25	3437.53	4084.30
	100	2566.75	3063.95	2765.69	2577.83	3672.04	2776.26	3994.31
	500	2928.37	2803.62	2639.36	2507.08	4427.28	4404.41	3922.63
	1000	3021.27	2587.06	2877.91	2467.84	4482.99	4602.07	4091.80
	1500	2700.60	2590.89	2849.27	2482.05	4508.11	4594.18	3809.55

Table 3.9: MAE. Interval $[Q_a, Q_{1-a}]$

k	$\epsilon \backslash C$	0.1	1	10	100	1000	10000	100000
0	0.001	11319.10	3575.72	4071.37	3626.59	3671.97	3905.55	5612.24
	0.01	3637.03	3731.72	5344.16	3534.49	3692.20	3769.12	5528.25
	0.1	4128.08	3769.99	3822.51	3538.38	3768.07	3556.32	5576.15
	1	4114.21	3934.61	3679.58	3639.90	3546.70	3663.06	5645.37
	2	5336.41	3684.13	3633.02	3636.32	3881.87	**3312.15**	5631.39
	5	4214.51	6327.43	3636.38	3689.72	3961.85	5003.38	5656.34
	10	3952.54	3809.05	3769.64	3972.95	4126.21	3773.16	5610.92
	50	4341.77	4308.66	4296.98	5289.17	3442.81	3684.07	5685.06
	100	5864.19	4244.98	3764.20	4512.33	3561.44	3610.47	5668.92
	500	3835.73	3619.88	3481.19	3866.38	3696.39	3859.00	5647.23
	1000	3905.33	3826.13	4247.28	3669.99	3764.37	3611.11	5618.51
	1500	4632.51	3923.86	3553.37	3757.30	3755.07	3997.66	5583.04
0.01	0.001	5133.28	8218.76	8236.87	8215.47	8700.49	3735.69	7480.06
	0.01	6883.62	8225.75	8245.97	8213.51	8023.19	3806.07	7594.46
	0.1	9130.52	8224.52	8242.91	8214.26	8005.52	6725.50	7533.80
	1	9033.00	8222.46	8236.50	9147.00	7974.81	4219.05	7587.40
	2	8958.34	8207.06	8240.85	8191.92	7965.77	3744.07	7594.77
	5	8248.49	8223.48	8239.37	9117.14	7914.36	3731.60	7591.21
	10	8137.41	8216.37	8218.28	8170.04	7884.24	3851.65	7613.09
	50	6197.62	3628.40	3622.79	3770.23	3630.50	3659.00	7623.50
	100	3821.61	3677.69	3750.67	3764.96	3711.95	3510.43	7613.53
	500	4037.90	3957.69	5499.51	3677.59	3615.02	3571.24	7615.88
	1000	3991.06	4191.06	12322.40	8997.97	3568.99	4565.58	7577.69
	1500	3935.83	3694.13	8844.46	3475.48	3446.13	**3411.15**	7472.52
0.05	0.001	3789.05	3739.20	4005.49	3660.79	3613.42	3861.29	4826.76
	0.01	4208.50	3696.61	3571.41	3806.09	3752.19	3742.04	4831.36
	0.1	3920.85	5134.08	3612.19	3513.57	3656.49	4820.96	4819.56
	1	3973.86	5657.70	4046.10	8168.62	**3424.11**	3736.03	4862.35
	2	4186.57	3811.67	3800.23	3778.87	4378.85	3606.65	4838.58
	5	6708.62	3835.02	3609.89	3557.34	3514.21	3636.19	4031.60
	10	3594.31	3642.04	3779.38	3494.61	3550.80	3877.10	3510.46
	50	3760.81	5612.43	4165.71	4479.93	4941.81	4460.55	3959.37
	100	3869.34	4207.07	4013.66	6083.34	3526.17	4794.56	10046.00
	500	4884.67	3891.77	3659.79	4935.17	3515.85	4765.23	7755.52
	1000	4252.70	3548.48	3558.99	3518.94	3603.99	4730.03	7747.10
	1500	4802.01	3884.58	4042.52	3610.62	3546.17	4692.98	7909.41
0.1	0.001	3703.82	6397.87	6283.50	5953.76	3657.51	3871.00	6935.82
	0.01	3948.21	6304.60	6239.19	3962.57	3597.24	3573.01	7003.83
	0.1	7673.07	6201.15	6324.74	3648.30	3499.53	3835.95	6967.70
	1	6072.82	6215.08	6350.76	3575.25	3756.95	3686.94	6971.93
	2	5940.69	6351.52	6228.91	3556.87	4595.88	3592.04	7021.61
	5	5990.23	6408.07	6062.28	3724.65	3615.24	6974.15	7027.77
	10	6008.50	6333.60	6095.35	3721.05	3771.07	6851.05	7201.36
	50	3775.48	3710.31	3695.59	5370.07	3562.66	6857.61	7076.19
	100	3808.47	3702.25	3664.39	3680.84	3667.41	6819.46	6921.27
	500	4056.45	4293.35	3589.10	3528.88	**3439.63**	6992.88	7093.70
	1000	4455.30	4180.02	3733.45	3495.54	4074.33	7097.27	7077.80
	1500	3729.98	3785.89	4258.38	3558.07	3949.87	7039.57	6991.32

Table 3.10: RMSE. Interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$

k	$\epsilon \backslash C$	0.1	1	10	100	1000	10000	100000
	0.001	4574.58	3708.67	5225.38	3673.44	3970.93	3870.86	6993.52
	0.01	3748.62	3627.15	3550.45	3827.51	4776.32	3813.32	6912.75
	0.1	3677.97	3483.01	3714.66	4179.88	3764.01	3813.42	6806.42
	1	3811.98	3593.63	3540.67	3648.11	**3286.57**	3746.24	6947.60
	2	4414.92	3905.95	3500.10	3540.76	6296.83	4417.05	6898.57
0.2	5	4862.08	3732.55	3655.27	3620.85	5739.01	5458.80	6895.83
	10	3894.01	4470.63	3762.47	3890.03	3631.28	3733.86	6894.30
	50	6714.45	4262.35	5102.91	3636.12	3782.98	7000.88	6878.72
	100	4222.14	3518.63	3869.43	3585.53	3678.72	7072.95	6873.88
	500	4054.90	6667.62	3460.92	3610.74	3570.69	7024.06	6774.66
	1000	4148.46	3552.59	3790.67	3572.28	3609.00	6933.84	6648.84
	1500	3660.92	3918.37	3635.78	3605.39	3521.59	6804.78	6444.16
	0.001	3827.30	3756.42	4182.10	3626.63	3671.05	3896.95	**3288.79**
	0.01	3638.59	3751.82	3608.76	3632.17	3601.46	3798.62	5649.47
	0.1	3678.91	3638.62	3595.67	3619.23	3470.27	3821.41	5321.87
	1	3879.33	3641.26	3514.20	3600.65	4752.06	3617.68	3994.55
	2	4753.87	4504.39	3623.02	3749.10	3907.21	3800.86	4542.81
0.5	5	4892.93	4145.48	4070.11	4216.28	3457.10	4084.12	4701.40
	10	3682.97	4299.43	3968.99	3441.36	3305.31	3813.45	4664.51
	50	4272.42	3583.43	4787.04	3735.99	3645.94	4513.71	4897.98
	100	4546.60	3574.94	3819.47	3664.57	3664.18	3785.50	4911.87
	500	3718.34	3573.21	3770.37	3833.47	3579.05	3946.12	4757.06
	1000	3619.25	3941.42	4793.98	3947.26	3618.14	3637.59	4791.72
	1500	4807.29	3554.54	4039.48	3551.92	3582.28	4733.42	4699.28
	0.001	5692.42	3646.05	5367.72	3469.55	3601.36	3907.02	4788.49
	0.01	4401.95	3693.44	4365.82	3508.17	3582.36	3793.01	4685.99
	0.1	5847.80	4064.32	3665.05	3574.77	3611.85	3626.17	4184.07
	1	4123.32	4724.75	3575.64	3459.81	5234.26	3821.28	3824.70
	2	4426.78	3748.97	3733.31	3797.69	5367.48	3810.22	3772.94
0.75	5	4316.04	3626.03	3575.34	3641.89	4666.53	3577.78	3636.48
	10	3783.91	3594.02	3701.41	4700.05	**3398.10**	3640.88	4242.38
	50	4922.31	3983.20	3415.49	3553.63	3592.78	3488.82	4222.12
	100	3689.64	3913.53	7551.17	3654.46	3584.48	4313.92	3772.08
	500	3983.94	4661.12	6806.15	3675.50	3492.38	3494.97	6035.49
	1000	4029.09	3995.88	3737.45	3639.03	3587.36	3723.70	3658.42
	1500	5999.36	4100.38	3754.91	3690.52	3844.67	3796.85	3926.40
	0.001	4128.16	4183.07	5183.64	3604.22	3594.73	3857.57	5583.93
	0.01	4565.07	3665.57	3553.93	3478.67	3643.82	3756.18	5598.08
	0.1	4468.78	3753.63	3620.01	3543.12	**3388.10**	3600.83	5603.10
	1	4235.98	3714.19	3868.05	3517.73	7611.00	3718.95	5583.40
	2	3969.07	3937.75	3669.67	3668.33	7745.81	3664.83	5583.25
1	5	5150.88	3812.56	3626.69	3639.56	7839.28	3573.74	5582.38
	10	5377.14	3962.76	3552.08	3559.00	7905.63	5283.59	5583.96
	50	7027.88	7320.04	7338.28	7346.38	7325.76	4755.49	5623.92
	100	7022.92	7314.54	7352.96	7336.78	7345.32	5881.55	5563.69
	500	6963.32	7445.11	7373.33	7347.38	7271.79	5684.45	5605.52
	1000	6780.99	7225.78	7385.82	7270.21	7428.40	5604.50	5471.64
	1500	6765.66	7431.36	7231.79	7529.79	7351.45	5699.92	5422.52

Table 3.11: RMSE. Interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$

$2a$	$\epsilon\backslash C$	0.1	1	10	100	1000	10000	100000
	0.001	7968.61	3910.66	3562.75	3675.89	**3460.60**	4308.48	8193.99
	0.01	7938.79	4683.53	3574.54	3654.03	3979.48	3574.31	8205.21
	0.1	7955.27	8417.59	5318.17	3579.03	3561.28	5535.62	8172.67
	1	6734.64	5352.00	3542.58	5981.18	4841.99	3727.49	8189.25
	2	4852.48	3734.37	3664.87	4251.67	6572.50	3506.82	8207.09
0	5	4973.24	6335.80	3678.04	3820.11	10625.30	5510.75	8165.20
	10	5207.01	3965.71	3710.32	3713.61	8744.16	8116.04	8174.45
	50	7706.58	6846.75	8156.00	11650.40	10358.50	8354.07	8524.69
	100	7596.27	7426.39	9532.59	10707.50	10315.40	8681.55	8197.47
	500	7516.51	7657.07	9775.03	10414.20	10861.30	8099.91	8161.31
	1000	7718.19	7538.32	9791.77	9451.85	9987.97	8087.81	8025.75
	1500	7339.38	7478.43	10475.40	9907.25	11624.50	8524.43	7936.27
	0.001	8892.04	8581.60	10067.90	9243.79	9104.63	3942.26	7681.38
	0.01	9928.79	8651.02	8772.08	9317.38	9116.48	3685.38	7695.02
	0.1	9057.31	8770.58	8892.17	9485.06	7927.92	**3521.19**	7800.90
	1	8379.20	9357.95	9399.96	8909.44	9393.96	3679.78	7786.19
	2	8715.52	8769.67	9246.86	9575.23	9775.81	3741.27	7735.61
0.01	5	8673.46	8845.89	8430.49	11249.00	9380.46	5552.93	7752.27
	10	9031.78	11147.50	8864.21	8989.99	8995.75	8131.85	7759.13
	50	7706.58	6846.75	8156.00	11650.40	10358.50	8069.66	7729.19
	100	7596.27	7426.39	9532.59	10707.50	10315.40	7894.22	8117.64
	500	7516.51	7657.07	9775.03	10414.20	10861.30	7705.56	8145.81
	1000	7718.19	7538.32	9791.77	9451.85	9987.97	7579.69	7953.88
	1500	7339.38	7478.43	10475.40	9907.25	11624.50	7514.13	7820.83
	0.001	7886.85	7747.20	7695.03	7660.69	7622.62	4153.65	9860.66
	0.01	7869.29	7726.26	7689.52	7659.47	7540.20	3549.52	9370.97
	0.1	7827.95	7723.30	7680.84	7646.88	7491.78	3579.58	9308.24
	1	7805.06	7717.44	7675.62	7645.76	7474.06	5039.17	9433.61
	2	7781.42	7715.29	7669.08	7646.78	7485.76	4049.39	9329.04
0.05	5	7778.51	7717.78	7668.35	7654.05	7511.77	7453.97	9425.49
	10	7770.10	7710.68	7665.13	7640.24	7594.50	14022.10	9475.32
	50	4397.67	5021.15	4101.25	3620.44	3680.86	10855.90	9791.83
	100	3979.29	3616.86	3817.97	3748.67	**3498.77**	10093.80	10269.40
	500	4012.51	3888.82	3647.72	3727.22	5062.00	10026.20	10140.20
	1000	4210.01	3872.18	3689.93	3587.36	6670.04	9955.74	10017.80
	1500	5464.91	4054.66	4302.52	3826.73	8160.03	9855.79	9980.20
	0.001	3992.11	3830.82	3775.93	3617.94	7715.03	4042.37	8536.53
	0.01	3636.74	3657.39	3431.14	4961.07	7718.45	3614.14	8557.16
	0.1	3785.67	3847.89	3673.14	8370.87	7714.75	4111.38	8581.80
	1	3780.40	3930.47	3574.96	8022.96	7716.50	3772.44	8746.28
	2	3934.12	3671.28	4199.21	7681.82	7753.24	**3348.94**	8471.32
0.1	5	4032.47	3660.44	3618.07	7719.23	7713.52	8674.74	8609.38
	10	3627.16	3677.70	3628.80	7723.33	7716.01	8712.29	8324.23
	50	6429.00	3895.79	3729.50	3818.91	3378.95	8799.26	5145.93
	100	6009.14	5435.54	6937.80	4621.81	3992.93	8754.91	3903.91
	500	7584.26	4069.28	17300.90	3559.37	3729.92	8728.74	4158.34
	1000	5973.41	4198.71	5428.70	4827.98	3777.36	8792.98	4301.99
	1500	6663.31	3693.41	5278.47	5465.87	3612.07	8649.68	3583.70

Table 3.12: RMSE. Interval $[Q_a, Q_{1-a}]$

$2a$	$\epsilon \backslash C$	0.1	1	10	100	1000	10000	100000
	0.001	4904.20	5977.53	3779.70	3780.04	3713.63	3899.80	8984.28
	0.01	5605.16	4370.99	3640.12	3686.47	3673.52	3709.78	9073.41
	0.1	3972.21	4818.86	3545.38	3746.84	3569.74	3667.19	9035.25
	1	4991.24	3677.55	4312.87	3652.74	3479.27	3599.70	9164.58
	2	4142.54	3760.06	4126.02	3631.37	5494.16	3684.83	9110.87
0.2	5	4941.05	3740.66	5289.66	3522.49	5953.85	4675.27	9101.67
	10	10211.20	4497.64	4159.79	3691.60	4056.16	3507.80	9099.23
	50	5234.92	4479.16	3707.86	3655.00	3711.63	3744.11	9093.04
	100	4211.10	6210.85	4078.55	3566.51	3662.80	4908.79	8938.82
	500	4011.09	4884.89	4351.68	3494.58	3641.45	3498.33	8996.82
	1000	3993.23	3670.69	3627.86	4506.54	**3292.85**	7525.17	8986.34
	1500	4302.15	3674.50	3526.23	8138.59	3727.43	7432.20	8991.76
	0.001	7163.69	3970.14	3715.58	3599.01	3635.16	3925.24	8408.00
	0.01	4157.72	3770.84	3777.42	3586.69	3521.96	3605.14	8406.13
	0.1	3699.77	3707.69	5791.96	3553.77	3537.00	3439.75	8407.97
	1	4125.27	4101.13	4794.39	3638.59	3580.44	3759.09	8413.68
	2	3943.05	3963.96	3663.00	3546.00	3605.33	3507.68	8392.49
0.5	5	4007.52	4347.11	3681.40	3569.72	**3391.72**	3559.15	8405.38
	10	3708.23	3660.54	3646.33	3577.59	3979.59	8747.95	8395.11
	50	3936.52	3602.18	3642.20	3553.99	3646.75	8410.29	8419.92
	100	4592.71	3558.81	3636.78	3632.43	3583.06	8456.23	8424.25
	500	5224.26	3737.88	3635.14	3929.26	3517.29	8416.79	8420.44
	1000	4390.37	3827.70	3512.69	3441.22	3984.74	8400.83	8405.19
	1500	4736.09	13161.50	3563.73	3519.90	3610.17	8357.03	8385.13
	0.001	4561.81	3653.66	5075.52	3733.18	3796.50	3824.43	7271.65
	0.01	4604.51	3653.84	4994.71	3689.30	5342.39	3588.68	4003.83
	0.1	5056.89	3797.15	5038.06	3590.37	3705.12	4931.95	**3453.53**
	1	4258.65	3673.15	3599.40	5071.63	3650.23	3729.52	3885.88
	2	3824.71	3657.81	3635.45	4393.82	3624.19	3537.44	3643.37
1	5	3854.98	3619.83	3838.18	4231.54	3572.68	3964.56	3607.47
	10	4064.15	7036.31	3598.74	3598.12	3705.30	4540.52	3757.86
	50	3753.95	3602.27	3614.52	3620.03	3624.17	4451.16	7087.42
	100	3779.20	4301.07	3627.77	3587.51	4821.67	3912.93	6824.13
	500	4085.39	3940.26	3743.10	3634.84	5704.04	5521.75	6674.31
	1000	4261.96	3580.81	4019.48	3585.85	5771.53	5886.58	6757.16
	1500	3893.08	3707.88	4017.28	3587.48	5802.82	5898.73	6337.29

Table 3.13: RMSE. Interval $[Q_a, Q_{1-a}]$

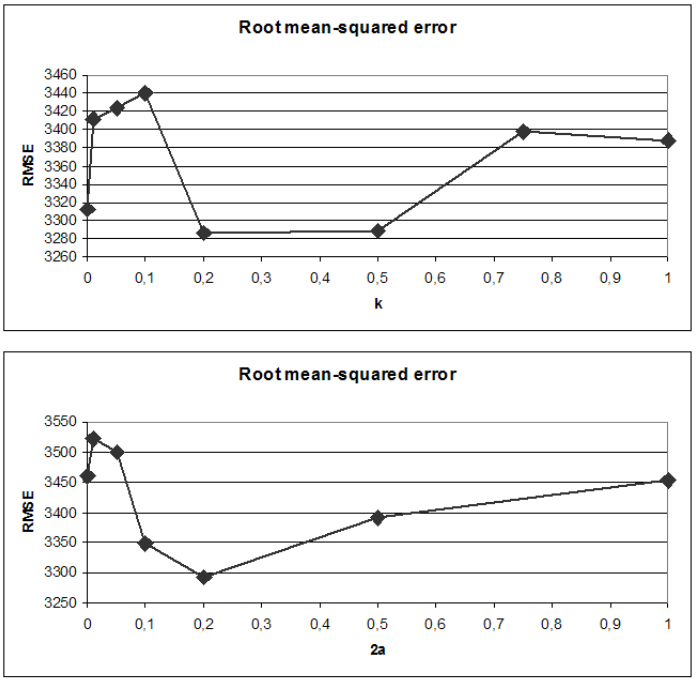

Figure 3.4: Best results for the root mean-squared error. Up: interval $[\overline{x}_j - k\sigma_{x_j}, \overline{x}_j + k\sigma_{x_j}]$. Down: interval $[Q_a, Q_{1-a}]$

3.7 Conclusions and extensions

In this work, a regression problem based on Support Vector Regression has been studied, where the elements of the database are, instead of points, sets with certain geometrical properties. Two different formulations have been proposed, depending on the distance used to measure the error between the predicted interval and the real one: the maximum distance and the Hausdorff distance.

The obtained models generalize the standard ϵ-Support Vector Regression approach to the case of having interval-valued data. In particular, the model for the maximum distance generalizes the formulation given in [107, 108] for data with some kind of noise or perturbations, which are supposed to be unknown but bounded for a given norm.

Several computational experiments with real datasets have been performed. In particular, our formulation allows to improve the results obtained in [70] for a cardiological example. Our tool has also been tested when imputation for missing data is done via intervals based on the mean and deviation of the non-missing values or on quantiles, obtaining good results for regression when using non-degenerate intervals. All these experiments display our tool as a competitive model for regression with uncertainty on the data.

An extensive study of the fitness of the model has been performed for the different values of C and ϵ, parameters of the ϵ-SVR model, and k and a, which determine the length of the intervals in the experiments with missing values. The aim is to display the sensitivity of the results with respect to the possible values of the parameters. To obtain an optimal selection of the meta-parameters C and ϵ, different strategies can be followed. For example, in [28], C is taken as a measurement depending on the mean and standard deviation of the elements of the training sample, and ϵ depends on the deviation of the noise and the size of the training sample.

As possible extensions, we propose the study of other formulations for our model. One can observe that, in formulation (3.4), the optimization problem has been posed when using the Euclidean norm for the objective function. However, the use of the l_1-norm or the l_∞-norm gives us similar expressions which are linear programs in case of considering the maximum distance for the constraints of the problem. Furthermore, other different distances (apart from maximum and Hausdorff distances) can be introduced in the constraints of Problem (3.4). Experiments with all these formulations can be done in the future to try to select the most suitable formulation for our problem.

Likewise, the introduction of kernels in the model is another topic which deserves further studies.

Chapter 4

Kernel Support Vector Regression with imprecise output

Contents

4.1 Introduction

In Chapter 3, we have studied the regression problem where imprecision affects to the predictor and the dependent variables. In this chapter, we study the particular case where only the dependent variable is imprecise. Then, we adapt the standard ϵ-Support Vector Regression methodology to the regression problem with single-valued input and interval-valued output, building two hyperplanes to give an interval output for each element of the database. The dual of the problem is also studied to allow the introduction of non-linear regressors via kernels.

The structure of the chapter is the following. In Section 4.2, we formulate the optimization problem to solve in the case of interval output. The dual formulation is obtained in Section 4.3 and we study the way of recovering an optimal solution for the primal problem, when having an optimal solution for the dual formulation. Kernels are introduced in Section 4.4 to be able to model non-linear relations in the dataset. Some computational experiments are performed with the primal and dual formulations in Section 4.5. Section 4.6 finishes the chapter with some discussion.

4.2 Modeling the problem

In this chapter, we consider a database $\Omega \subset \mathbb{R}^d \times \mathbb{R}$ with elements $i = (x_i, Y_i) \in \Omega$, where $x_i \in \mathbb{R}^d$ is the vector of single-valued predictor variables and Y_i is an interval in $\mathbb{R}$, $Y_i = [\tilde{l}_i, \tilde{u}_i]$, with $\tilde{l}_i \leq \tilde{u}_i$ (that is, the dependent variable is affected by uncertainty).

This problem can also be approached via the model proposed in Chapter 3, by using formulations (3.19) or (3.31) for the case in which the uncertainty only affects to the dependent variable Y_i, and the predictor variables are single-valued. But, in that case, given a new element, the predicted output is a singleton instead of an interval,

$$x \hookrightarrow f(x) := \omega^\top x + \beta.$$

However, if interval values are allowed as output, a better performance (in the sense of predicted intervals closer to the ones in the dataset) may be expected.

Hence, another model is proposed here, where two hyperplanes for approximating the lower and upper bounds of the dependent variable are used. Our aim will be to seek the pairs of coefficients (ω_L, β_L) and (ω_U, β_U) such that

$$dist([\omega_L^\top x_i + \beta_L, \omega_U^\top x_i + \beta_U], [\tilde{l}_i, \tilde{u}_i]) \leq \epsilon, \forall i \in \Omega, \tag{4.1}$$

for a given distance measure defined on the set of intervals.

Given a new element x, the predicted interval output will be built as follows,

$$x \hookrightarrow f(x) := [\omega_L^\top x + \beta_L, \omega_U^\top x + \beta_U]. \tag{4.2}$$

Then, given a training sample $I = \{(x_1, Y_1), (x_2, Y_2), \ldots, (x_N, Y_N)\} \subseteq \Omega$, we solve an optimization problem to obtain the optimal parameters (ω_L, β_L) and (ω_U, β_U).

When *dist* is the Hausdorff distance (3.2), we obtain that constraint (4.1) is transformed into

$$\max\left\{|\tilde{u}_i - (\omega_U^\top x + \beta_U)|, |\tilde{l}_i - (\omega_L^\top x + \beta_L)|\right\} \leq \epsilon, \ \forall i \in I,$$

and this is equivalent to say that

$$|\tilde{u}_i - (\omega_U^\top x + \beta_U)| \leq \epsilon, \quad \forall i \in I \tag{4.3}$$

$$|\tilde{l}_i - (\omega_L^\top x + \beta_L)| \leq \epsilon, \quad \forall i \in I. \tag{4.4}$$

A similar analysis can be done when *dist* is the maximum distance (3.1), although we have chosen to study the model with the Hausdorff distance because the best results for the interval-valued input and output problem in Chapter 3 were obtained with this distance.

According to rule (4.2), we must also add that

$$\omega_L^\top x_i + \beta_L \leq \omega_U^\top x_i + \beta_U, \ \forall i \in I. \tag{4.5}$$

With constraints (4.3)-(4.5), we can write the optimization problem where the sum of the squared Euclidean norms of ω_L and ω_U must be minimized (following the standard ϵ-SVR approach, see Subsection 1.2.1) as follows,

$$\begin{aligned}
\min_{\omega_L, \omega_U, \beta_L, \beta_U} \quad & \frac{1}{2}\sum_{j=1}^{d}(\omega_{Lj}^2 + \omega_{Uj}^2) \\
\text{s.t.} \quad & \tilde{u}_i - \sum_{j=1}^{d}\omega_{Uj}x_{ij} - \beta_U \leq \epsilon, \ \forall i \in I \\
& \sum_{j=1}^{d}\omega_{Uj}x_{ij} + \beta_U - \tilde{u}_i \leq \epsilon, \ \forall i \in I \\
& \tilde{l}_i - \sum_{j=1}^{d}\omega_{Lj}x_{ij} - \beta_L \leq \epsilon, \ \forall i \in I \\
& \sum_{j=1}^{d}\omega_{Lj}x_{ij} + \beta_L - \tilde{l}_i \leq \epsilon, \ \forall i \in I \\
& \sum_{j=1}^{d}\omega_{Lj}x_{ij} + \beta_L \leq \sum_{j=1}^{d}\omega_{Uj}x_{ij} + \beta_U, \ \forall i \in I.
\end{aligned} \tag{4.6}$$

By adding slack variables, a Soft-Margin version, as the following convex quadratic formulation, is obtained,

$$\begin{aligned}
\min_{\omega_L,\omega_U,\beta_L,\beta_U,\xi,\xi^*,\eta,\eta^*} \quad & \frac{1}{2}\sum_{j=1}^{d}(\omega_{Lj}^2+\omega_{Uj}^2)+C\sum_{i\in I}(\xi_i+\xi_i^*+\eta_i+\eta_i^*) \\
\text{s.t.} \quad & \tilde{u}_i-\sum_{j=1}^{d}\omega_{Uj}x_{ij}-\beta_U\leq\epsilon+\xi_i,\ \forall i\in I \\
& \sum_{j=1}^{d}\omega_{Uj}x_{ij}+\beta_U-\tilde{u}_i\leq\epsilon+\xi_i^*,\ \forall i\in I \\
& \tilde{l}_i-\sum_{j=1}^{d}\omega_{Lj}x_{ij}-\beta_L\leq\epsilon+\eta_i,\ \forall i\in I \\
& \sum_{j=1}^{d}\omega_{Lj}x_{ij}+\beta_L-\tilde{l}_i\leq\epsilon+\eta_i^*,\ \forall i\in I \\
& \sum_{j=1}^{d}\omega_{Lj}x_{ij}+\beta_L\leq\sum_{j=1}^{d}\omega_{Uj}x_{ij}+\beta_U,\ \forall i\in I \\
& \xi_i,\xi_i^*,\eta_i,\eta_i^*\geq 0,\ \forall i\in I.
\end{aligned} \tag{4.7}$$

Before building the dual problem in the following section, which will help to derive an optimal solution of Problem (4.7), we consider the trivial case in which the objective function is equal to zero.

Proposition 4.1 *Problem (4.7) has optimal solution equal to zero iff the parameter* $\epsilon>0$ *satisfies*

$$2\epsilon \geq \max_{i\in I}\{\tilde{u}_i\}-\min_{i\in I}\{\tilde{u}_i\} \tag{4.8}$$

$$2\epsilon \geq \max_{i\in I}\{\tilde{l}_i\}-\min_{i\in I}\{\tilde{l}_i\}. \tag{4.9}$$

Proof.
The objective function of Problem (4.7) is the sum of non-negative elements, thus, the value of the objective function must be greater than or equal to zero. Since our aim is to minimize that objective function, the minimum value which could be reached is zero. Then, we study under which assumptions this situation is possible.

The objective function is equal to zero iff $\omega_{Uj}=\omega_{Lj}=0$, $j=1,\ldots,d$, and all the slack variables ξ_i, ξ_i^*, η_i, η_i^* are also zero. Then, the set of constraints of Problem (4.7) remains as follows,

$$\left.\begin{array}{l}\tilde{u}_i-\beta_U\leq\epsilon,\ \ \forall i\in I\\ \beta_U-\tilde{u}_i\leq\epsilon,\ \ \forall i\in I\end{array}\right\}\ \text{i.e.,}\ \ \tilde{u}_i-\epsilon\leq\beta_U\leq\tilde{u}_i+\epsilon,\ \forall i\in I$$

$$\left.\begin{array}{l}\tilde{l}_i-\beta_L\leq\epsilon,\ \ \forall i\in I\\ \beta_L-\tilde{l}_i\leq\epsilon,\ \ \forall i\in I\end{array}\right\}\ \text{i.e.,}\ \ \tilde{l}_i-\epsilon\leq\beta_L\leq\tilde{l}_i+\epsilon,\ \forall i\in I$$

$$\beta_L\leq\beta_U$$

or equivalently,

$$\max_{i\in I}\{\tilde{u}_i\} - \epsilon \leq \beta_U \leq \min_{i\in I}\{\tilde{u}_i\} + \epsilon \tag{4.10}$$

$$\max_{i\in I}\{\tilde{l}_i\} - \epsilon \leq \beta_L \leq \min_{i\in I}\{\tilde{l}_i\} + \epsilon \tag{4.11}$$

$$\beta_L \leq \beta_U. \tag{4.12}$$

Observe that we can always choose $\beta_L \leq \beta_U$, since $\tilde{l}_i \leq \tilde{u}_i$, $\forall i \in I$. For example, one can select $\beta_L = \max_{i\in I}\{\tilde{l}_i\} - \epsilon \leq \max_{i\in I}\{\tilde{u}_i\} - \epsilon = \beta_U$.

Then, the solution is feasible iff the intervals where β_U and β_L must be contained, according to expressions (4.10)-(4.11), are non-empty, that is,

$$\begin{aligned}\max_{i\in I}\{\tilde{u}_i\} - \epsilon &\leq \min_{i\in I}\{\tilde{u}_i\} + \epsilon \\ \max_{i\in I}\{\tilde{l}_i\} - \epsilon &\leq \min_{i\in I}\{\tilde{l}_i\} + \epsilon,\end{aligned}$$

and these constraints are equivalent to expressions (4.8)-(4.9). □

Remark 4.1 *Observe that if the optimal solution of Problem (4.7) is equal to zero, all the elements of the database will have the same predicted interval output,*

$$x \hookrightarrow f(x) := [\beta_L, \beta_U].$$

4.3 Building the dual problem

In this section, the dual formulation of Problem (4.7) is obtained. We first construct the dual program, obtaining an optimal solution of (4.7) in the standard input space. Later, in Section 4.4, we extend these results to the case where the data are mapped to a higher dimensional feature space.

4.3.1 Dual formulation

Below, we build the dual formulation of Problem (4.7). The dual program can also be used to obtain an optimal solution and allows us to introduce a kernel structure in the objective function.

Theorem 4.1 *Problem (4.7) has a finite optimal solution iff the following concave quadratic maximization problem has a finite optimal solution,*

$$\begin{array}{ll}\max\limits_{\lambda,\lambda^*,\mu,\mu^*,\nu} & -\frac{1}{2}\sum\limits_{i,l\in I}\left[(\lambda_i-\lambda_i^*+\nu_i)(\lambda_l-\lambda_l^*+\nu_l)+(\mu_i-\mu_i^*-\nu_i)(\mu_l-\mu_l^*-\nu_l)\right]x_i^\top x_l \\ & -\epsilon\sum\limits_{i\in I}(\lambda_i+\lambda_i^*+\mu_i+\mu_i^*)+\sum\limits_{i\in I}(\lambda_i-\lambda_i^*)\tilde{u}_i+\sum\limits_{i\in I}(\mu_i-\mu_i^*)\tilde{l}_i \\ s.t. & \sum\limits_{i\in I}(\lambda_i-\lambda_i^*+\nu_i)=0 \\ & \sum\limits_{i\in I}(\mu_i-\mu_i^*-\nu_i)=0 \\ & 0\le\lambda_i,\lambda_i^*,\mu_i,\mu_i^*\le C,\ \forall i\in I \\ & 0\le\nu_i,\quad \forall i\in I.\end{array} \tag{4.13}$$

Proof.
Since Problem (4.7) is a linearly-constrained convex quadratic program, one has that it admits a finite optimal solution iff its dual has a finite optimal solution and, in that case, the two values coincide (see Section 6.6 in [5]). Then, the only thing to prove is that formulation (4.13) is the dual program of Problem (4.7).

To build the dual of Problem (4.7), firstly, we introduce nonnegative Lagrange multipliers for every constraint and we compute the Lagrangean function L in the primal and dual variables,

$$\begin{aligned}L \;=\; & \frac{1}{2}\sum_{j=1}^{d}(\omega_{Lj}^2+\omega_{Uj}^2)+C\sum_{i\in I}(\xi_i+\xi_i^*+\eta_i+\eta_i^*) \\ & +\sum_{i\in I}\lambda_i(\tilde{u}_i-\sum_{j=1}^{d}\omega_{Uj}x_{ij}-\beta_U-\epsilon-\xi_i)+\sum_{i\in I}\lambda_i^*(\sum_{j=1}^{d}\omega_{Uj}x_{ij}+\beta_U-\tilde{u}_i-\epsilon-\xi_i^*) \\ & +\sum_{i\in I}\mu_i(\tilde{l}_i-\sum_{j=1}^{d}\omega_{Lj}x_{ij}-\beta_L-\epsilon-\eta_i)+\sum_{i\in I}\mu_i^*(\sum_{j=1}^{d}\omega_{Lj}x_{ij}+\beta_L-\tilde{l}_i-\epsilon-\eta_i^*) \\ & +\sum_{i\in I}\nu_i(\sum_{j=1}^{d}\omega_{Lj}x_{ij}+\beta_L-\sum_{j=1}^{d}\omega_{Uj}x_{ij}-\beta_U)-\sum_{i\in I}(\gamma_i\xi_i+\gamma_i^*\xi_i^*+\delta_i\eta_i+\delta_i^*\eta_i^*).\end{aligned}$$

The partial derivatives of the Lagrangean function are used to build the dual problem. We obtain the following constraints,

$$\frac{\partial L}{\partial \omega_{Uj}} = \omega_{Uj} - \sum_{i \in I} \lambda_i x_{ij} + \sum_{i \in I} \lambda_i^* x_{ij} - \sum_{i \in I} \nu_i x_{ij} = 0, \ j = 1, \ldots, d \quad (4.14)$$

$$\frac{\partial L}{\partial \omega_{Lj}} = \omega_{Lj} - \sum_{i \in I} \mu_i x_{ij} + \sum_{i \in I} \mu_i^* x_{ij} + \sum_{i \in I} \nu_i x_{ij} = 0, \ j = 1, \ldots, d \quad (4.15)$$

$$\frac{\partial L}{\partial \beta_U} = -\sum_{i \in I} \lambda_i + \sum_{i \in I} \lambda_i^* - \sum_{i \in I} \nu_i = 0 \quad (4.16)$$

$$\frac{\partial L}{\partial \beta_L} = -\sum_{i \in I} \mu_i + \sum_{i \in I} \mu_i^* + \sum_{i \in I} \nu_i = 0 \quad (4.17)$$

$$\frac{\partial L}{\partial \xi_i} = C - \lambda_i - \gamma_i = 0, \ \forall i \in I \quad (4.18)$$

$$\frac{\partial L}{\partial \xi_i^*} = C - \lambda_i^* - \gamma_i^* = 0, \ \forall i \in I \quad (4.19)$$

$$\frac{\partial L}{\partial \eta_i} = C - \mu_i - \delta_i = 0, \ \forall i \in I \quad (4.20)$$

$$\frac{\partial L}{\partial \eta_i^*} = C - \mu_i^* - \delta_i^* = 0, \ \forall i \in I. \quad (4.21)$$

From constraints (4.14)-(4.15), we derive an expression for computing ω_U, ω_L as a function of the Lagrangean multipliers,

$$\omega_U = \sum_{i \in I} (\lambda_i - \lambda_i^* + \nu_i) x_i \quad (4.22)$$

$$\omega_L = \sum_{i \in I} (\mu_i - \mu_i^* - \nu_i) x_i. \quad (4.23)$$

From constraints (4.16)-(4.17), we obtain the following constraints for the dual problem,

$$\sum_{i \in I} (\lambda_i - \lambda_i^* + \nu_i) = 0 \quad (4.24)$$

$$\sum_{i \in I} (\mu_i - \mu_i^* - \nu_i) = 0. \quad (4.25)$$

Moreover, from constraints (4.18)-(4.21), the nonnegative multipliers λ_i, λ_i^*, μ_i and μ_i^* are bounded by the parameter C, that is,

$$0 \leq \lambda_i, \lambda_i^*, \mu_i, \mu_i^* \leq C, \ \forall i \in I. \quad (4.26)$$

By replacing the values of ω_L and ω_U, obtained in (4.22)-(4.23), in the Lagrangean function L and by using expressions (4.24)-(4.25) and the sets of constraints (4.18)-

(4.21), the objective function for the dual problem can be rewritten as follows,

$$\begin{aligned}
\tilde{L}(\lambda, \lambda^*, \mu, \mu^*, \nu) &= -\frac{1}{2}\sum_{j=1}^{d}\left(\sum_{i\in I}(\lambda_i - \lambda_i^* + \nu_i)x_{ij}\right)\left(\sum_{l\in I}(\lambda_l - \lambda_l^* + \nu_l)x_{lj}\right) \\
&\quad -\frac{1}{2}\sum_{j=1}^{d}\left(\sum_{i\in I}(\mu_i - \mu_i^* - \nu_i)x_{ij}\right)\left(\sum_{l\in I}(\mu_l - \mu_l^* - \nu_l)x_{lj}\right) \\
&\quad -\epsilon\sum_{i\in I}(\lambda_i + \lambda_i^* + \mu_i + \mu_i^*) + \sum_{i\in I}(\lambda_i - \lambda_i^*)\tilde{u}_i + \sum_{i\in I}(\mu_i - \mu_i^*)\tilde{l}_i \\
&= -\frac{1}{2}\sum_{i,l\in I}\left[(\lambda_i - \lambda_i^* + \nu_i)(\lambda_l - \lambda_l^* + \nu_l) + (\mu_i - \mu_i^* - \nu_i)(\mu_l - \mu_l^* - \nu_l)\right] x_i^\top x_l \\
&\quad -\epsilon\sum_{i\in I}(\lambda_i + \lambda_i^* + \mu_i + \mu_i^*) + \sum_{i\in I}(\lambda_i - \lambda_i^*)\tilde{u}_i + \sum_{i\in I}(\mu_i - \mu_i^*)\tilde{l}_i. \qquad (4.27)
\end{aligned}$$

Then, by maximizing the objective function (4.27) and by adding constraints (4.24)-(4.26), along with the non-negativity of the variables ν_i, we obtain the dual formulation (4.13), which is a concave quadratic maximization problem in the variables λ, λ^*, μ, μ^* and ν.

□

4.3.2 Reconstruction of an optimal solution for the primal problem

In the following, we show how to construct an optimal solution of the primal problem (4.7), given an optimal solution of the dual problem (4.13).

From expressions (4.22)-(4.23), we obtain the values for ω_U and ω_L. For the remaining variables (β_L, β_U, and the slack variables), we use that the following Karush-

Kuhn-Tucker conditions must be satisfied (see Section 4.3 in [5]),

$$\lambda_i \cdot (\tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} - \beta_U - \epsilon - \xi_i) = 0, \quad \forall i \in I \tag{4.28}$$

$$\lambda_i^* \cdot (\sum_{j=1}^{d} \omega_{Uj} x_{ij} + \beta_U - \tilde{u}_i - \epsilon - \xi_i^*) = 0, \quad \forall i \in I \tag{4.29}$$

$$\mu_i \cdot (\tilde{l}_i - \sum_{j=1}^{d} \omega_{Lj} x_{ij} - \beta_L - \epsilon - \eta_i) = 0, \quad \forall i \in I \tag{4.30}$$

$$\mu_i^* \cdot (\sum_{j=1}^{d} \omega_{Lj} x_{ij} + \beta_L - \tilde{l}_i - \epsilon - \eta_i^*) = 0, \quad \forall i \in I \tag{4.31}$$

$$\nu_i \cdot (\sum_{j=1}^{d} \omega_{Lj} x_{ij} + \beta_L - \sum_{j=1}^{d} \omega_{Uj} x_{ij} - \beta_U) = 0, \quad \forall i \in I \tag{4.32}$$

$$\xi_i \cdot (C - \lambda_i) = 0, \quad \forall i \in I \tag{4.33}$$

$$\xi_i^* \cdot (C - \lambda_i^*) = 0, \quad \forall i \in I \tag{4.34}$$

$$\eta_i \cdot (C - \mu_i) = 0, \quad \forall i \in I \tag{4.35}$$

$$\eta_i^* \cdot (C - \mu_i^*) = 0, \quad \forall i \in I \tag{4.36}$$

$$0 \leq \lambda_i, \lambda_i^*, \mu_i, \mu_i^* \leq C, \quad \forall i \in I. \tag{4.37}$$

Lemma 4.1 *Given $(\lambda, \lambda^*, \mu, \mu^*, \nu)$ a solution of the Karush-Kuhn-Tucker system (4.28)-(4.37), one has that*

$$\lambda_i \cdot \lambda_i^* = 0, \quad \forall i \in I \tag{4.38}$$

$$\mu_i \cdot \mu_i^* = 0, \quad \forall i \in I. \tag{4.39}$$

Proof.
Suppose that there exists $i_0 \in I$ such that $\lambda_{i_0}, \lambda_{i_0}^* > 0$. According to (4.28)-(4.29), the two corresponding constraints would become active and, for any $\epsilon > 0$, the following equalities would be satisfied simultaneously,

$$\begin{aligned} \tilde{u}_{i_0} - \sum_{j=1}^{d} \omega_{Uj} x_{i_0 j} - \beta_U &= \epsilon + \xi_{i_0} > 0 \\ \tilde{u}_{i_0} - \sum_{j=1}^{d} \omega_{Uj} x_{i_0 j} - \beta_U &= -\epsilon - \xi_{i_0}^* < 0, \end{aligned}$$

which is a contradiction. Hence, $\lambda_i \cdot \lambda_i^* = 0$, $\forall i \in I$.

With a similar reasoning, one can show that $\mu_i \cdot \mu_i^* = 0$, $\forall i \in I$. □

Remark 4.2 *The relative position of points with respect to the ϵ-insensitive tube around $H_U : \omega_U^\top x + \beta_U$ is obtained from the values of the dual variables λ, λ^*. Indeed, let $(\lambda, \lambda^*, \mu, \mu^*, \nu)$ be a solution of the Karush-Kuhn-Tucker system (4.28)-(4.37) and let $(\omega_L, \omega_U, \beta_L, \beta_U, \xi, \xi^*, \eta, \eta^*)$ be optimal for (4.7). By (4.38), there are five possible relative positions of points w.r.t. the ϵ-insensitive tube around the hyperplane H_U (see Figure 4.1), which are studied separately.*

1. *Above the tube: if $\lambda_i = C$, by (4.28) and since $\xi_i \geq 0$, one has that*

$$\tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} - \beta_U = \epsilon + \xi_i \geq \epsilon. \tag{4.40}$$

2. *On the upper boundary of the tube: if $0 < \lambda_i < C$ then, by (4.33), $\xi_i = 0$, and by (4.28),*

$$\tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} - \beta_U = \epsilon. \tag{4.41}$$

3. *Inside the tube: if $\lambda_i = \lambda_i^* = 0$, by (4.33)-(4.34), the slack variables are also zero, $\xi_i = \xi_i^* = 0$, and, since $(\omega_L, \omega_U, \beta_L, \beta_U, \xi, \xi^*, \eta, \eta^*)$ is feasible for (4.7),*

$$\tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} - \beta_U \leq \epsilon \tag{4.42}$$

$$\sum_{j=1}^{d} \omega_{Uj} x_{ij} + \beta_U - \tilde{u}_i \leq \epsilon. \tag{4.43}$$

4. *On the lower boundary of the tube: if $0 < \lambda_i^* < C$ then, by (4.34), $\xi_i^* = 0$, and by (4.29),*

$$\sum_{j=1}^{d} \omega_{Uj} x_{ij} + \beta_U - \tilde{u}_i = \epsilon. \tag{4.44}$$

5. *Below the tube: if $\lambda_i^* = C$, by (4.29) and since $\xi_i^* \geq 0$,*

$$\sum_{j=1}^{d} \omega_{Uj} x_{ij} + \beta_U - \tilde{u}_i = \epsilon + \xi_i^* \geq \epsilon. \tag{4.45}$$

In the following remark, the same reasoning is repeated for the hyperplane $H_L : \omega_L^\top x + \beta_L$.

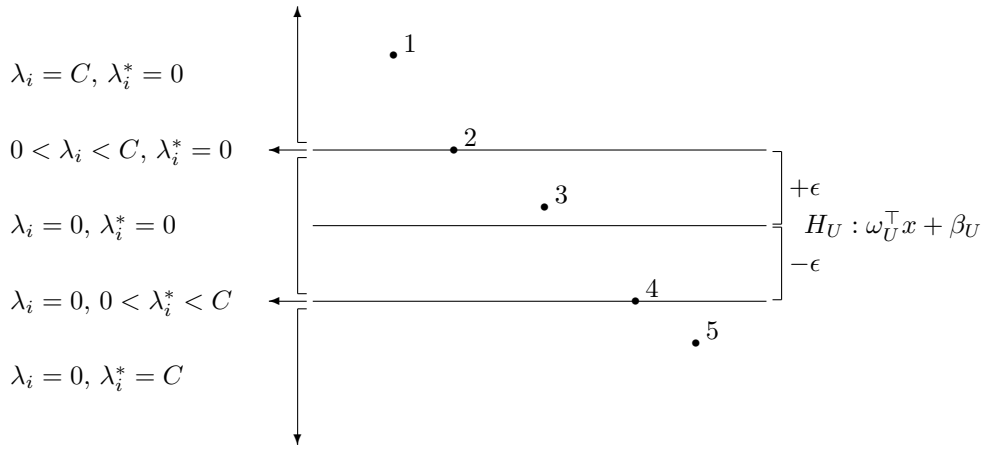

Figure 4.1: Geometrical idea of the position of $\tilde{u}_i$ according to the value of λ_i and λ_i^*

Remark 4.3 *Similarly, μ, μ^* determine the relative position of points w.r.t. the ϵ-insensititve tube. Let $(\lambda, \lambda^*, \mu, \mu^*, \nu)$ be a solution of the Karush-Kuhn-Tucker system (4.28)-(4.37) and let $(\omega_L, \omega_U, \beta_L, \beta_U, \xi, \xi^*, \eta, \eta^*)$ be optimal for (4.7). As done for $\tilde{u}_i$ in Remark 4.2, the position of the lower bounds $\tilde{l}_i$ of the elements of the training sample I with respect to the hyperplane $H_L : \omega_L^\top x + \beta_L$ can be derived from the values of the dual variables μ, μ^*.*

By (4.39), one has that μ_i or μ_i^* must be zero. Then, there are five types of points concerning their situation with respect to the ϵ-insensitive tube around H_L (see Figure 4.2):

1. *Above the tube: if $\mu_i = C$, by (4.30) and since $\eta_i \geq 0$,*

$$\tilde{l}_i - \sum_{j=1}^{d} \omega_{Lj} x_{ij} - \beta_L = \epsilon + \eta_i \geq \epsilon. \tag{4.46}$$

2. *On the upper boundary of the tube: if $0 < \mu_i < C$ then, by (4.35), $\eta_i = 0$, and by (4.30),*

$$\tilde{l}_i - \sum_{j=1}^{d} \omega_{Lj} x_{ij} - \beta_L = \epsilon. \tag{4.47}$$

3. *Inside the tube: if $\mu_i = \mu_i^* = 0$, by (4.35)-(4.36), the slack variables are also zero, $\eta_i = \eta_i^* = 0$, and then*

$$\tilde{l}_i - \sum_{j=1}^{d} \omega_{Lj} x_{ij} - \beta_L \leq \epsilon \tag{4.48}$$

$$\sum_{j=1}^{d} \omega_{Lj} x_{ij} + \beta_L - \tilde{l}_i \leq \epsilon. \tag{4.49}$$

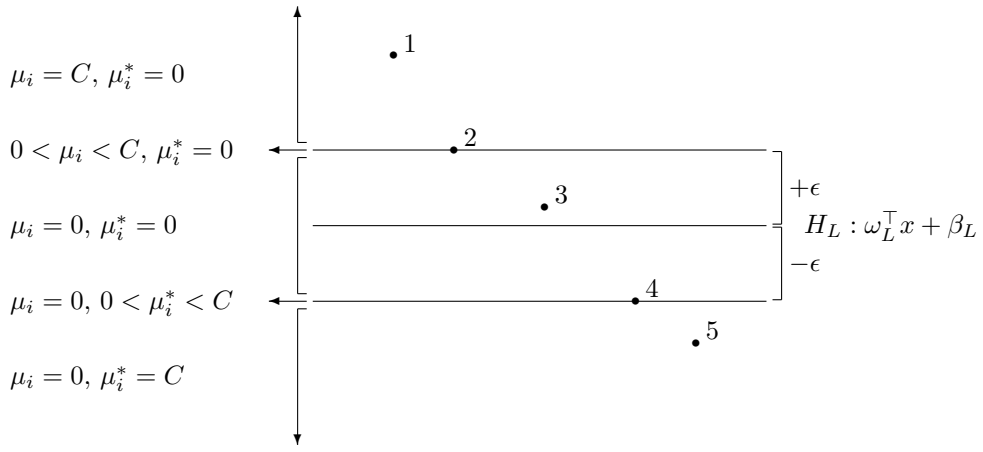

Figure 4.2: Geometrical idea of the position of $\tilde{l}_i$ according to the value of μ_i and μ_i^*

4. *On the lower boundary of the tube: if* $0 < \mu_i^* < C$ *then, by (4.36),* $\eta_i^* = 0$*, and by (4.31),*

$$\sum_{j=1}^{d} \omega_{Lj} x_{ij} + \beta_L - \tilde{l}_i = \epsilon. \tag{4.50}$$

5. *Below the tube: if* $\mu_i^* = C$*, by (4.31) and since* $\eta_i^* \geq 0$*,*

$$\sum_{j=1}^{d} \omega_{Lj} x_{ij} + \beta_L - \tilde{l}_i = \epsilon + \mu_i^* \geq \epsilon. \tag{4.51}$$

Theorem 4.2 *Let* $(\lambda, \lambda^*, \mu, \mu^*, \nu)$ *be a solution of the Karush-Kuhn-Tucker system (4.28)-(4.37) and let* $(\omega_L, \omega_U, \beta_L, \beta_U, \xi, \xi^*, \eta, \eta^*)$ *be optimal for (4.7). Then:*

1. *If there exists* $i_0 \in I$ *such that* $0 < \lambda_{i_0} < C$*, then there exists a unique optimal value of* β_U *for Problem (4.7),*

$$\beta_U = \tilde{u}_{i_0} - \sum_{j=1}^{d} \omega_{Uj} x_{i_0 j} - \epsilon. \tag{4.52}$$

Analogously, if there exists $i_1 \in I$ *such that* $0 < \lambda_{i_1}^* < C$*, one has a unique optimal value of* β_U *for Problem (4.7),*

$$\beta_U = \tilde{u}_{i_1} - \sum_{j=1}^{d} \omega_{Uj} x_{i_1 j} + \epsilon. \tag{4.53}$$

Otherwise, if $\lambda_i, \lambda_i^* \in \{0, C\}$, $\forall i \in I$, *the set of solutions for* β_U *is the following interval*

$$\beta_U \in \big[\max\{\max_{\{i:\ \lambda_i+\lambda_i^*=0\}}(\tilde{u}_i - \sum_{j=1}^{d}\omega_{Uj}x_{ij}) - \epsilon, \max_{\{i:\ \lambda_i^*=C\}}(\tilde{u}_i - \sum_{j=1}^{d}\omega_{Uj}x_{ij}) + \epsilon\},$$
$$\min\{\min_{\{i:\ \lambda_i+\lambda_i^*=0\}}(\tilde{u}_i - \sum_{j=1}^{d}\omega_{Uj}x_{ij}) + \epsilon, \min_{\{i:\ \lambda_i=C\}}(\tilde{u}_i - \sum_{j=1}^{d}\omega_{Uj}x_{ij}) - \epsilon\}\big]. \quad (4.54)$$

2. *If there exists* $i_0 \in I$ *with* $0 < \mu_{i_0} < C$, *then there exists a unique optimal solution for* β_L *in Problem (4.7),*

$$\beta_L = \tilde{l}_{i_0} - \sum_{j=1}^{d}\omega_{Lj}x_{i_0 j} - \epsilon. \quad (4.55)$$

Analogously, if there exists $i_1 \in I$ *with* $0 < \mu_{i_1}^* < C$, *one has a unique optimal solution for* β_L *in Problem (4.7),*

$$\beta_L = \tilde{l}_{i_1} - \sum_{j=1}^{d}\omega_{Lj}x_{i_1 j} + \epsilon. \quad (4.56)$$

Otherwise, if $\mu_i, \mu_i^* \in \{0, C\}$, $\forall i \in I$, *the set of solutions for* β_L *is the following interval*

$$\beta_L \in \big[\max\{\max_{\{i:\ \mu_i+\mu_i^*=0\}}(\tilde{l}_i - \sum_{j=1}^{d}\omega_{Lj}x_{ij}) - \epsilon, \max_{\{i:\ \mu_i^*=C\}}(\tilde{l}_i - \sum_{j=1}^{d}\omega_{Lj}x_{ij}) + \epsilon\},$$
$$\min\{\min_{\{i:\ \mu_i+\mu_i^*=0\}}(\tilde{l}_i - \sum_{j=1}^{d}\omega_{Lj}x_{ij}) + \epsilon, \min_{\{i:\ \mu_i=C\}}(\tilde{l}_i - \sum_{j=1}^{d}\omega_{Lj}x_{ij}) - \epsilon\}\big]. \quad (4.57)$$

Proof.
By Lemma 4.1, for any $i \in I$, either λ_i or λ_i^* are zero (and the same for μ_i and μ_i^*).

1. The cases $0 < \lambda_{i_0} < C$ or $0 < \lambda_{i_1} < C$ for some i_0, i_1, are analyzed in Remark 4.2 (cases 2 and 4).

   Now, we analyze the case in which $\lambda_i, \lambda_i^* \in \{0, C\}$, $\forall i \in I$. For every i such that $\lambda_i = 0$, we have that either $\lambda_i^* = C$, and in such a case, by Remark 4.2, case 5,

$$\beta_U \ \geq \ \tilde{u}_i - \sum_{j=1}^{d}\omega_{Uj}x_{ij} + \epsilon,$$

or $\lambda_i^* = 0$, and thus, by Remark 4.2, case 3,

$$\begin{aligned} \beta_U &\geq \tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} - \epsilon \\ \beta_U &\leq \tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} + \epsilon. \end{aligned}$$

Moreover, for every i such that $\lambda_i = C$, we have by Remark 4.2, case 1, that

$$\beta_U \leq \tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} - \epsilon.$$

Hence,

$$\beta_U \geq \max\{\tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} - \epsilon : \ \lambda_i = 0, \lambda_i^* = 0\}$$

and

$$\beta_U \geq \max\{\tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} + \epsilon : \ \lambda_i = 0, \lambda_i^* = C\}.$$

This leads to

$$\beta_U \geq \max\{\max_{\{i:\ \lambda_i + \lambda_i^* = 0\}} (\tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij}) - \epsilon, \max_{\{i:\ \lambda_i^* = C\}} (\tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij}) + \epsilon\}.$$

Analogously, we obtain the upper bound for the interval of solutions for β_U as

$$\beta_U \leq \min\{\min_{\{i:\ \lambda_i + \lambda_i^* = 0\}} (\tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij}) + \epsilon, \min_{\{i:\ \lambda_i = C\}} (\tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij}) - \epsilon\}.$$

2. With an analogous reasoning on the sets of variables μ_i, μ_i^* to that used with variables λ_i and λ_i^*, we obtain expressions (4.55)-(4.57).

□

Lemma 4.2 *Let $(\lambda, \lambda^*, \mu, \mu^*, \nu)$ be optimal for the dual problem (4.13). Then, for any k such that $\nu_k > 0$, one has that*

$$\sum_{i \in I} (\mu_i - \mu_i^* - \lambda_i + \lambda_i^* - 2\nu_i) x_i^\top x_k = \theta, \tag{4.58}$$

where θ is a constant.

Proof.
If there exists k such that $\nu_k > 0$, then, by (4.32), one has that

$$\sum_{j=1}^{d} \omega_{Lj} x_{kj} + \beta_L = \sum_{j=1}^{d} \omega_{Uj} x_{kj} + \beta_U.$$

By using the expressions for ω_U and ω_L, given in (4.22)-(4.23), we obtain that

$$\sum_{i \in I} (\mu_i - \mu_i^* - \nu_i) x_i^\top x_k + \beta_L = \sum_{i \in I} (\lambda_i - \lambda_i^* + \nu_i) x_i^\top x_k + \beta_U,$$

and this leads to

$$\sum_{i \in I} (\mu_i - \mu_i^* - \lambda_i + \lambda_i^* - 2\nu_i) x_i^\top x_k = \beta_U - \beta_L = \theta.$$

□

Theorem 4.3 *Let $(\lambda, \lambda^*, \mu, \mu^*, \nu)$ be optimal for the dual problem (4.13), such that λ_i, λ_i^*, μ_i, $\mu_i^* \in \{0, C\}$, $\forall i \in I$. Let ω_U, ω_L be defined by (4.22)-(4.23). One has that:*

1. *If $\nu_i = 0$, for all $i \in I$, then any β_U, β_L satisfying (4.54) and (4.57), respectively, are such that an optimal solution exists for Problem (4.7) of the form $(\omega_L, \omega_U, \beta_L, \beta_U, \xi, \xi^*, \eta, \eta^*)$.*

2. *If there exists $k \in I$ such that $\nu_k > 0$, then any β_U, β_L satisfying (4.54) and (4.57), respectively, and also satisfying*

$$\beta_U - \beta_L = \theta, \tag{4.59}$$

with θ the constant defined in (4.58), are such that an optimal solution exists for Problem (4.7) of the form $(\omega_L, \omega_U, \beta_L, \beta_U, \xi, \xi^*, \eta, \eta^*)$.

Proof.
We have to show that for ω_U, ω_L defined in (4.22)-(4.23) and for any β_U, β_L in the intervals (4.54) and (4.57), there exist ξ, ξ^*, η, η^* such that the pairs $(\omega_L, \omega_U, \beta_L, \beta_U, \xi, \xi^*, \eta, \eta^*)$ and $(\lambda, \lambda^*, \mu, \mu^*, \nu)$ jointly satisfy the KKT system (4.28)-(4.37).

If $\nu_i = 0$, for all $i \in I$, then (4.32) is automatically satisfied, else, by construction of β_U and β_L, they satisfy (4.59), and thus (4.32) also holds.

Hence, we only need to show that for any β_U and β_L in the intervals (4.54) and (4.57), ξ, ξ^*, η, η^* exist satisfying (4.28)-(4.31) and (4.33)-(4.36).

Consider β_U in the interval (4.54), this means that

$$\tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} - \epsilon \leq \beta_U \leq \tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} + \epsilon, \quad \forall i \in I: \ \lambda_i = \lambda_i^* = 0 \tag{4.60}$$

$$\beta_U \geq \tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} + \epsilon, \qquad \forall i \in I: \ \lambda_i = 0, \lambda_i^* = C \tag{4.61}$$

$$\beta_U \leq \tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} - \epsilon, \qquad \forall i \in I: \ \lambda_i = C, \lambda_i^* = 0. \tag{4.62}$$

We check that constraints (4.28)-(4.29) and (4.33)-(4.34) are satisfied, by considering the possible values for λ_i and λ_i^* (by taking into account (4.38)).

- For every $i \in I$ such that $\lambda_i = \lambda_i^* = 0$, (4.28)-(4.29) are trivially satisfied and (4.33)-(4.34) are satisfied for $\xi_i = \xi_i^* = 0$.
- For every $i \in I$ such that $\lambda_i = 0$ and $\lambda_i^* = C$, (4.28) is trivially satisfied, and (4.33) is satisfied with $\xi_i = 0$.

  By (4.29), we compute the value of ξ_i^*,

  $$\xi_i^* = \sum_{j=1}^{d} \omega_{Uj} x_{ij} + \beta_U - \tilde{u}_i - \epsilon,$$

  which is non-negative by expression (4.61).
- For every $i \in I$ such that $\lambda_i = C$ and $\lambda_i^* = 0$, (4.29) is trivially satisfied, and (4.34) is satisfied with $\xi_i^* = 0$.

  By (4.28), we obtain the value ξ_i as

  $$\xi_i = \tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} - \beta_U - \epsilon,$$

  which is bigger than or equal to zero by expression (4.62).

A similar reasoning can be done with β_L to check that (4.30)-(4.31) and (4.35)-(4.36) are satisfied. □

Corollary 4.1 *Let $(\lambda, \lambda^*, \mu, \mu^*, \nu)$ be a solution of the Karush-Kuhn-Tucker system (4.28)-(4.37) and let $(\omega_L, \omega_U, \beta_L, \beta_U, \xi, \xi^*, \eta, \eta^*)$ be optimal for (4.7). Then:*

1. If $\lambda_i = C$, then

$$\xi_i = \tilde{u}_i - \sum_{j=1}^{d} \omega_{Uj} x_{ij} - \beta_U - \epsilon. \tag{4.63}$$

2. If $\lambda_i^* = C$, then

$$\xi_i^* = \sum_{j=1}^{d} \omega_{Uj} x_{ij} + \beta_U - \tilde{u}_i - \epsilon. \tag{4.64}$$

3. If $\mu_i = C$, then

$$\eta_i = \tilde{l}_i - \sum_{j=1}^{d} \omega_{Lj} x_{ij} - \beta_L - \epsilon. \tag{4.65}$$

4. If $\mu_i^* = C$, then

$$\eta_i^* = \sum_{j=1}^{d} \omega_{Lj} x_{ij} + \beta_L - \tilde{l}_i - \epsilon. \tag{4.66}$$

In the following result, the uniqueness of solution for Problem (4.7), under conditions of Proposition 4.1, is studied.

Proposition 4.2 *Let $\epsilon > 0$ be a parameter in Problem (4.7) satisfying*

$$\epsilon \geq \frac{1}{2} \max \left\{ \max_{i \in I} \{\tilde{u}_i\} - \min_{i \in I} \{\tilde{u}_i\}, \max_{i \in I} \{\tilde{l}_i\} - \min_{i \in I} \{\tilde{l}_i\} \right\}. \tag{4.67}$$

Then, Problem (4.7) has a unique optimal solution iff

$$\epsilon = \frac{1}{2}(\max_{i \in I} \{\tilde{u}_i\} - \min_{i \in I} \{\tilde{u}_i\}) = \frac{1}{2}(\max_{i \in I} \{\tilde{l}_i\} - \min_{i \in I} \{\tilde{l}_i\}). \tag{4.68}$$

Proof.
When $\epsilon > 0$ satisfies (4.67), by Proposition 4.1 one has that the optimal solution of Problem (4.7) satisfies that $\omega_{Uj} = \omega_{Lj} = 0$, $j = 1, \ldots, d$, and the slack variables ξ_i, ξ_i^*, η_i, η_i^* are also zero. The constraints of Problem (4.7) are reduced to

$$\max_{i \in I} \{\tilde{u}_i\} - \epsilon \leq \beta_U \leq \min_{i \in I} \{\tilde{u}_i\} + \epsilon \tag{4.69}$$

$$\max_{i \in I} \{\tilde{l}_i\} - \epsilon \leq \beta_L \leq \min_{i \in I} \{\tilde{l}_i\} + \epsilon \tag{4.70}$$

$$\beta_L \leq \beta_U. \tag{4.71}$$

Then, the uniqueness must be studied only in the variables β_L and β_U.

In one direction, if expression (4.68) is satisfied, constraints (4.69)-(4.71) are transformed into the following expressions,

$$\begin{array}{rl} \frac{1}{2}(\min_{i\in I}\{\tilde{u}_i\}+\max_{i\in I}\{\tilde{u}_i\}) & \leq \beta_U \leq \quad \frac{1}{2}(\min_{i\in I}\{\tilde{u}_i\}+\max_{i\in I}\{\tilde{u}_i\}) \\ \frac{1}{2}(\min_{i\in I}\{\tilde{l}_i\}+\max_{i\in I}\{\tilde{l}_i\}) & \leq \beta_L \leq \quad \frac{1}{2}(\min_{i\in I}\{\tilde{l}_i\}+\max_{i\in I}\{\tilde{l}_i\}) \\ & \beta_L \leq \beta_U. \end{array}$$

Denoting by $u = \frac{1}{2}(\min_{i\in I}\{\tilde{u}_i\}+\max_{i\in I}\{\tilde{u}_i\})$ and by $l = \frac{1}{2}(\min_{i\in I}\{\tilde{l}_i\}+\max_{i\in I}\{\tilde{l}_i\})$, the bounds of these degenerate intervals, one has that the unique possible solution is $\beta_U = u$ and $\beta_L = l$, since it is clear that $l \leq u$. Then, Problem (4.7) admits a unique optimal solution.

In the other direction, we denote by $\epsilon_1 = \frac{1}{2}(\max_{i\in I}\{\tilde{u}_i\} - \min_{i\in I}\{\tilde{u}_i\})$ and by $\epsilon_2 = \frac{1}{2}(\max_{i\in I}\{\tilde{l}_i\} - \min_{i\in I}\{\tilde{l}_i\})$. We prove that if ϵ is strictly bigger than ϵ_1 or ϵ_2, then the solution is not unique. We consider two different cases, when $\epsilon_1 = \epsilon_2$ and when $\epsilon_1 \neq \epsilon_2$.

- Case 1: $\epsilon_1 = \epsilon_2$.

  Suppose that $\epsilon > \epsilon_1$, that is, there exists an amount $\delta > 0$, such that $\epsilon = \epsilon_1 + \delta = \epsilon_2 + \delta$. Then, replacing ϵ in constraints (4.69)-(4.71), we obtain

  $$\begin{array}{rl} u-\delta & \leq \beta_U \leq \quad u+\delta \\ l-\delta & \leq \beta_L \leq \quad l+\delta \\ & \beta_L \leq \beta_U. \end{array}$$

  Since $\delta > 0$, non-degenerate intervals can be found for β_L and β_U satisfying this set of constraints, for example, all the solutions such $\beta_L \in [l-\delta, l]$ and $\beta_U \in [u, u+\delta]$.

- Case 2: $\epsilon_1 \neq \epsilon_2$.

  We study, for instance, the case $\epsilon_1 < \epsilon_2$ (the opposite is analogous), that is, there exists an amount $\delta_1 > 0$ such that $\epsilon_2 = \epsilon_1 + \delta_1$. Suppose that $\epsilon \geq \epsilon_2$, that is, there exists an amount $\delta_2 \geq 0$ such that $\epsilon = \epsilon_2 + \delta_2$, or analogously, $\epsilon = \epsilon_1 + \delta_1 + \delta_2$.

  Then, replacing ϵ in constraints (4.69)-(4.71), one obtains that

  $$\begin{array}{rl} u-\delta_1-\delta_2 & \leq \beta_U \leq \quad u+\delta_1+\delta_2 \\ l-\delta_2 & \leq \beta_L \leq \quad l+\delta_2 \\ & \beta_L \leq \beta_U. \end{array}$$

Then, all the solutions in the following intervals, $\beta_L \in [l-\delta_2, l]$ and $\beta_U \in [u, u+\delta_1+\delta_2]$, satisfy this set of constraints, and, since $\delta_1 > 0$, at least the interval for β_U is non-degenerate.

□

4.4 Kernel-based dual formulation

Kernels are used to project the data from the input space $\mathcal{X} \subseteq \mathbb{R}^d$ into a high dimensional feature space, where more abstract features of the data can be exploited. This way, since the data are projected to the feature space via a usually non-linear mapping, non-linear relations between the data can be extracted by means of a linear regressor (see [31, 58, 100]).

From Problem (4.13) and by replacing $x_i^\top x_l$ by another general kernel structure $K(x_i, x_l)$, we can rewrite the dual of Problem (4.7) as the following concave quadratic maximization problem,

$$\begin{array}{ll} \max\limits_{\lambda,\lambda^*,\mu,\mu^*,\nu} & -\frac{1}{2}\sum\limits_{i,l\in I}\left[(\lambda_i-\lambda_i^*+\nu_i)(\lambda_l-\lambda_l^*+\nu_l)+(\mu_i-\mu_i^*-\nu_i)(\mu_l-\mu_l^*-\nu_l)\right]K(x_i,x_l) \\ & -\epsilon\sum\limits_{i\in I}(\lambda_i+\lambda_i^*+\mu_i+\mu_i^*)+\sum\limits_{i\in I}(\lambda_i-\lambda_i^*)\tilde{u}_i+\sum\limits_{i\in I}(\mu_i-\mu_i^*)\tilde{l}_i \\ \text{s.t.} & \sum\limits_{i\in I}(\lambda_i-\lambda_i^*+\nu_i)=0 \\ & \sum\limits_{i\in I}(\mu_i-\mu_i^*-\nu_i)=0 \\ & 0\le\lambda_i,\lambda_i^*,\mu_i,\mu_i^*\le C,\ \forall i\in I \\ & 0\le\nu_i, \quad \forall i\in I. \end{array} \tag{4.72}$$

When a general kernel structure has been introduced in the problem, explicit expressions of ω_U and ω_L cannot be computed, because one has that

$$\omega_U = \sum_{i\in I}(\lambda_i-\lambda_i^*+\nu_i)\phi(x_i) \tag{4.73}$$

$$\omega_L = \sum_{i\in I}(\mu_i-\mu_i^*-\nu_i)\phi(x_i) \tag{4.74}$$

and the mapping ϕ is unknown in general. However, given a new element x, we can always obtain the predicted interval output, since the kernel is known.

Expressions for β_U and β_L are obtained from expressions (4.52)-(4.57) by only replacing $x_l^\top x_i$ by the corresponding kernel value. Thus, when $\exists i_0 \in I : 0 < \lambda_{i_0} < C$,

β_U can be built by using expression (4.52), as follows,

$$\begin{aligned}\beta_U &= \tilde{u}_{i_0} - \sum_{l\in I}(\lambda_l - \lambda_l^* + \nu_l)\phi(x_l)^\top\phi(x_{i_0}) - \epsilon \\ &= \tilde{u}_{i_0} - \sum_{l\in I}(\lambda_l - \lambda_l^* + \nu_l)K(x_l, x_{i_0}) - \epsilon.\end{aligned}$$

If $\exists i_1 \in I : 0 < \lambda_{i_1}^* < C$, the value of β_U is obtained from (4.53),

$$\beta_U = \tilde{u}_{i_1} - \sum_{l\in I}(\lambda_l - \lambda_l^* + \nu_l)K(x_l, x_{i_1}) + \epsilon,$$

and, if $\lambda_i, \lambda_i^* \in \{0, C\}$, $\forall i \in I$, from (4.54), β_U belongs to

$$\max\{\max_{\{i:\lambda_i+\lambda_i^*=0\}} \tilde{u}_i - \sum_{l\in I}(\lambda_l - \lambda_l^* + \nu_l)K(x_l, x_i) - \epsilon, \max_{\{i:\lambda_i^*=C\}} \tilde{u}_i - \sum_{l\in I}(\lambda_l - \lambda_l^* + \nu_l)K(x_l, x_i) + \epsilon\} \leq \beta_U$$
$$\leq \min\{\min_{\{i:\lambda_i+\lambda_i^*=0\}} \tilde{u}_i - \sum_{l\in I}(\lambda_l - \lambda_l^* + \nu_l)K(x_l, x_i) + \epsilon, \min_{\{i:\lambda_i=C\}} \tilde{u}_i - \sum_{l\in I}(\lambda_l - \lambda_l^* + \nu_l)K(x_l, x_i) - \epsilon\}.$$

Analogous expressions are derived for β_L from expressions (4.55)-(4.57). If $\exists i_0 \in I$: $0 < \mu_{i_0} < C$ or $\exists i_1 \in I : 0 < \mu_{i_1}^* < C$, we obtain, from expressions (4.55) and (4.56), respectively, the following expressions,

$$\begin{aligned}\beta_L &= \tilde{l}_{i_0} - \sum_{l\in I}(\mu_l - \mu_l^* - \nu_l)K(x_l, x_{i_0}) - \epsilon \\ \beta_L &= \tilde{l}_{i_1} - \sum_{l\in I}(\mu_l - \mu_l^* - \nu_l)K(x_l, x_{i_1}) + \epsilon,\end{aligned}$$

and if $\mu_i, \mu_i^* \in \{0, C\}$, $\forall i \in I$, one has that, from (4.57), β_L satisfies

$$\max\{\max_{\{i:\mu_i+\mu_i^*=0\}} \tilde{l}_i - \sum_{l\in I}(\mu_l - \mu_l^* - \nu_l)K(x_l, x_i) - \epsilon, \max_{\{i:\mu_i^*=C\}} \tilde{l}_i - \sum_{l\in I}(\mu_l - \mu_l^* - \nu_l)K(x_l, x_i) + \epsilon\} \leq \beta_L$$
$$\leq \min\{\min_{\{i:\mu_i+\mu_i^*=0\}} \tilde{l}_i - \sum_{l\in I}(\mu_l - \mu_l^* - \nu_l)K(x_l, x_i) + \epsilon, \min_{\{i:\mu_i=C\}} \tilde{l}_i - \sum_{l\in I}(\mu_l - \mu_l^* - \nu_l)K(x_l, x_i) - \epsilon\}.$$

Finally, given a new element x, the predicted interval is the following,

$$\begin{aligned}x \hookrightarrow f(x) &:= [\omega_L^\top x + \beta_L, \omega_U^\top x + \beta_U] \\ &= [\sum_{i\in I}(\mu_i - \mu_i^* - \nu_i)K(x_i, x) + \beta_L, \sum_{i\in I}(\lambda_i - \lambda_i^* + \nu_i)K(x_i, x) + \beta_U],\end{aligned} \tag{4.75}$$

where the only thing that changes with respect to the primal problem is that the scalar products $\omega_L^\top x$ and $\omega_U^\top x$ have been replaced by the corresponding kernel values to avoid using explicit expressions of ω_L and ω_U, which could not be computed, according to (4.73)-(4.74).

4.5 Computational experiment

4.5.1 Error measures

For the numerical experiments, several measures have been used to study the fitness of the model. These measurements are the *lower* and *upper bound root mean-squared error*, $RMSE_l$ and $RMSE_u$, and the *mean Hausdorff distance*, $\overline{d}_H$, between the predicted and the real interval outputs, defined in Chapter 3 (expressions (3.34)-(3.36)).

4.5.2 Results for resubstitution

In this section, the formulation (4.7) is applied to solve the problem with single-valued input and interval-valued output in a database of cars. This dataset, which has been used previously in [44], shows the relationship between a set of (single-valued) characteristics of a determined car (predictor variables) and an interval score given by a set of experts (dependent variable). Observe that in the original dataset (see [44]), the scores are given in a fuzzy way (but these fuzzy spreads are always built under the same assumption: one to the left and to the right when possible, but taking into account that the final interval must be contained in [0,10]).

The database with the interval output for 50 different cars is shown in Table 4.1. The characteristics of each car studied are: X_1 = *price,* X_2 = *displacement,* X_3 = *potential,* X_4 = *speed,* X_5 = *acceleration,* X_6 = *urban fuel consumption,* X_7 = *extra urban fuel consumption* and X_8 = *cost/Km.* The variable Y shows the interval score given by the experts according to the characteristics.

First, we compute the predicted interval for the score of each car via resubstitution (see [35]), i.e., when the training sample contains all the elements of the dataset.

The primal and dual formulations have been tested by using this database. All these programs have been solved by using AMPL+MINOS. Concerning the kernel-based dual formulation, the two following kernels ([100, 102]) have been studied:

- Polynomial:

$$K(x, y) = (x^\top y + b)^p,\ p \in \mathbb{N},\ b \geq 0 \tag{4.76}$$

- Radial Basis Function (RBF):

$$K(x, y) = \exp\{-\frac{\|x - y\|^2}{2\sigma^2}\},\ \sigma > 0. \tag{4.77}$$

Different experiments have been performed with several values of the parameters C and ϵ, namely, for every pair (C, ϵ), with $C = 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1,$ and

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	Y
1	21330	1598	120	200	10.5	8.8	15.6	0.41	[5, 8]
2	29864	1781	150	222	8.9	8.8	15.6	0.5	[7, 10]
3	26830	1895	118	206	10.4	8.8	16.7	0.49	[5, 8]
4	26004	1997	110	191	12.5	13.7	22.2	0.29	[4, 6]
5	17613	1998	133	195	9	8.2	15.6	0.42	[1, 4]
6	18120	1596	103	187	10.7	8.9	15.9	0.38	[4, 6]
7	13170	1396	75	165	15.6	10.5	15.6	0.36	[3, 6]
8	19290	1997	136	200	9.6	7.9	14.9	0.44	[1, 4]
9	26100	1998	150	210	9.6	7.2	13.3	0.48	[4, 6]
10	29128	1988	155	215	9.5	7.5	12.8	0.52	[7, 10]
11	28715	1998	129	210	11	7.3	14.9	0.62	[6, 9]
12	26494	1995	140	203	11.3	8.1	13.7	0.5	[3, 6]
13	22931	1997	136	208	10.8	8.7	15.4	0.51	[5, 8]
14	24248	1985	152	215	8.5	7.9	14.9	0.52	[5, 8]
15	19898	1595	105	192	11.3	8.6	15.5	0.41	[5, 8]
16	22200	1948	136	205	9.7	8.5	15.9	0.46	[4, 6]
17	30320	1970	150	215	8.5	7.5	14.7	0.53	[0, 3]
18	19095	1390	75	173	12	16.7	12.2	0.43	[5, 8]
19	37390	2393	165	222	9.2	7.3	13.5	0.65	[7, 10]
20	63812	2771	193	232	10.1	5.7	11.9	0.88	[6, 9]
21	32656	1781	180	228	7.4	9.2	15.9	0.55	[4, 6]
22	54021	2793	193	228	8.6	7.3	12.7	0.84	[6, 9]
23	20199	1997	90	175	14.5	14.3	21.7	0.3	[6, 9]
24	15250	1596	103	180	11.5	9.6	16.9	0.37	[3, 6]
25	28379	1997	147	208	10	8.2	14.1	0.15	[4, 6]
26	60942	3996	280	240	7.3	5.8	11.2	0.86	[1, 4]
27	83666	3996	280	240	7.3	5.8	11.2	1.13	[4, 6]
28	10750	1242	80	174	11.2	13.5	20	0.37	[5, 8]
29	66623	4293	281	250	6.7	5.7	11.2	1.01	[7, 10]
30	36772	1998	163	223	9.1	7.4	14.3	0.65	[6, 9]
31	23235	1796	120	193	9	9.8	17.5	0.54	[4, 6]
32	22176	1781	125	202	9.7	9.4	16.1	0.45	[5, 8]
33	40852	2171	170	226	9.1	8.2	14.1	0.63	[4, 6]
34	37701	2446	129	201	12.1	9.2	16.9	0.39	[1, 4]
35	22125	1998	133	206	10.2	7.7	14.1	0.46	[1, 4]
36	12100	1242	60	155	14.3	13.7	20.8	0.3	[5, 8]
37	14530	1242	80	170	12.5	10.6	18.9	0.32	[1, 4]
38	11078	1242	75	167	13.1	11.5	17.2	0.3	[5, 8]
39	15597	1596	101	185	11	11	18.5	0.37	[6, 9]
40	72562	3996	281	250	6.7	5.8	11.6	1	[5, 8]
41	32030	1998	220	243	7.3	6.8	12.3	0.61	[5, 8]
42	32660	1796	118	202	5.9	10.4	17.5	0.52	[5, 8]
43	20193	1598	102	182	10.8	10.4	18.9	0.4	[5, 8]
44	40619	2597	170	219	9.5	6.1	11.8	0.71	[7, 10]
45	64764	3199	224	240	8.2	5.8	12.2	0.92	[6, 9]
46	93117	4966	306	250	6.5	5.3	11.4	1.23	[7, 10]
47	11104	1360	75	170	13.2	11.2	18.9	0.3	[7, 10]
48	76132	3387	300	280	5.2	5.8	11.8	1.03	[7, 10]
49	11336	1149	60	160	15	12.7	19.2	0.45	[7, 10]
50	15423	1390	75	171	13.5	11.8	18.9	0.34	[7, 10]

Table 4.1: 'Car scores' database (single-valued input and interval-valued output)

	$RMSE_l$	$RMSE_u$	$\overline{d}_H$
Primal	1.6104	1.6834	1.3856
Polynomial, $b = 1$	1.8051	1.8999	1.5370
Polynomial, $b = 100$	1.7880	1.9022	1.5456
Polynomial, $b = 10000$	1.7844	1.8912	1.5388
RBF, $\sigma = 1000$	1.6054	1.6663	1.2025
RBF, $\sigma = 2000$	**1.5730**	**1.6266**	**1.0395**
RBF, $\sigma = 5000$	1.6125	1.6925	1.0878
RBF, $\sigma = 7500$	1.6349	1.7202	1.2098
RBF, $\sigma = 10000$	1.6515	1.7203	1.3276

Table 4.2: Best results for $RMSE_l$, $RMSE_u$ and $\overline{d}_H$ for different methods via resubstitution

$\epsilon = 0.0001, 0.001, 0.01,\ 0.05, 0.1, 0.5, 1, 1.5, 2, 2.5$. Several kernels have been used by changing the parameters b ($p = 1$ for all the experiments), in the polynomial kernel, and σ, in the RBF kernel. The best results (those with the smallest value for the sum $RMSE_l + RMSE_u$ or with the smallest value of $\overline{d}_H$) for the primal and several kernel-based dual formulations are shown in Table 4.2. The best performance in the primal formulation was obtained with $\epsilon = 0.5$ and with $C = 1000$. Concerning the dual formulation, the best results were obtained for the RBF-kernel with $\sigma = 2000$ and with $C = 10$. With $\epsilon = 0.0001$ the smallest value for the mean Hausdorff distance was obtained, and with $\epsilon = 1$ we obtained the smallest sum of the root mean-squared errors. One can observe that the best results (especially, when considering $\overline{d}_H$ as the fitness measurement) are obtained with RBF-kernels, which seem to work well to fit this dataset.

The predicted intervals for the minimum values of $\overline{d}_H$ in the primal and dual formulations are presented in Table 4.3 ($C = 1000$, $\epsilon = 0.5$ for the primal, $C = 10$, $\epsilon = 0.001$, $\sigma = 2000$ for the dual). One can observe that the non-linear regressor approximates better most of the intervals than the linear regressor.

4.5.3 Results for leave-one-out

Below, the performance of the regressors built via our methodology when applying the leave-one-out (LOO) strategy (see e.g. [53, 66]) to the cars dataset is shown. The LOO strategy is more interesting than the resubstitution because this way we can study the behaviour of the regressor when a new element arrives.

Several combinations of the parameters C and ϵ and for the parameters of the kernels b and σ have been considered. The performance with the primal and dual formulations has been studied.

Tables 4.4-4.5 display the results for the root mean-squared errors, for the different

	Real	Primal	Dual		Real	Primal	Dual
1	[5 , 8]	[4,4 , 7,3]	[5,0 , 8,0]	26	[1 , 4]	[2,6 , 5,5]	[1,0 , 4,0]
2	[7 , 10]	[5,5 , 8,5]	[6,8 , 9,9]	27	[4 , 6]	[4,7 , 7,7]	[4,0 , 6,0]
3	[5 , 8]	[5,7 , 8,7]	[4,0 , 6,2]	28	[5 , 8]	[5,4 , 8,4]	[5,9 , 9,0]
4	[4 , 6]	[5,4 , 8,4]	[4,0 , 6,0]	29	[7 , 10]	[4,6 , 7,5]	[7,0 , 10,0]
5	[1 , 4]	[2,8 , 5,7]	[3,3 , 5,2]	30	[6 , 9]	[5,9 , 8,9]	[6,0 , 9,0]
6	[4 , 6]	[3,6 , 6,5]	[4,0 , 6,0]	31	[4 , 6]	[4,9 , 7,9]	[4,9 , 7,4]
7	[3 , 6]	[4,4 , 7,3]	[3,0 , 6,0]	32	[5 , 8]	[4,7 , 7,6]	[4,7 , 7,5]
8	[1 , 4]	[3,4 , 6,2]	[4,0 , 6,7]	33	[4 , 6]	[5,7 , 8,7]	[4,1 , 7,0]
9	[4 , 6]	[3,6 , 6,5]	[4,0 , 6,0]	34	[1 , 4]	[4,0 , 6,9]	[5,6 , 8,6]
10	[7 , 10]	[4,2 , 7,1]	[6,4 , 9,5]	35	[1 , 4]	[4,2 , 7,1]	[4,6 , 7,4]
11	[6 , 9]	[6,5 , 9,4]	[6,0 , 9,0]	36	[5 , 8]	[4,7 , 7,7]	[5,0 , 8,0]
12	[3 , 6]	[4,3 , 7,2]	[3,9 , 6,0]	37	[1 , 4]	[3,9 , 6,9]	[2,4 , 5,5]
13	[5 , 8]	[5,5 , 8,4]	[4,7 , 7,4]	38	[5 , 8]	[3,8 , 6,8]	[5,9 , 9,0]
14	[5 , 8]	[4,6 , 7,5]	[5,0 , 7,3]	39	[6 , 9]	[4,5 , 7,5]	[3,4 , 6,2]
15	[5 , 8]	[4,3 , 7,2]	[5,0 , 8,0]	40	[5 , 8]	[4,5 , 7,5]	[5,0 , 8,0]
16	[4 , 6]	[4,4 , 7,3]	[4,6 , 7,4]	41	[5 , 8]	[4,5 , 7,5]	[5,0 , 8,0]
17	[0 , 3]	[4,4 , 7,4]	[6,5 , 9,6]	42	[5 , 8]	[4,5 , 7,5]	[4,8 , 7,8]
18	[5 , 8]	[5,5 , 8,5]	[4,9 , 7,4]	43	[5 , 8]	[4,2 , 7,2]	[5,1 , 8,2]
19	[7 , 10]	[5,5 , 8,4]	[5,8 , 8,7]	44	[7 , 10]	[4,9 , 7,8]	[4,6 , 7,5]
20	[6 , 9]	[6,5 , 9,5]	[6,0 , 9,0]	45	[6 , 9]	[5,9 , 8,8]	[6,0 , 9,0]
21	[4 , 6]	[5,1 , 8,2]	[4,8 , 7,8]	46	[7 , 10]	[4,8 , 7,8]	[7,0 , 10,0]
22	[6 , 9]	[6,0 , 9,0]	[6,0 , 9,0]	47	[7 , 10]	[4,3 , 7,3]	[5,9 , 9,0]
23	[6 , 9]	[5,5 , 8,5]	[4,6 , 7,7]	48	[7 , 10]	[6,8 , 9,9]	[7,0 , 10,0]
24	[3 , 6]	[3,5 , 6,4]	[3,0 , 6,0]	49	[7 , 10]	[6,5 , 9,5]	[5,9 , 9,0]
25	[4 , 6]	[0,6 , 3,5]	[5,6 , 8,5]	50	[7 , 10]	[4,9 , 7,9]	[3,3 , 6,2]

Table 4.3: Predicted interval output for the primal and dual formulations with the smallest values of $\overline{d}_H$

combinations of the parameters C and ϵ with the primal formulation and with the RBF-kernel formulation. The best results for the sum of the two errors have been marked in bold.

Likewise, Table 4.6 present the results obtained when we use the Hausdorff distance to measure the error between the predicted and the observed interval and we compute the mean Hausdorff distance ($\overline{d}_H$) for all the elements of the database, for the primal and the RBF-kernel formulation. The lowest value of $\overline{d}_H$ is in bold in each table.

One can observe that, in Tables 4.4-4.6, the behaviour of the measurements $RMSE_l$, $RMSE_u$ and $\overline{d}_H$ is not very stable with respect to the parameter ϵ, especially for those values $\epsilon \geq 0.5$.

In Table 4.7, the best results for the two measurements, for the different formulations are shown. Although all the results are quite similar, one can observe that the polynomial kernel seems to be better suited when using $RMSE_l$ and $RMSE_u$ as fitness measurements, whereas the RBF-kernel has a better behaviour when studying the fitness via $\overline{d}_H$.

4.5.4 Comparison with point estimation

Table 4.8 compares the performance of the two possible models described in Chapter 3, formulations (3.19) and (3.31), to solve the problem with single-valued input and interval-valued output, with the model proposed in this chapter.

The table shows the best results for the three fitness measurements when performing resubstitution and leave-one-out. The first two rows display the best results when using formulations (3.19) and (3.31) for the maximum and Hausdorff distances (with only one hyperplane to compute), respectively. The drawback of these formulations is that the predicted interval for each record of the database is degenerate (point estimation instead of interval estimation). In the last two rows, we present the results with formulation (4.7) (the new one introduced in this chapter for the single-input and interval-output situation) and its kernel-based dual formulation (4.72) (with two hyperplanes to compute), respectively. All the kernels used in the previous computational experiment have been taken into account to select the best results.

One can observe that the results obtained with formulations with two hyperplanes are much better on this dataset than those obtained with the formulations with only one hyperplane, that is, one can say that interval estimates are much better than only point estimates. This way, the introduction of formulation (4.7) becomes justified.

Finally, with the results obtained in these experiments, we have performed the following measurement. We have considered the midpoint of the interval outputs in the original dataset, and we have also obtained the midpoint of the predicted intervals via

	C	0.0001		0.001		0.01		0.1		1	
	$\epsilon \backslash RMSE$	l	u	l	u	l	u	l	u	l	u
Primal	0.0001	2.09	2.24	2.15	2.30	2.10	2.25	2.06	2.49	2.09	2.71
	0.001	2.09	2.24	2.15	2.30	2.10	2.25	2.06	2.49	2.09	2.71
	0.01	2.08	2.23	2.15	2.29	2.10	2.25	2.06	2.48	2.09	2.70
	0.05	2.10	2.24	2.12	2.26	2.10	2.25	2.05	2.46	2.07	2.67
	0.1	2.08	2.23	2.07	2.21	2.11	2.26	2.02	2.39	2.04	2.62
	0.5	1.96	2.19	1.95	2.27	1.96	2.20	1.99	2.23	2.26	2.47
	1	2.16	2.11	2.05	2.08	2.12	2.13	2.22	2.23	2.58	2.65
	1.5	2.07	2.17	2.02	2.15	2.04	2.27	2.11	2.32	2.38	2.70
	2	2.00	2.21	1.98	2.18	1.92	2.12	1.95	1.99	2.64	2.64
	2.5	**1.89**	**1.99**	1.91	1.99	1.98	2.01	2.06	2.07	2.34	2.33
RBF $\sigma = 1000$	0.0001	1.96	2.10	1.99	2.15	1.95	2.10	1.96	2.10	2.06	2.20
	0.001	1.96	2.10	1.94	2.08	1.99	2.15	1.96	2.10	2.06	2.20
	0.01	1.96	2.09	1.94	2.08	1.94	2.08	2.00	2.16	2.06	2.20
	0.05	1.96	2.08	1.94	2.07	1.94	2.07	1.95	2.08	2.06	2.20
	0.1	1.95	2.07	1.93	2.06	1.93	2.06	1.94	2.06	2.05	2.20
	0.5	2.05	2.01	1.97	2.01	1.97	2.01	1.97	2.01	2.05	2.11
	1	2.71	2.07	2.52	2.14	2.52	2.14	2.50	2.12	2.06	2.06
	1.5	2.01	1.99	2.01	1.99	2.01	1.99	1.99	2.00	1.94	2.09
	2	1.97	2.11	1.97	2.11	1.97	2.10	1.96	2.08	**1.94**	**1.98**
	2.5	1.96	2.05	1.96	2.05	1.96	2.05	1.97	2.05	2.06	2.09
RBF $\sigma = 2000$	0.0001	1.94	2.15	1.93	2.10	1.95	2.10	1.97	2.11	2.14	2.29
	0.001	1.99	2.15	1.99	2.15	1.96	2.12	1.97	2.11	2.14	2.29
	0.01	1.99	2.06	1.99	2.15	1.99	2.15	1.97	2.14	2.13	2.29
	0.05	**1.93**	**2.06**	1.98	2.14	1.98	2.15	2.00	2.17	2.17	2.35
	0.1	1.97	2.13	1.97	2.13	1.98	2.13	2.00	2.16	2.16	2.29
	0.5	2.14	1.99	2.14	2.09	2.14	2.09	2.14	2.10	2.30	2.32
	1	2.33	2.07	2.51	2.18	2.51	2.18	2.51	2.17	2.45	2.30
	1.5	1.94	2.15	2.29	2.01	2.29	2.01	2.29	2.01	1.99	2.08
	2	1.97	2.11	1.97	2.11	1.97	2.10	1.96	2.08	2.00	2.74
	2.5	1.96	2.05	1.96	2.05	1.96	2.05	1.97	2.05	2.16	2.12

Table 4.4: $RMSE_l$ and $RMSE_u$ via LOO (primal and RBF-kernel, $\sigma = 1000, 2000$)

	C	0.0001		0.001		0.01		0.1		1	
	$\epsilon \backslash RMSE$	l	u	l	u	l	u	l	u	l	u
	0.0001	2.07	2.28	1.98	2.08	2.03	2.14	1.99	2.18	2.07	2.25
	0.001	2.01	2.23	2.14	2.39	1.99	2.08	2.03	2.15	2.07	2.25
	0.01	1.99	2.21	1.96	2.11	2.15	2.32	1.99	2.09	2.10	2.25
	0.05	2.01	2.21	1.96	2.10	1.95	2.09	2.03	2.16	2.22	2.32
RBF	0.1	2.00	2.44	1.95	2.09	1.94	2.08	2.14	2.31	2.18	2.32
$\sigma = 5000$	0.5	2.18	2.17	1.99	2.04	1.99	2.04	1.98	2.04	2.28	2.19
	1	2.46	2.29	2.65	2.14	2.69	2.18	2.70	2.14	2.48	2.06
	1.5	1.95	2.16	1.98	2.15	2.34	2.01	2.40	2.02	2.05	2.44
	2	2.00	2.35	1.97	2.32	1.97	2.33	1.94	2.37	1.99	2.13
	2.5	2.01	2.09	1.96	2.05	1.96	2.05	**1.96**	**2.04**	2.01	2.06
	0.0001	2.26	2.19	2.20	2.21	2.20	2.25	2.08	2.24	1.96	2.09
	0.001	1.96	2.17	2.01	2.28	2.08	2.21	2.09	2.26	2.05	2.22
	0.01	1.96	2.17	1.97	2.14	1.99	2.23	2.09	2.22	2.08	2.32
	0.05	1.96	2.17	1.96	2.13	1.96	2.13	1.99	2.20	2.11	2.35
RBF	0.1	1.96	2.17	1.96	2.12	1.96	2.12	1.99	2.23	2.09	2.14
$\sigma = 7500$	0.5	1.92	2.20	1.95	2.09	1.98	2.09	1.95	2.09	2.21	2.27
	1	2.32	2.16	2.32	2.34	2.32	2.34	2.32	2.28	2.22	2.10
	1.5	2.05	2.39	2.05	2.60	2.05	2.77	2.01	2.59	1.86	2.70
	2	1.90	1.99	1.90	1.99	1.90	1.99	1.90	1.99	1.94	2.55
	2.5	2.03	2.05	1.97	2.02	1.98	2.02	2.00	2.04	**1.91**	**1.95**
	0.0001	2.11	2.23	1.97	2.19	1.95	2.12	2.04	2.17	2.10	2.24
	0.001	1.94	2.12	1.88	2.04	1.98	2.19	2.02	2.17	2.08	2.29
	0.01	1.94	2.12	1.94	2.04	**1.92**	**1.98**	1.99	2.20	2.29	2.47
	0.05	2.03	2.12	1.93	2.18	1.96	2.07	2.12	2.23	2.06	2.18
RBF	0.1	1.93	2.07	1.92	2.04	1.88	2.03	1.88	2.05	2.04	2.27
$\sigma = 10000$	0.5	1.95	2.03	1.93	2.01	2.02	1.97	1.93	2.13	2.09	2.37
	1	2.54	2.07	2.67	2.09	2.67	2.06	2.69	2.04	2.89	2.31
	1.5	2.05	2.25	2.48	2.59	2.47	2.77	2.15	2.67	1.93	2.74
	2	1.91	1.99	1.91	1.99	1.91	1.99	1.92	1.99	2.04	2.02
	2.5	2.03	2.05	2.03	2.05	2.04	2.05	2.07	2.08	2.10	2.08

Table 4.5: $RMSE_l$ and $RMSE_u$ via LOO (RBF-kernel, $\sigma = 5000, 7500, 10000$)

$\epsilon\backslash C$		0.0001	0.001	0.01	0.1	1		0.0001	0.001	0.01	0.1	1
0.0001		1.78	1.87	1.80	2.10	2.26		1.80	**1.62**	1.72	1.73	1.77
0.001		1.78	1.87	1.80	2.10	2.26		1.72	1.82	1.62	1.73	1.77
0.01		1.77	1.87	1.80	2.09	2.25		1.70	1.70	1.78	1.63	1.84
0.05		1.78	1.84	1.81	2.07	2.21		1.72	1.71	1.67	1.69	1.90
0.1	Primal	1.79	1.80	1.83	2.01	2.15	RBF	1.96	1.71	1.68	1.79	1.84
0.5		1.85	1.91	1.88	1.88	2.00	$\sigma = 5000$	1.96	1.78	1.78	1.74	2.04
1		1.94	1.85	1.92	2.00	2.12		2.52	2.58	2.57	2.51	2.18
1.5		1.85	1.84	1.90	1.95	2.06		1.95	1.84	2.16	2.14	2.26
2		1.83	1.84	1.81	1.76	2.07		2.04	2.00	2.01	2.05	1.83
2.5		**1.69**	1.71	1.70	1.79	1.93		1.82	1.76	1.76	1.77	1.81
0.0001		1.64	1.68	1.64	1.65	1.77		1.86	1.78	1.80	1.77	1.70
0.001		1.64	**1.60**	1.68	1.65	1.77		1.72	1.86	1.72	1.71	1.78
0.01		1.64	1.60	1.60	1.69	1.77		1.72	1.70	1.82	1.73	1.88
0.05		1.65	1.61	1.61	1.62	1.77		1.74	1.71	1.71	1.77	1.82
0.1	RBF	1.66	1.62	1.62	1.62	1.78	RBF	1.76	1.72	1.72	1.83	1.73
0.5	$\sigma = 1000$	1.84	1.78	1.78	1.77	1.80	$\sigma = 7500$	1.90	1.84	1.86	1.82	2.11
1		2.54	2.46	2.45	2.41	1.89		2.46	2.50	2.49	2.39	2.06
1.5		1.84	1.84	1.84	1.84	1.92		2.17	2.33	2.42	2.30	2.33
2		1.64	1.64	1.64	1.66	1.71		**1.62**	1.62	1.62	1.63	2.23
2.5		1.76	1.76	1.76	1.78	1.84		1.74	1.74	1.74	1.79	1.71
0.0001		1.68	1.64	1.64	1.66	1.81		1.96	1.76	1.72	1.71	1.69
0.001		1.68	1.68	1.68	1.66	1.81		1.86	1.64	1.74	1.71	1.81
0.01		1.66	1.68	1.68	1.70	1.81		1.86	1.62	**1.62**	1.75	2.01
0.05		**1.61**	1.69	1.69	1.70	1.92		1.93	1.73	1.69	1.75	1.67
0.1	RBF	1.70	1.70	1.70	1.70	1.85	RBF	1.68	1.66	1.66	1.66	1.75
0.5	$\sigma = 2000$	1.88	1.88	1.88	1.86	2.00	$\sigma = 10000$	1.75	1.86	1.90	1.85	2.05
1		2.30	2.40	2.40	2.36	2.24		2.60	2.63	2.61	2.56	2.70
1.5		1.81	2.06	2.06	2.07	1.81		2.03	2.69	2.79	2.50	2.38
2		1.64	1.64	1.64	1.66	2.24		1.72	1.72	1.72	1.73	1.84
2.5		1.76	1.76	1.76	1.78	1.91		1.74	1.74	1.74	1.79	1.86

Table 4.6: $\overline{d}_H$ via LOO (primal and RBF-kernel, σ =1000, 2000, 5000, 7500, 10000)

	$RMSE_l$	$RMSE_u$	$\overline{d}_H$
Primal	1.8852	1.9924	1.6944
Polynomial, $b = 1$	1.8259	1.9722	1.6229
Polynomial, $b = 100$	**1.8256**	**1.9720**	1.6225
Polynomial, $b = 10000$	1.8260	1.9720	1.6227
RBF, $\sigma = 1000$	1.9415	1.9758	**1.6002**
RBF, $\sigma = 2000$	1.9341	2.0646	1.6100
RBF, $\sigma = 5000$	1.9610	2.0388	1.6201
RBF, $\sigma = 7500$	1.9062	1.9464	1.6200
RBF, $\sigma = 10000$	1.9179	1.9798	1.6219

Table 4.7: Best results for $RMSE_l$, $RMSE_u$ and $\overline{d}_H$ for different methods via leave-one-out

	Resubstitution			Leave-one-out		
Formulation	$RMSE_l$	$RMSE_u$	$\overline{d}_H$	$RMSE_l$	$RMSE_u$	$\overline{d}_H$
Maximum distance	2.1907	2.1403	2.6676	2.5564	2.3673	3.0196
Hausdorff distance	2.0617	2.2531	2.6556	2.3519	2.4633	2.9861
Primal	1.6104	1.6834	1.3856	1.8852	1.9924	1.6944
Dual (kernel)	1.5730	1.6266	1.0395	1.8256	1.9720	1.6002

Table 4.8: Comparison between the best results obtained for formulations with one hyperplane (top) and for formulations with two hyperplanes (bottom)

formulation (4.7) and the dual formulation (4.72). We consider the mean l_1-distance, defined as

$$\overline{d}_1 = \frac{1}{n} \sum_{i \in \Omega} |\tilde{y}_i - \hat{y}_i|, \tag{4.78}$$

where $\tilde{y}_i$ is the center of the interval output of the element $i \in \Omega$, $\hat{y}_i$ is the center of the predicted interval for $i \in \Omega$, and n is the cardinal of Ω.

We measure the mean l_1-distance $\overline{d}_1$ between the midpoints of the real interval outputs and the prediction given by formulation (3.31) (the model described in Chapter 3) for the Hausdorff distance (only one hyperplane). We also measure $\overline{d}_1$ between the center of the real intervals and the centers of the predicted intervals obtained with formulations (4.7) and (4.72). We have selected, in each model, the combination of parameters (C, ϵ) which gave the minimum $\overline{d}_H$ in Table 4.8. The values of this measure, for resubstitution and leave-one-out, are displayed in Table 4.9.

One can observe that the best results, for resubstitution and for leave-one-out, are obtained for the kernel-based dual formulation. Hence, formulation (4.72) is still better in this dataset when only a point-prediction (instead of an interval-prediction) is asked for, since the midpoints of the intervals obtained with this formulation are closer to the midpoints of the real intervals than the point estimations obtained with formulation (3.31).

This indicates that, in a regression problem with imprecise output, when one wants a single-valued estimate, one obtains quite similar result-quality when one does a single-point estimation directly (using the Hausdorff distance formulation (3.31)) than rather an interval estimation (primal formulation (4.7)) and then taking its midpoint (that is, eliminating the additional uncertainty information given by the interval output). In fact, for leave-one-out, the results are slightly better for the interval estimation methodology (formulation (4.7)) and clearly better (for resubstitution and leave-one-out) when kernels are introduced (formulation (4.72)), since they allow to extend the model to study more abstract relations between data.

	Resubstitution	Leave-one-out
Hausdorff distance formulation (3.31)	1.2456	1.5761
Primal formulation (4.7)	1.2755	1.5666
Kernel-based dual formulation (4.72)	0.9349	1.5102

Table 4.9: $\overline{d}_1$ for formulations with one hyperplane (first row) and for formulations with two hyperplanes (second and third rows)

4.6 Conclusions and extensions

In this chapter, a regression problem for data with single-valued predictor variables and interval-valued dependent variable has been analyzed.

The proposed model is based on the standard ϵ-Support Vector Regression approach, and two hyperplanes must be computed to approximate the lower and upper bounds of the dependent variable. The dual formulation of this optimization problem has been obtained, allowing the introduction of a kernel structure to use non-linear regressors in the data.

The computational experiments show that the introduction of these kernel structures allows to improve the results when measuring the error between the predicted and observed intervals.

The results for the kernel-based formulation via leave-one-out are quite unstable with respect to the parameter ϵ. Thus, the correct choice of the parameter ϵ remains as a critical issue for future work, which can be approached by using some of the strategies proposed in the literature. Thus, in [28], for example, values of C and ϵ are given, based on the outputs of the training sample.

The formulation has been obtained by using the Hausdorff distance in the constraints and by minimizing the sum of the Euclidean norms of the hyperplanes in the objective function. The use of other distances and norms to define the optimization problem is another topic which deserves further study.

Chapter 5

Multiple Instance Classification with separating balls

Contents

5.1 Introduction

In Chapter 2, we studied a classification problem with imprecise data, where the elements of the database were continuous sets in $\mathbb{R}^d$. In this chapter, we also study a related classification problem where the elements are also affected by imprecision, but this imprecision is shown through a discrete set of feature vectors in $\mathbb{R}^d$ for every element, such that only some of these vectors determine the class which the element belongs to. As explained in Subsection 1.4.1, this is called a Multiple Instance Classification Problem.

The tools described in Chapter 2 cannot be directly applied to this situation, due to the finite cardinality of these sets. Instead, we model this problem by using two concentric balls as classifiers and by following the strategy of maximizing the margin, as done in Support Vector Machines with hyperplanes. We obtain a formulation as a nonlinear mixed-integer program and necessary conditions of optimality are derived. We use these conditions of optimality to build a Variable Neighbourhood Search algorithm to obtain a solution.

This work is structured as follows. In Section 5.2, the problem is formulated as one of maximizing a margin, in a similar way to the strategy used in Support Vector Machines methods. In Section 5.3, the existence of optimal solution in our problem is studied, and in Section 5.4, several necessary conditions of optimality are derived, leading to a finite dominating set which can be determined in polynomial time, as seen in Section 5.5. In Section 5.6, a VNS algorithm is proposed, by using the optimality conditions, to solve the problem. Two extensions of this VNS algorithm, when we introduce p separating balls in the classification rule and when there are more than two possible labels in the database, are studied in 5.7. The algorithm is tested on several real and artificial databases and the results are shown in Section 5.8. Finally, Section 5.9 includes some discussion and conclusions.

5.2 Modeling the problem

5.2.1 Defining the classification rule

Consider a database Ω formed by elements $i = (X_i, Y_i) \in \Omega$, where each X_i is a finite set of feature vectors (called a *bag*), $X_i = \{x_1, x_2, \ldots, x_{K_i}\}$, with $x_k \in \mathbb{R}^d$, $k = 1, \ldots, K_i$, and where Y_i is the corresponding class defined by means of a label +1 or -1.

In this chapter, the classification rule is defined in terms of a ball with center x_0 and radius r. Thus, our problem will be to compute the parameters $x_0 \in \mathbb{R}^d$ and

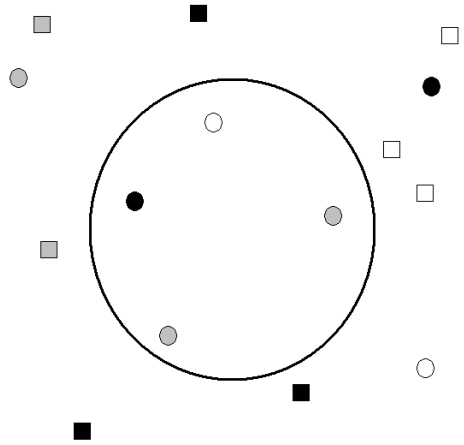

Figure 5.1: Construction of the classifier

$r \in \mathbb{R}_+$ which define the corresponding ball, such that the classification rule, expressed according to the MI assumption, is the following:

Given a bag $X \subset \mathbb{R}^d$,

$$\begin{aligned} &\bullet \text{ classify in } G_+, \text{ if } \exists x \in X \text{ such that } \|x - x_0\|^2 < r^2, \\ &\bullet \text{ classify in } G_-, \text{ otherwise, i.e., if } \forall x \in X, \|x - x_0\|^2 \geq r^2. \end{aligned} \tag{5.1}$$

In other words, once x_0, r are obtained, a bag X will be labelled as member of G_- only if X is fully contained in the complementary set of the open ball $B(x_0, r)$ centered at x_0 with radius r, and as member of G_+ otherwise.

In Figure 5.1, an example in dimension 2 is drawn. The circles of the same colour represent the instances of each bag of G_+, whereas the squares of a determined colour represent the instances of a bag of G_-. Our problem is to build a ball such that it contains at least one circle of each colour and it does not contain any squares.

Different balls may exist separating the instances in G_+ and G_-. For instance, for the example depicted in Figure 5.1, an alternative separating ball is given in Figure 5.2. In order to choose one ball, we follow the strategy successfully used in Support Vector Machines (see Section 1.1), and maximize a margin, which will be defined in a similar way to that used in SVMs.

Then, given a database, a training sample $I \subseteq \Omega$, extracted from it, will be used to build the optimization problem which must be solved in order to obtain optimal parameters (optimal in the sense that the margin is maximized), center and radius of $B(x_0, r)$, for constructing the classification rule.

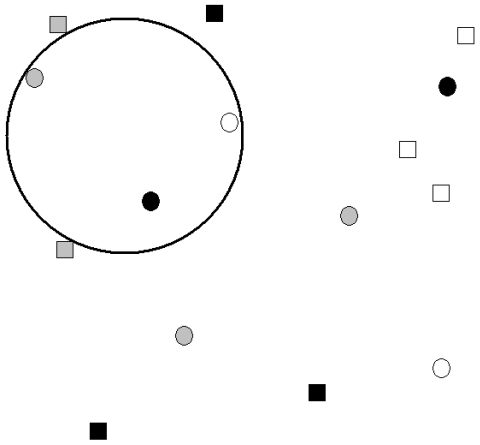

Figure 5.2: An alternative separating ball

5.2.2 The optimization problem

According to the classification rule defined in (5.1), given an element i and its set of feature vectors X_i, it is assigned to the group

$$\begin{aligned} G_+ &: \text{ if } \exists x \in X_i : \|x - x_0\|^2 < r^2, \\ G_- &: \text{ if } \forall x \in X_i, \|x - x_0\|^2 \geq r^2, \end{aligned}$$

or equivalently,

$$\begin{aligned} G_+ &: \text{ if } \min_{x \in X_i} \|x - x_0\|^2 < r^2, \quad \text{i.e., } \quad \text{if } \max_{x \in X_i} (r^2 - \|x - x_0\|^2) > 0, \\ G_- &: \text{ if } \min_{x \in X_i} \|x - x_0\|^2 \geq r^2, \quad \text{i.e., } \quad \text{if } \min_{x \in X_i} (\|x - x_0\|^2 - r^2) \geq 0. \end{aligned}$$

Given a training set $I = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)\}$, we define

$$\begin{aligned} G_+ &= \{i \in I : Y_i = +1\}, \\ G_- &= \{i \in I : Y_i = -1\}. \end{aligned}$$

The optimization problem we want to solve is

$$\max_{x_0, r} \min \left\{ \min_{i \in G_+} \max_{x \in X_i} (r^2 - \|x - x_0\|^2), \min_{i \in G_-} \min_{x \in X_i} (\|x - x_0\|^2 - r^2) \right\} \tag{5.2}$$

which can also be written as

$$\max_{(x_j)_j \in \prod_{j \in G_+} X_j} \max_{x_0, r} \min \left\{ \min_{j \in G_+} (r^2 - \|x_j - x_0\|^2), \min_{i \in G_-} \min_{x \in X_i} (\|x - x_0\|^2 - r^2) \right\}. \tag{5.3}$$

Denote by Δ the margin, that is, the minimum between the two distances considered in the formulation of Problem (5.3),

$$\Delta = \min\left\{\min_{j\in G_+}(r^2 - \|x_j - x_0\|^2), \min_{i\in G_-}\min_{x\in X_i}(\|x - x_0\|^2 - r^2)\right\}. \tag{5.4}$$

Then, the problem remains as follows,

$$\begin{array}{lll} \max\limits_{(x_j)_j\in\prod_{j\in G_+}X_j} & \max\limits_{x_0,r,\Delta} & \Delta \\ & \text{s.t.} & \Delta \leq \min\limits_{j\in G_+}(r^2 - \|x_j - x_0\|^2) \\ & & \Delta \leq \min\limits_{i\in G_-}\min\limits_{x\in X_i}(\|x - x_0\|^2 - r^2) \end{array}$$

or equivalently,

$$\begin{array}{lll} \max\limits_{(x_j)_j\in\prod_{j\in G_+}X_j} & \max\limits_{x_0,r,\Delta} & \Delta \\ & \text{s.t.} & \|x_j - x_0\|^2 \leq r^2 - \Delta,\ \forall j \in G_+ \\ & & \|x - x_0\|^2 \geq r^2 + \Delta,\ \forall x \in \bigcup\limits_{i\in G_-} X_i. \end{array} \tag{5.5}$$

If we denote by $r_+^2 = r^2 - \Delta$, $r_-^2 = r^2 + \Delta$, one has that $\Delta = \dfrac{r_-^2 - r_+^2}{2}$, and Problem (5.5) is equivalent to

$$\begin{array}{lll} \max\limits_{(x_j)_j\in\prod_{j\in G_+}X_j} & \max\limits_{x_0,r_+,r_-} & r_-^2 - r_+^2 \\ & \text{s.t.} & \|x_j - x_0\|^2 \leq r_+^2,\ \forall j \in G_+ \\ & & \|x - x_0\|^2 \geq r_-^2,\ \forall x \in \bigcup\limits_{i\in G_-} X_i. \end{array} \tag{5.6}$$

Hence, our problem can be seen as that of obtaining two concentric balls $B(x_0, r_+)$, $B(x_0, r_-)$, where $B(x_0, r_+)$ contains every instance x_j (of those previously selected) from the bags of the group G_+, $B(x_0, r_-)$ does not contain strictly any instance from the bags of the group G_- and the difference between the squares of the radii is as large as possible (see Figure 5.3 for the example in dimension 2). Thus, we use (x_0, r_+, r_-) to denote a finite solution of Problem (5.6).

Our aim will be to characterize the existence of optimal solutions and to obtain some necessary conditions to describe an optimal solution of Problem (5.6), in order to obtain an optimal solution of our original problem. The reasonings used for obtaining these conditions look like others which appear in some problems in Location Theory (see e.g. [22, 91]).

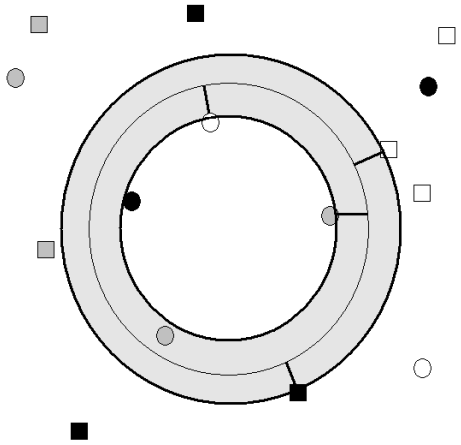

Figure 5.3: Construction of the two concentric balls

5.3 Existence of finite optimal solution

The following result explains the assumptions under which a finite optimal solution of Problem (5.6) can be obtained. For the proof of this result, some notions on Voronoi diagrams (see [87, 99]) are needed.

Given the sets of points $\{y_1, \ldots, y_m\}$, the *nearest-point Voronoi polytope* associated with the point y_k is defined as

$$V_k = \bigcap_{i=1,\ldots,m} \{x : \|x - y_k\| \leq \|x - y_i\|\}. \tag{5.7}$$

Analogously, the *farthest-point Voronoi polytope* associated with the point y_k is defined as

$$W_k = \bigcap_{i=1,\ldots,m} \{x : \|x - y_k\| \geq \|x - y_i\|\}. \tag{5.8}$$

The sets $V = \{V_k : \ k = 1, \ldots, m\}$ and $W = \{W_k : \ k = 1, \ldots, m\}$ are called the *nearest-point* and the *farthest-point Voronoi diagrams*, respectively (see [87, 99]).

Theorem 5.1 *There exists a finite optimal solution (x_0, r_+, r_-) of Problem (5.6) iff for every choice of representatives from bags of G_+, $S = (x_j)_j \in \prod_{j \in G_+} X_j$, the intersection of the convex hull of this set of instances S and the convex hull of all the instances of bags of G_- is non-empty, that is, if $CH(S) \cap CH(\bigcup_{i \in G_-} X_i) \neq \emptyset$, $\forall S \in \prod_{j \in G_+} X_j$.*

Proof.
Denote by $S = (x_j)_j \in \prod_{j \in G_+} X_j$, a choice of representatives of bags in G_+, define $P(S)$ as

$$\begin{array}{ll} \max\limits_{x_0, r_+, r_-} & r_-^2 - r_+^2 \\ \text{s.t.} & \|x_j - x_0\|^2 \leq r_+^2, \ \forall x_j \in S \\ & \|x - x_0\|^2 \geq r_-^2, \ \forall x \in \bigcup\limits_{i \in G_-} X_i \end{array} \tag{5.9}$$

and denote by $z(S)$ its optimal value, possibly $+\infty$.

With this notation, Problem (5.6) can be written as

$$\max_{(x_j)_j \in \prod\limits_{j \in G_+} X_j} \quad z(\{x_j : \ j \in G_+\}).$$

Firstly, we will see that if there exists a selection of representatives S for which the intersection $CH(S) \cap CH(\bigcup_{i \in G_-} X_i)$ is empty, then the solution of Problem (5.6) is unbounded.

Indeed, if this intersection is empty for this selection of representatives, one can find a hyperplane $H : \{p^\top x = c\}$, with $p \in \mathbb{R}^d$, $\|p\| = 1$ and $c \in \mathbb{R}$, separating strictly the two convex hulls, satisfying that the halfspace $\{p^\top x > c\}$ contains $CH(S)$.

Construct the nearest-point Voronoi diagram V for the complete set of instances of bags in G_- and the farthest-point Voronoi diagram W associated to the set S, as defined in expressions (5.7)-(5.8). Consider the intersection of the two diagrams in the space of dimension d. The obtained tessellation has a finite number of cells.

Let C be an unbounded cell of that tessellation such that, for each x in the cell, one has that $x + \lambda p$ also belongs to the cell, for any $\lambda \geq 0$, with p the vector defined previously for the hyperplane H. For this cell, there exists a point $a \in \bigcup_{i \in G_-} X_i$ which is the nearest one of this set to all the points in the cell and a point $b \in S$ which is the farthest one of S to every point in the cell.

Consider x_0 any point belonging to $C \cap \{p^\top x > c\}$ and define the straight line $g : x = x_0 + \lambda p$, for $\lambda \in \mathbb{R}$. The radii r_+ and r_- can be expressed as the distances from x_0 to b and a, that is, $r_+ = \|x_0 - b\|$ and $r_- = \|x_0 - a\|$ (see [85, 88, 99] for a detailed description on the topic).

Consider a_0 and b_0 the orthogonal projections of a and b on g. Observe that $a_0 \neq b_0$, since g is orthogonal to the separating hyperplane H and then, a_0 and b_0 are also separated by H. The objective function for the feasible solution (x_0, r_+, r_-) can be expressed as follows,

$$\begin{array}{rcl} r_-^2 - r_+^2 & = & \|x_0 - a_0\|^2 + \|a - a_0\|^2 - \|x_0 - b_0\|^2 - \|b - b_0\|^2 \\ & = & \|x_0 - a_0\|^2 - \|x_0 - b_0\|^2 + C. \end{array}$$

Since $p = \frac{(b_0-a_0)}{\|b_0-a_0\|}$, if we move x_0 along the straight line $g : x = x_0 + \lambda p$, for $\lambda > 0$ increasing, the closest point in $\bigcup_{i\in G_-} X_i$ and the farthest point in S to the new center $x_0 + \lambda p$ continue being the same points a and b, since the new center is also included in $C \cap \{p^\top x > c\}$, and the objective function increases linearly in λ,

$$\begin{aligned} r'^2_- - r'^2_+ &= \|x_0 + \lambda p - a_0\|^2 - \|x_0 + \lambda p - b_0\|^2 + C \\ &= \|x_0 - a_0\|^2 + 2\lambda(x_0 - a_0)^\top p - \|x_0 - b_0\|^2 - 2\lambda(x_0 - b_0)^\top p + C \\ &= \|x_0 - a_0\|^2 - \|x_0 - b_0\|^2 + C + 2\lambda(b_0 - a_0)^\top \tfrac{(b_0-a_0)}{\|b_0-a_0\|} \\ &= r^2_- - r^2_+ + 2\lambda\|b_0 - a_0\| > r^2_- - r^2_+. \end{aligned}$$

In fact, the larger the value of λ, the better the solution. The solution is thus unbounded.

Now, we will see that if for every choice S of representatives from bags of G_+, the intersection $CH(S) \cap CH(\bigcup_{i\in G_-} X_i)$ is non-empty, then a finite optimal solution of Problem (5.6) can be found.

For every possible set of representatives S, consider Problem (5.9) and build the nearest-point Voronoi diagram V for $\bigcup_{i\in G_-} X_i$, the farthest-point Voronoi diagram W for S, and the tessellation obtained by intersecting the two diagrams, whose number of cells is finite. For each cell, consider $a \in \bigcup_{i\in G_-} X_i$, the nearest point of this set to all the points in the cell, and $b \in S$, the farthest point of S to every point in the cell.

Thus, for any possible center x_0 in this cell, the radii r_+ and r_- can be computed as the distances to those two points, $r_- = \|x_0 - a\|$ and $r_+ = \|x_0 - b\|$. Problem (5.9) restricted to the cell can be expressed as follows,

$$\begin{array}{ll} \max\limits_{x_0} & \|x_0 - a\|^2 - \|x_0 - b\|^2 \\ \text{s.t.} & \|x_j - x_0\|^2 \leq \|x_0 - b\|^2, \ \forall x_j \in S \\ & \|x_k - x_0\|^2 \geq \|x_0 - a\|^2, \ \forall x_k \in \bigcup\limits_{i\in G_-} X_i. \end{array} \tag{5.10}$$

In fact, Problem (5.10) can be rewritten as the following linear program in x_0,

$$\begin{array}{ll} \min\limits_{x_0} & 2x_0^\top(a - b) + \|b\|^2 - \|a\|^2 \\ \text{s.t.} & 2x_0^\top(b - x_j) + \|x_j\|^2 - \|b\|^2 \leq 0, \ \forall x_j \in S \\ & 2x_0^\top(x_k - a) + \|a\|^2 - \|x_k\|^2 \leq 0, \ \forall x_k \in \bigcup\limits_{i\in G_-} X_i. \end{array} \tag{5.11}$$

The dual problem corresponding to the minimization problem (5.11) is the following,

$$\begin{array}{ll} \max\limits_{\lambda_j,\mu_k} & \|b\|^2 - \|a\|^2 + \sum\limits_j \lambda_j \ (\|x_j\|^2 - \|b\|^2) - \sum\limits_k \mu_k \ (\|x_k\|^2 - \|a\|^2) \\ \text{s.t.} & (1 - \sum\limits_k \mu_k) \ a + \sum\limits_k \mu_k x_k = (1 - \sum\limits_j \lambda_j) \ b + \sum\limits_j \lambda_j x_j \\ & \lambda_j, \ \mu_k \geq 0, \ \forall j, \ \forall k. \end{array} \tag{5.12}$$

If we denote μ_a and λ_b the lagrangian multipliers corresponding to the constraints in Problem (5.11) defined by a and b, the constraint in Problem (5.12) can also be expressed as

$$(1 - \sum_{k \neq a} \mu_k)\ a + \sum_{k \neq a} \mu_k x_k = (1 - \sum_{j \neq b} \lambda_j)\ b + \sum_{j \neq b} \lambda_j x_j. \tag{5.13}$$

Let $x \in CH(S) \cap CH(\bigcup_{i \in G_-} X_i)$ (this point can be defined because this intersection in non-empty by assumption). This means, there exist $\mu'_k \geq 0$ $(\forall k : x_k \in \bigcup_{i \in G_-} X_i)$, $\sum_k \mu'_k = 1$ and $\lambda'_j \geq 0$ $(\forall j : x_j \in S)$, $\sum_j \lambda'_j = 1$, such that $x = \sum_k \mu'_k x_k = \sum_j \lambda'_j x_j$. Defining $\mu_k = \mu'_k$, for $k \neq a$, and $\lambda_j = \lambda'_j$, for $j \neq b$, we obtain (5.13).

Hence, the dual problem is feasible and consequently its corresponding primal problem (5.9) will have a finite optimal solution. If we repeat the process for every cell (the number of cells is finite) and for every possible S, we conclude that there is a finite optimal solution for Problem (5.6).

□

Remark 5.1 *If the solution of Problem (5.6) is unbounded, according to the proof of Theorem 5.1, we can move the center x_0 of any original feasible solution along the direction p and the objective function continues improving. Then, the separating concentric balls are transformed in two hyperplanes $\{p^\top x = d\}$ and $\{p^\top x = e\}$, with $d > c > e$, and such that the closed halfspace $\{p^\top x \geq d\}$ contains $CH((x_j)_j)$ whereas $\{p^\top x \leq e\}$ contains $CH(\bigcup_{i \in G_-} X_i)$.*

Theorem 5.1 characterizes the existence of a finite optimal solution in Problem (5.6). The following results will be used to describe a procedure to check the existence of solution of Problem (5.6) (or equivalently, to check that the two groups are not linearly separable) by solving a mixed-integer linear problem.

Theorem 5.2 *Problem (5.6) does not have a finite optimal solution iff there exists a selection of representatives from bags in G_+, $S \in \prod_{j \in G_+} X_j$ and there exist $p \in \mathbb{R}^d$, $\beta \in \mathbb{R}$ such that*

$$p^\top x + \beta \geq 1, \qquad \forall x \in S, \tag{5.14}$$

$$p^\top x + \beta \geq 0, \quad \forall x \in \bigcup_{j \in G_+} X_j, \tag{5.15}$$

$$p^\top x + \beta < 1, \quad \forall x \in \bigcup_{i \in G_-} X_i. \tag{5.16}$$

Proof.
By Theorem 5.1, there does not exist a finite optimal solution of Problem (5.6) iff there

exists a selection $S \in \prod_{j \in G_+} X_j$ such that a hyperplane $H : \{q^\top x = \alpha\}$ separates strictly the convex hulls of the sets of points S and $\bigcup_{i \in G_-} X_i$. In that case, one has that

$$q^\top x \geq \alpha_S, \qquad \forall x \in S, \qquad \text{with } \alpha_S = \min_{x \in S} q^\top x, \tag{5.17}$$

$$q^\top x \geq \alpha_+, \quad \forall x \in \bigcup_{j \in G_+} X_j, \quad \text{with } \alpha_+ = \min_{x \in \bigcup_{j \in G_+} X_j} q^\top x, \tag{5.18}$$

$$q^\top x \leq \alpha_-, \quad \forall x \in \bigcup_{i \in G_-} X_i, \quad \text{with } \alpha_- = \max_{x \in \bigcup_{i \in G_-} X_i} q^\top x, \tag{5.19}$$

satisfying $\alpha_S \geq \alpha_+$ and $\alpha_S > \alpha_-$.

In case $\alpha_S > \alpha_+$, by subtracting α_+ in both sides of inequalities (5.17) and (5.18) and by dividing by $\alpha_S - \alpha_+$, we obtain

$$\frac{q^\top x - \alpha_+}{\alpha_S - \alpha_+} \geq \frac{\alpha_S - \alpha_+}{\alpha_S - \alpha_+}, \quad \text{i.e.,} \quad \frac{q^\top x - \alpha_+}{\alpha_S - \alpha_+} \geq 1, \ \forall x \in S, \tag{5.20}$$

$$\frac{q^\top x - \alpha_+}{\alpha_S - \alpha_+} \geq \frac{\alpha_+ - \alpha_+}{\alpha_S - \alpha_+}, \quad \text{i.e.,} \quad \frac{q^\top x - \alpha_+}{\alpha_S - \alpha_+} \geq 0, \ \forall x \in \bigcup_{j \in G_+} X_j. \tag{5.21}$$

Likewise, by performing the same operations in expression (5.19) and by taking into account that $\alpha_S > \alpha_-$, one has that

$$\frac{q^\top x - \alpha_+}{\alpha_S - \alpha_+} \leq \frac{\alpha_- - \alpha_+}{\alpha_S - \alpha_+} < 1, \ \forall x \in \bigcup_{i \in G_-} X_i. \tag{5.22}$$

Calling $p = \frac{1}{\alpha_S - \alpha_+} q$ and $\beta = -\frac{\alpha_+}{\alpha_S - \alpha_+}$ in expressions (5.20)-(5.22), we obtain expressions (5.14)-(5.16).

In case $\alpha_S = \alpha_+$, we sum $1 - \alpha_S$ in both sides of constraints (5.17)-(5.19), and we obtain

$$\begin{array}{lcll} q^\top x - \alpha_S + 1 & \geq & \alpha_S - \alpha_S + 1, & \text{i.e., } q^\top x + 1 - \alpha_S \geq 1, \ \forall x \in S, \\ q^\top x - \alpha_S + 1 & \geq & \alpha_+ - \alpha_S + 1, & \text{i.e., } q^\top x + 1 - \alpha_S \geq 1 \geq 0, \ \forall x \in \bigcup_{j \in G_+} X_j, \\ q^\top x - \alpha_S + 1 & \leq & \alpha_- - \alpha_S + 1 < 1, & \forall x \in \bigcup_{i \in G_-} X_i, \end{array}$$

by taking into account that $\alpha_- - \alpha_S < 0$. Calling $p = q$ and $\beta = 1 - \alpha_S$, expressions (5.14)-(5.16) are obtained.

□

Theorem 5.3 *Problem (5.6) does not have a finite optimal solution iff the solution of*

the problem

$$\begin{aligned} \min_{p\in\mathbb{R}^d,\beta,z\in\mathbb{R},y_l^j\in\{0,1\}} \quad & z & \\ s.t. \quad & y_l^j \in \{0,1\},\ \forall x_l \in X_j, \forall j \in G_+ & (5.23)\\ & \textstyle\sum_l y_l^j = 1,\ \forall j \in G_+ & (5.24)\\ & p^\top x_l + \beta \geq y_l^j,\ \forall x_l \in X_j, \forall j \in G_+ & (5.25)\\ & p^\top x + \beta \leq z,\ \forall x \in \textstyle\bigcup_{i\in G_-} X_i, & (5.26) \end{aligned}$$

is strictly smaller than 1.

Proof.
By Theorem 5.2, Problem (5.6) does not have a finite optimal solution iff constraints (5.14)-(5.16) are satisfied for a determined set of representatives $S \in \prod_{j\in G_+} X_j$, and for parameters $p \in \mathbb{R}^d$, $\beta \in \mathbb{R}$.

In the mixed-integer linear program with constraints (5.23)-(5.26), the binary variables defined in constraints (5.23) express that an instance x_l is chosen as the representative from its bag X_j when the corresponding variable y_l^j is equal to 1. Likewise, constraints (5.24) impose that exactly one representative is chosen from each bag X_j. Then, for any feasible solution of this problem, a selection of representatives from bags in G_+ is selected in constraints (5.23)-(5.24).

Constraints (5.25) represent that the expressions (5.14)-(5.15) are reached, while constraints (5.26) are related to expression (5.16). Then, expressions (5.14)-(5.16) are fully satisfied iff a solution of the mixed-integer linear problem is found with $z < 1$.

□

From now on, we will assume that the existence of optimal solution of Problem (5.6) has been already checked and that the two groups are not linearly separable, and hence, the classifier cannot be a hyperplane, but a ball.

5.4 Necessary conditions for optimality

Below, we derive some conditions that a finite feasible solution (x_0, r_+, r_-) must satisfy to be eligible for optimality.

For describing these optimal solutions, the concept of active point will also be necessary. An instance x from a bag of the group G_+ is an *active point* for the ball $B(x_0, r_+)$ iff the distance from x to x_0 is exactly r_+, that is, $\|x - x_0\| = r_+$. Thus, the set of active points of G_+, which is denoted by A_+, is formed by the instances which

lie on the boundary of the ball $B(x_0, r_+)$ (respectively, the set A_- of the active points of G_- is formed by the instances lying on the boundary of $B(x_0, r_-)$).

During the proofs, we will see that a determined solution is not optimal by finding another solution which gives a better value of the objective function.

Theorem 5.4 *If (x_0, r_+, r_-) is an optimal solution, then there exists at least one active point in each group G_+ and G_-, that is, the sets A_+ and A_- of active points are non-empty.*

Proof.
Suppose that A_+ is empty. Since (x_0, r_+, r_-) is a feasible solution of Problem (5.6), at least one instance x_j from each bag X_j of the group G_+ is strictly (due to the emptiness of A_+) contained inside the ball $B(x_0, r_+)$, that is, $\|x_j - x_0\|^2 < r_+^2$.

Then, it is sufficient to define r'_+ as the distance from x_0 to the farthest instance (of a bag of G_+) contained in the ball $B(x_0, r_+)$, that is, $r'_+ = \max_{x \in (\bigcup_{j \in G_+} X_j) \cap B(x_0, r_+)} \|x - x_0\|$, which is strictly smaller than r_+ and (x_0, r'_+, r_-) is a feasible solution which improves the value of the objective function.

On the other hand, suppose that A_- is an empty set. Then, for any x belonging to any bag of G_-, one has that $\|x - x_0\|^2 > r_-^2$, and it is sufficient to take $r'_- = \min_{x \in \bigcup_{i \in G_-} X_i} \|x - x_0\|$, which is strictly bigger than r_-. Hence, (x_0, r_+, r'_-) improves the objective function.

In both cases, we conclude that the initial solution (x_0, r_+, r_-) cannot be optimal.

□

Theorem 5.5 *If (x_0, r_+, r_-) is an optimal solution, one has that:*

1. *If $r_+ < r_-$, then there must exist at least two active points in G_-.*
2. *If $r_+ > r_-$, then there must exist at least two active points in G_+.*
3. *If $r_+ = r_-$ and $\left(\bigcup_{j \in G_+} X_j\right) \cap \left(\bigcup_{i \in G_-} X_i\right) = \emptyset$ (i.e., there is not any instance common to a bag in G_+ and to a bag in G_-), then there must exist at least two active points in G_+ and two in G_-.*

Proof.
By Theorem 5.4, if (x_0, r_+, r_-) is optimal, there must exist at least an active point a in G_- and an active point b in G_+. Below, we will see there are more active points in each case.

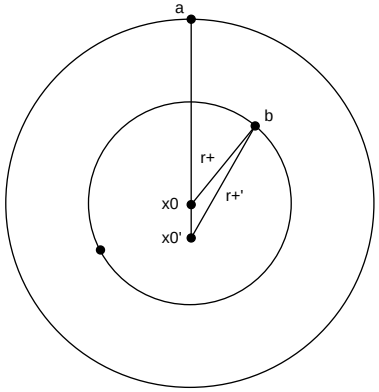

Figure 5.4: Illustration for the proof of Theorem 5.5, case $r_+ < r_-$

1. When $r_+ < r_-$, suppose there is only one active point a in the set A_- (see Figure 5.4). Consider the direction $p = x_0 - a$ and it will be proved to be a direction of improvement for the objective function. Indeed, if we move x_0 an amount $\epsilon > 0$, small enough (for not touching new active points) in the direction $u = \frac{p}{\|p\|}$, one has that $r'_- = r_- + \epsilon$.

   And the other radius must be measured as the maximum distance from $x'_0 = x_0 + \epsilon u$ to the points belonging to A_+.

   If the new center x'_0 is nearer to all the active points in A_+, the radius $r'_+ \leq r_+$ and the difference between the squares of the radii increases.

   Otherwise, the new radius r'_+ will be, at most, the distance from x'_0 to the point b which belonged to A_+ and which is now the furthest. Anyway, one has that $r'_+ \leq r_+ + \epsilon$, by triangular inequality on b, x_0 and x'_0, and the value of the objective function improves strictly since

$$\begin{aligned} r'^2_- - r'^2_+ &\geq (r_- + \epsilon)^2 - (r_+ + \epsilon)^2 \\ &= r^2_- - r^2_+ + 2\epsilon(r_- - r_+) > r^2_- - r^2_+. \end{aligned}$$

   Hence, we have found a new feasible solution (x'_0, r'_+, r'_-) with a better value of the objective function of Problem (5.6), contradicting the optimality of (x_0, r_+, r_-).

2. When $r_+ > r_-$, suppose now there is only one active point b in the set A_+ and consider the direction $q = b - x_0$. If we move x_0 an amount $\epsilon > 0$, small enough, in direction $v = \frac{q}{\|q\|}$, one has that $r'_+ = r_+ - \epsilon$, whereas the other radius, r'_- must be taken as the minimum distance from $x'_0 = x_0 + \epsilon v$ to the points belonging to A_-. And now, with a symmetric reasoning to that used in the case $r_+ < r_-$, we obtain that (x'_0, r'_+, r'_-) improves the objective function.

3. When $r_+ = r_-$, suppose there is only one active point a in the set A_- and consider the direction $p = x_0 - a$. If we move x_0 along the direction $u = \frac{p}{\|p\|}$ and with an analogous reasoning to that used in case $r_+ < r_-$, we obtain a new feasible solution (x'_0, r'_+, r'_-), with $x'_0 = x_0 + \epsilon u$, $r'_- = r_- + \epsilon$ and r'_+ the distance from x'_0 to a point b in A_+ (the farthest one to x'_0).

   If b is nearer to x'_0 than to x_0, r'_+ has decreased and then, the objective function has improved. Otherwise, two situations may arise: either x'_0, a and b are not collinear or $a = b$, but this latter situation cannot occur under the assumption that there are not any common instances in the bags of G_+ and G_-. Therefore, by strict triangular inequality $r'_+ < r_+ + \epsilon$, the objective function improves since

$$r'^2_- - r'^2_+ > (r_- + \epsilon)^2 - (r_+ + \epsilon)^2 = r^2_- - r^2_+.$$

   If we suppose there is only one active point b in A_+, the reasoning to follow is analogous (by symmetry).

□

Remark 5.2 *For the following results, we need to add the assumption that the instances of the database Ω are in general position, in the sense that any set of n instances of the database, with $1 \leq n \leq d+1$, are always affinely independent, that is, its rank is equal to $n-1$.*

Remark 5.3 *Note that assumption stated in part 3 of Theorem 5.5 is less restrictive and is included in the assumption stated in Remark 5.2.*

Theorem 5.6 *Under the assumption that the data are in general position, any optimal solution has at least $d+2$ active points associated.*

Proof.
Let $A_+ = \{x_{j_1}, \ldots, x_{j_s}\}$ and $A_- = \{x_{k_1}, \ldots, x_{k_t}\}$ be the active points of the groups G_+ and G_-, respectively (where $s, t \geq 1$, according to Theorem 5.4).

Consider the mediatrices of these two sets of active points,

$$\begin{aligned} med(A_+) &= \{x \in \mathbb{R}^d : \|x - x_{j_1}\|^2 = \|x - x_{j_2}\|^2, \ldots, \|x - x_{j_1}\|^2 = \|x - x_{j_s}\|^2\} \\ med(A_-) &= \{x \in \mathbb{R}^d : \|x - x_{k_1}\|^2 = \|x - x_{k_2}\|^2, \ldots, \|x - x_{k_1}\|^2 = \|x - x_{k_t}\|^2\} \end{aligned}$$

and the intersection of these two mediatrices, $med(A_+) \cap med(A_-)$, is formed by these points in $\mathbb{R}^d$ satisfying simultaneously the two sets of equations, that is,

$$\begin{array}{rcl} \|x - x_{j_1}\|^2 & = & \|x - x_{j_2}\|^2 \\ & \vdots & \\ \|x - x_{j_1}\|^2 & = & \|x - x_{j_s}\|^2 \\ \|x - x_{k_1}\|^2 & = & \|x - x_{k_2}\|^2 \\ & \vdots & \\ \|x - x_{k_1}\|^2 & = & \|x - x_{k_t}\|^2. \end{array} \tag{5.27}$$

Observe that (5.27) is equivalent to the linear system

$$\begin{array}{rcl} 2(x_{j_2} - x_{j_1})^\top x & = & \|x_{j_2}\|^2 - \|x_{j_1}\|^2 \\ & \vdots & \\ 2(x_{j_s} - x_{j_1})^\top x & = & \|x_{j_s}\|^2 - \|x_{j_1}\|^2 \\ 2(x_{k_2} - x_{k_1})^\top x & = & \|x_{k_2}\|^2 - \|x_{k_1}\|^2 \\ & \vdots & \\ 2(x_{k_t} - x_{k_1})^\top x & = & \|x_{k_t}\|^2 - \|x_{k_1}\|^2. \end{array} \tag{5.28}$$

which contains at most $s + t - 2$ linearly independent equations.

Hence, $dim(med(A_+) \cap med(A_-))$, the dimension of the affine space obtained by intersecting the mediatrices $med(A_+)$ and $med(A_-)$, satisfies

$$dim(med(A_+) \cap med(A_-)) \geq d - (s + t - 2).$$

Suppose that the number of active points is strictly smaller than $d + 2$, that is, $s + t \leq d + 1$. Then,

$$dim(med(A_+) \cap med(A_-)) \geq 1.$$

In that case, the intersection of the mediatrices would contain at least a straight line. We will build one straight line where we will find a better solution.

Since the center x_0 must be at the same distance of every active point in G_+ (and at the same distance of every active point in G_-, respectively), it must be in the intersection of the mediatrices of A_+ and A_-, that is, x_0 satisfies the linear system (5.28).

Consider $a \in A_-$ and $b \in A_+$, two active points (see Figure 5.5). The objective function of Problem (5.6) is linear in x_0,

$$r_-^2 - r_+^2 = \|x_0 - a\|^2 - \|x_0 - b\|^2 = 2(b - a)^\top x_0 + \|a\|^2 - \|b\|^2$$

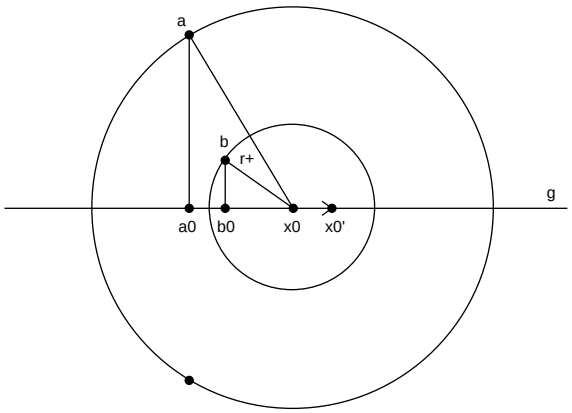

Figure 5.5: Illustration for the proof of Theorem 5.6

Consider $y \in med(A_+) \cap med(A_-)$, with $y \neq x_0$, and the straight line $g : x = x_0 + \lambda(y - x_0)$, with $\lambda \in \mathbb{R}$. This straight line is included in the intersection of the mediatrices, in fact these two manifolds coincide when $dim(med(A_+) \cap med(A_-)) = 1$.

Consider a_0 and b_0 the orthogonal projections of the points a and b on g. Then, the objective function can be written as

$$\begin{aligned} r_-^2 - r_+^2 &= \|x_0 - a\|^2 - \|x_0 - b\|^2 \\ &= \|x_0 - a_0\|^2 + \|a_0 - a\|^2 - \|x_0 - b_0\|^2 - \|b_0 - b\|^2 \\ &= 2(b_0 - a_0)^\top x_0 + \|a_0\|^2 - \|b_0\|^2 + \|a_0 - a\|^2 - \|b_0 - b\|^2 \\ &= 2(b_0 - a_0)^\top x_0 + C, \end{aligned} \tag{5.29}$$

where C is a constant depending on a, b, a_0 and b_0.

And if we take $p = b_0 - a_0$, for $a_0 \neq b_0$, we obtain a direction of improvement along the line in a neighborhood of the point x_0. Indeed, if we move x_0 an amount $\epsilon > 0$ (small enough for not finding new active points associated to the solution) in the direction p, the new value of the objective function is

$$\begin{aligned} r_-'^2 - r_+'^2 &= 2(b_0 - a_0)^\top (x_0 + \epsilon(b_0 - a_0)) + C = r_-^2 - r_+^2 + 2\epsilon\|b_0 - a_0\|^2 \\ &> r_-^2 - r_+^2. \end{aligned}$$

Thus, we have obtained a new solution $(x_0 + \epsilon p, r'_+, r'_-)$, with the same sets of active points and a better value of the objective function, therefore, the original solution (x_0, r_+, r_-) cannot be optimal.

In case $a_0 = b_0$, it is sufficient to take other two active points whose orthogonal projections on g do not coincide. This is always possible if $dim(med(A_+) \cap med(A_-)) > 1$, because, in that case, we only have to choose a different straight line g. If $dim(med(A_+) \cap med(A_-)) = 1$, we can always find at least two active points,

$a' \in A_-$ and $b' \in A_+$, with different orthogonal projections, otherwise, the $d+1$ active points would lie on the same hyperplane and this is not possible under the assumption of the data being in general position.

□

Remark 5.4 *Without the general position assumption, one still obtains existence of an optimal solution with at least $d+2$ active points (although the uniqueness is not guaranteed, since other solutions with the same value of the objective function and only $d+1$ active points can be found).*

Indeed, if $dim(med(A_+) \cap med(A_-)) = 1$ and the $d+1$ active points are cohyperplanar, their orthogonal projections on g will coincide. In that case, since $a_0 = b_0$, expression (5.29) remains as follows

$$r_-^2 - r_+^2 = \|a_0 - a\|^2 - \|b_0 - b\|^2$$

which does not depend on x_0, therefore, we can move x_0 along the straight line until a new point becomes active and the value of the objective function remains constant. Then, a solution (x_0', r_+', r_-') with $d+2$ active points can be reached, although the solution is not unique, because any solution with x_0^* belonging to the interval $[x_0, x_0']$ would have the same value of the objective function (and only $d+1$ active points).

Definition 5.1 *Given two sets of points $A^+ \subset \bigcup_{j \in G_+} X_j$ and $A^- \subset \bigcup_{i \in G_-} X_i$, consider the intersection $med(A^+) \cap med(A^-)$. If $card(A^+ \cup A^-) = d+2$, $med(A^+) \cap med(A^-) = \{x_0\}$ and all points of A^+ and A^- are active at x_0, we say that x_0 is generated by A^+ and A^-.*

Theorem 5.6 asserts that any optimal solution is generated by its active points.

Definition 5.2 *Given a problem called P of type (5.6), any new problem obtained by adding instances to bags in G_- and/or removing instances from bags in G_+ is called an extension of P.*

Lemma 5.1 *If P' is an extension of P, then the optimal value of P' is smaller than or equal to the optimal value of P.*

Proof.
Given P a problem of type (5.6), if we add instances to bags in G_-, the resulting problem has more constraints. Likewise, if we remove instances from the bags in G_+, the number of possible choices of representatives of these bags is smaller (in the

combinatorial part of the problem). In both cases, the optimal value of the resulting problem P' will be smaller than or equal to the optimal value of P.

□

Theorem 5.7 *Under the assumption of the data being in general position, if* (x_0, r_+, r_-) *is an optimal solution generated by the sets* A^+ *and* A^-*, all points of* A^+ *come from different bags.*

Proof.
Suppose we have $d+2$ active points associated to the solution (x_0, r_+, r_-) of Problem (5.6), P, and a set B of those instances, with cardinality at least equal to two, coming from the same bag X_j in G_+.

Drop that set B of active points, except for one of them, b, and consider the problem P' which is an extension of P where we have removed the set $B \backslash \{b\}$ in the bag X_j in G_+. Then, (x_0, r_+, r_-) continues being a feasible solution with the same value of the objective function, but the number of active points is now strictly smaller than $d+2$, then the solution cannot be optimal and a solution with a better value of the objective function can be found. By Lemma 5.1, the optimal value of problem P' is a lower bound of the optimal value of problem P. Then, the solution (x_0, r_+, r_-) cannot be optimal for P.

□

Theorem 5.8 *If* (x_0, r_+, r_-) *is an optimal solution, then the intersection of the convex hulls of the two groups of active points* A_+ *and* A_- *is a non-empty set.*

Proof.
Suppose $CH(A_+) \cap CH(A_-)$, the intersection of the convex hulls of the sets of active points A_+ and A_-, is empty. Then, one can find a hyperplane $H : \{p^\top x = c\}$ which strictly separates these two convex hulls, with $p \in \mathbb{R}^d$, $\|p\| = 1$ and $c \in \mathbb{R}$, such that the halfspace which contains $CH(A_+)$ is defined by $\{p^\top x > c\}$. Consider the straight line $g : x = x_0 + \lambda p$, with $\lambda \in \mathbb{R}$. If we move x_0 along this straight line a certain amount $\epsilon > 0$, small enough, the objective function will be improved.

To prove this latter, it is sufficient to consider as a the active point from A_- closest to the new center $x_0 + \epsilon p$, and as b the point from A_+ which maximizes the distance from the new center to A_+, and their corresponding orthogonal projections a_0 and b_0 to the straight line g. We obtain a new feasible solution $(x_0 + \epsilon p, r'_+, r'_-)$, where $r'_- = \|x_0 + \epsilon p - a\|$ and $r'_+ = \|x_0 + \epsilon p - b\|$. Observe that $a_0 \neq b_0$, because the straight

line is orthogonal to the separating hyperplane and hence, a_0 and b_0 are also separated by the hyperplane H.

The objective function for the initial solution (x_0, r_+, r_-) is

$$\begin{aligned} r_-^2 - r_+^2 &= \|x_0 - a_0\|^2 + \|a - a_0\|^2 - \|x_0 - b_0\|^2 - \|b - b_0\|^2 \\ &= \|x_0 - a_0\|^2 - \|x_0 - b_0\|^2 + C \end{aligned}$$

On the other hand, since $p = \frac{(b_0 - a_0)}{\|b_0 - a_0\|}$, the value of the objective function for $(x_0 + \epsilon p, r'_+, r'_-)$ is better,

$$\begin{aligned} r_-'^2 - r_+'^2 &= \|x_0 + \epsilon p - a_0\|^2 - \|x_0 + \epsilon p - b_0\|^2 + C \\ &= \|x_0 - a_0\|^2 + 2\epsilon(x_0 - a_0)^\top p - \|x_0 - b_0\|^2 - 2\epsilon(x_0 - b_0)^\top p + C \\ &= \|x_0 - a_0\|^2 - \|x_0 - b_0\|^2 + C + 2\epsilon(b_0 - a_0)^\top \frac{(b_0 - a_0)}{\|b_0 - a_0\|} \\ &= r_-^2 - r_+^2 + 2\epsilon\|b_0 - a_0\| > r_-^2 - r_+^2. \end{aligned}$$

Then, we conclude that (x_0, r_+, r_-) with $CH(A_+) \cap CH(A_-) = \emptyset$ cannot be an optimal solution.

□

5.5 A polynomial algorithm in fixed dimension

The results obtained in the previous section show that a finite dominating set of solutions can be built in polynomial time, when the dimension d is fixed.

Hence, an algorithm to find an optimal solution can be constructed as follows.

Algorithm 5.1

1. *Choose $d + 2$ active points, by taking into account the conditions obtained in Theorems 5.4-5.8.*

2. *Compute the center and the radii associated.*

3. *Check the feasibility of the solution*

4. *Once given the finite dominating set of solutions (for every possible choice of $d+2$ active points), choose the optimal one.*

For every possible choice of $d+2$ active points, satisfying the necessary conditions of optimality, one has to compute the center and the radii of the associated solution.

The center x_0 is built as the intersection of the mediatrices of the two sets of active points, A_+ and A_-.

In other words, if $x_{i_1}, \ldots, x_{i_s}$ are the points in A_+, with $1 \leq s \leq d+1$ (by Theorem 5.4), and $x_{i_{s+1}}, \ldots, x_{i_{d+2}}$ in A_-, x_0 is obtained as solution to

$$\begin{array}{rcl} \|x_0 - x_{i_1}\|^2 & = & \|x_0 - x_{i_2}\|^2 \\ & \vdots & \\ \|x_0 - x_{i_1}\|^2 & = & \|x_0 - x_{i_s}\|^2 \\ \|x_0 - x_{i_{s+1}}\|^2 & = & \|x_0 - x_{i_{s+2}}\|^2 \\ & \vdots & \\ \|x_0 - x_{i_{s+1}}\|^2 & = & \|x_0 - x_{i_{d+2}}\|^2. \end{array} \tag{5.30}$$

Observe that (5.30) is equivalent to the linear system

$$\begin{array}{rcl} 2(x_{i_2} - x_{i_1})^\top x_0 & = & \|x_{i_2}\|^2 - \|x_{i_1}\|^2 \\ & \vdots & \\ 2(x_{i_s} - x_{i_1})^\top x_0 & = & \|x_{i_s}\|^2 - \|x_{i_1}\|^2 \\ 2(x_{i_{s+2}} - x_{i_{s+1}})^\top x_0 & = & \|x_{i_{s+2}}\|^2 - \|x_{i_{s+1}}\|^2 \\ & \vdots & \\ 2(x_{i_{d+2}} - x_{i_{s+1}})^\top x_0 & = & \|x_{i_{d+2}}\|^2 - \|x_{i_{s+1}}\|^2. \end{array} \tag{5.31}$$

and should have a unique solution (Theorem 5.6).

And the radii are computed as the distances from the center x_0 to any of the active points in each group, that is,

$$\begin{array}{rcll} r_- & = & \|x_0 - a\|, & a \in A_-, \\ r_+ & = & \|x_0 - b\|, & b \in A_+. \end{array}$$

To check the feasibility of this solution (x_0, r_+, r_-), one has to study the distances from the center x_0 to each bag of G_+ and to each instance of a bag of G_- and the following conditions must be satisfied,

$$\begin{array}{c} dist(x_0, X_j)^2 = \min\limits_{x \in X_j} \|x - x_0\|^2 \leq r_+^2, \forall j \in G_+ \\ dist(x_0, x)^2 = \|x - x_0\|^2 \geq r_-^2, \forall x \in X_i, \forall i \in G_-. \end{array}$$

And finally, an optimal solution will be that with the best value of the objective function.

Since the solution of the linear system defined in (5.31) can be obtained via Gauss elimination with complexity $\mathcal{O}(d^3)$, where d is the dimension of the space of our problem, an optimal solution by using Algorithm 5.1 can be found with complexity $\mathcal{O}(d^3 n^{d+2})$, where n represents the total number of instances in the two groups. For high dimensions, this algorithm cannot be applied in an efficient way, and a heuristic methodology, described in the following section, will be used to find the solution.

5.6 A VNS strategy to solve the problem

For solving Problem (5.6), which is a mixed-integer nonlinear problem, a heuristic method, called Variable Neighborhood Search (see e.g. [54, 81] for a description), will be used to take advantage of the combinatorial structure of the problem. This method is a recent metaheuristic based on systematic change of neighborhood within a local search, for solving combinatorial and global optimization problems, which has been successfully applied in different fields, such as in Location Theory (see e.g. [20, 54]).

The basic VNS algorithm works as follows.

Algorithm 5.2 (Mladenovic and Hansen)

- ***Initialization step.***
  Select the set of neighborhood structures $\mathcal{N}_k$, $k = 1, \ldots, k_{max}$ to be used in the search.
  Find an initial solution x.
  Choose a stopping condition.

- ***Main step.***

  1. *Set $k := 1$.*
  2. *Until $k = k_{max}$, repeat the following steps:*
     - *Generate a feasible solution x' at random from the k-th neighborhood of x (that is, $x' \in \mathcal{N}_k(x)$).*
     - *Apply some local search method with x' as initial solution (the new local optimum will be denoted by x'').*
     - *If the solution obtained x'' is better than x, move there ($x := x''$) and continue the search with $\mathcal{N}_1$; otherwise, set $k := k + 1$.*

Below, we describe the search space, the neighborhood structure and the local search used for performing the algorithm.

5.6.1 Search space

By Theorems 5.4 and 5.6, an optimal solution must have at least $d+2$ active points, one at least in each group. These conditions are used in order to define the search space of the algorithm.

The different solutions will be determined in terms of a selection of active points of each group. Each possible selection will contain s points from the bags of group G_+, with $1 \leq s \leq d+1$, and where all these points must come from different bags, that is, no more than one instance from each bag can be chosen (by Theorem 5.7).

Then, we take $d - s + 2$ points from the bags of the group G_- (belonging or not to different bags).

These points will represent, respectively, the sets of active points A_+ and A_-.

5.6.2 Initial solution

Two different strategies have been attempted to construct an initial solution for the VNS algorithm.

In the first strategy, the initial solution for the algorithm is built by choosing at random a set of $d+2$ active points belonging to the search space.

In the second strategy, we perform the following steps:

- Computing the centroid C_i of each bag i in G_+ (that is, the arithmetical mean of the coordinates of the instances). Computing the centroid C of this set of centroids C_i.
- Choosing, for each bag in G_+, the closest instance to the centroid C as the representative of that bag.
- Selecting $d+2$ active points:
  - $s = \min(card(G_+), d)$ instances from G_+, the s representatives of bags that are farthest from the centroid C,
  - $d - s + 2$ instances from G_-, the closest ones to C.

Observe that this set of $d+2$ active points belongs to the search space by construction.

5.6.3 Neighborhood structure

The neighborhood structure is defined by taking into account the possible choices of active points. Thus, the k-th neighborhood $\mathcal{N}_k(x)$ will be formed by all the possible sets of active points obtained by modifying k elements in the configuration of active points which generate x.

The only restrictions to be imposed will be that there must be at least one instance from each group and no more than one representative from each bag of the group G_+ in each configuration (by Theorems 5.4 and 5.7).

5.6.4 Calculating the center and the radii

Given the $d+2$ active points, the center x_0 is computed as the solution of the linear system (5.31). If the solution is not unique, a new active point is added (and consequently, a new equation is added to the linear system).

Once the center x_0 has been found, the radii are built in order to guarantee the feasibility of the solution. The radius r_+ is thus obtained as

$$r_+(x_0) = \max_{i \in G_+} \min_{x \in X_i} \|x_0 - x\| \tag{5.32}$$

and the new set of active points of the group G_+, will be formed by the points of the bags of G_+ (one or several points) with distance to the center equal to $r_+(x_0)$.

Analogously, the radius r_- is obtained as

$$r_-(x_0) = \min_{i \in G_-} \min_{x \in X_i} \|x_0 - x\| \tag{5.33}$$

and the new set of active points of G_- is formed by the set of points (at least one) with distance to the center equal to $r_-(x_0)$.

5.6.5 Local search

Given a set of $d+2$ active points, the corresponding solution $(x_0, r_+(x_0), r_-(x_0))$ is computed as explained before. But expressions (5.32)-(5.33) only guarantee two active points associated to that solution. Then, we try to recover $d+2$ active points.

For choosing the $d+2$ active points, the following steps must be performed:

- Computing the distance matrices between:
  - the instances of G_+ and x_0,
  - the instances of G_- and x_0.

- For each bag in G_+, selecting the closest instance to x_0 as its representative.
- Selecting $d+2$ active points:
  - $s = \min(card(G_+), d)$ instances from G_+, the s representatives of bags the furthest ones from x_0,
  - $d - s + 2$ instances from G_-, the closest ones to x_0.
- Computing the new center and radii.

We repeat the process until we obtain $d+2$ active points. Although the convergence is guaranteed (since the set of instances is finite), in practice, a maximum number of iterations is fixed (this is especially advisable for high values d).

5.6.6 Main step of the algorithm

Given a set of $d+2$ active points (an element of the search space) defining a solution x_0, we choose at random another feasible solution x_0' from the first neighbourhood of x_0, that is, $x_0' \in \mathcal{N}_1(x_0)$. We compute the radii $r_+(x_0')$ and $r_-(x_0')$. We apply the local search procedure to compute the new solution x_0''.

Now, we evaluate the objective function for the new solution x_0''. If the objective value has improved, we move to x_0'', that is, we set $x_0 := x_0''$ and we continue the process in $\mathcal{N}_1(x_0)$. If the objective function has not been improved, we choose another random x_0' from the same neighbourhood and we repeat the process until having selected a maximum number of h solutions in each neighbourhood.

After h iterations in a neighbourhood, if the solution has not improved, we set $k := k + 1$ and we continue the search in $\mathcal{N}_k(x_0)$, until $k = k_{max}$, where k_{max} is fixed to $d+2$ in our problem (this way, we can obtain a configuration of $d+2$ active points completely different to the original one).

Finally, the stopping condition is given by a fixed number of iterations.

5.7 Extensions of the VNS algorithm

5.7.1 The p-balls VNS algorithm

In most real databases, the value of the objective function for the solution of Problem (5.6) is negative, because one cannot construct the two separating concentric balls satisfying the constraints in Problem (5.6) and satisfying simultaneously that $r_+ \leq r_-$. In that case, one obtains a high misclassification rate for the training sample, when

trying to separate the two groups, and consequently, bad results for classification in the test sample.

A strategy to improve these results is to modify the initial classification rule (5.1) to allow the introduction of p separating balls. Thus, the new classification rule is defined now in terms of p balls, each ball l with center $x_{0,l} \in \mathbb{R}^d$ and radius $r_l \in \mathbb{R}_+$, $l = 1, \ldots, p$. According to the MI assumption, the classification rule remains as follows:

Given a bag $X \subset \mathbb{R}^d$,

$$\begin{array}{l} \bullet \text{ classify in } G_+, \text{ if } \exists x \in X,\ \exists l \in \{1, \ldots, p\} \text{ such that } \|x - x_{0,l}\|^2 < r_l^2 \\ \bullet \text{ classify in } G_-, \text{ otherwise, i.e., if } \forall x \in X,\ \forall l = 1, \ldots, p,\ \|x - x_{0,l}\|^2 \geq r_l^2. \end{array} \tag{5.34}$$

In the algorithm, before the training of the classifier, we apply the k-means clustering algorithm (see e.g. [55]), with $k = p$, to build clusters with the bags in G_+. Given the bags of G_+, the following steps are performed:

1. Computing the centroid C_i of each bag i in G_+.
2. Initial assignment: the set of bags is partitioned at random in p clusters (with the same size).
3. Computing the centroid $\tilde{C}_l$ of each cluster l, $l = 1, \ldots, p$ (the mean of the centroids C_i of the bags assigned to that cluster).
4. Computing the distance matrix between the centroid C_i of the i-th bag, $i \in G_+$, and the centroid $\tilde{C}_l$ of the l-th cluster, $l = 1, \ldots, p$.
5. Assigning the bag i to the cluster whose centroid $\tilde{C}_l$ is the closest one to C_i.
6. Repeating steps 3-5 while there are some changes in the assignment or while a fixed number of iterations is not reached.

Once the clusters have been constructed, we apply the VNS algorithm described in Section 5.6 for $G_{+1,l}$ being the bags of the cluster l and $G_{-1,l} = G_-$, $l = 1, \ldots, p$, we compute the p balls (for the p clusters) and we classify the testing sample with the rule (5.34).

5.7.2 Multi-class case

So far, we have only dealt with the classification problem for two groups. However, in many real situations, more than two groups appear in a classification problem. Different strategies can be found in the literature, most of them proposing to transform a multi-class problem in a series of two-class problems to be solved (see e.g., [56, 62, 98]).

We will use the *one-versus-one* strategy (see [49]) for our experiments. One has to construct a classifier for every possible pair of groups i and j. Since the classification rule is not symmetric, we will need to build the ball where the group i is G_+ and the group j is G_-, denoted by $B(x_0^{i,j}, r^{i,j})$, and the opposite, the ball $B(x_0^{j,i}, r^{j,i})$. Then, we need to construct $N(N-1)$ classifiers in total.

Given a bag X to be classified, for every pair of groups i and j, we compute:

$$intensity(i,j) = \frac{\min\limits_{x \in X} dist(x, x_0^{i,j})^2}{(r^{i,j})^2},$$

and we give one vote to group i, if $intensity(i,j) < intensity(j,i)$, or one vote to group j, otherwise. X is finally assigned to the group with the highest number of votes (following the Max Wins algorithm, [49]).

In case of tie between two groups i and j, we go back to compare the intensities for these two groups and we assign the bag to group i if $intensity(i,j) < intensity(j,i)$, and to group j otherwise.

5.8 Computational experiment

The classification problem and the VNS algorithm proposed for building the classifier have been implemented by using Matlab 6.5 on a computer with Pentium IV CPU 3.06 GHz. Several numerical experiments have been performed, with artificial and real databases.

In the experiments with artificial databases, we have solved the classification problem through 10-fold cross validation (see [57, 66]). With the real databases in Subsection 5.8.5, we have performed 5-fold cross validation.

With the training sample, we have applied the VNS algorithm to build the center x_0 and the radius r which define the classifier, and we have measured the classification accuracy, i.e., the percentage of well-classified bags, for the training sample and, later, for the test sample. The parameter k_{max} in the VNS algorithm has been fixed to $d+2$ (hence, it depends on the dimension of the problem), while the parameter h (the number of different solutions taken in each neighbourhood) and the maximum number of iterations in the local search (see Subsection 5.6.5) have been both set equal to 5.

Number of bags	1	2	5	10	50
Complete enumeration	1.9315	3.8101	0.3592	0.3036	0.0464
VNS (random initial solution)	1053.4	1664.6	0.3592	0.3036	0.0464
VNS (heuristic initial solution)	1053.4	1664.6	0.3592	0.3036	0.0464

Table 5.1: Value of the objective function for complete enumeration and for VNS

5.8.1 Full enumeration vs VNS algorithm

In this first experiment, we compare the results obtained via the VNS algorithm with those obtained via a full enumeration of the finite dominating set of solutions (obtained by Algorithm 5.1) in the optimization problem (5.6).

Since full enumeration is not an efficient way of obtaining solutions in high dimension, two sets G_+ and G_- have been built in dimension $d = 2$ with 50 instances in each one. Polar coordinates have been used for generating the instances. Thus, for an instance $(\rho\cos\theta, \rho\sin\theta)$ of G_-, θ comes from a uniform distribution $U(-\pi, \pi)$, and ρ is chosen from a uniform distribution $U(r_1, r_2)$, where $0 < r_1 < r_2$ (r_1 was fixed to 1 and r_2 to 2 for the experiment). The instances in G_+ were generated in the same way, excepting one instance in each bag which must be included in $B(0, r_1)$ (that is, ρ was chosen from a uniform distribution $U(0, r_1)$). That way, the spherical separability of the bags (in the sense that there exists a feasible solution (x_0, r_+, r_-) of Problem (5.6) satisfying that $r_+ < r_-$) is guaranteed.

The experiment was repeated with 50 instances in each group and changing the number of bags in the group G_+ (the numbers of bags were 1, 2, 5, 10 and 50), each bag with the same number of instances. Observe that the problem remains the same by changing the number of bags in G_-.

In Table 5.1, we show the values of the objective function obtained for Problem (5.6) by using a complete enumeration and the VNS algorithm, for 1000 iterations, with the two possible initial solutions described in Subsection 5.6.2 (random and heuristic centroid-based initial solutions).

One can observe that the VNS algorithm obtains the same results as those obtained via a full enumeration, excepting the case of only one and two bags in G_+. However, in that case, the real solution is $+\infty$ for the two cases, since the procedure to check the existence of finite optimal solution (described in Section 4) indicates that the sets are linearly separable.

A comparison of the behaviour of the objective function, for the two types of initial solution (random and heuristic) with respect to the number of iterations is depicted in Figure 5.6. We can observe that, in the three datasets, the two versions of the algorithm reach the optimal solution (or a very close value) with only a few iterations. However,

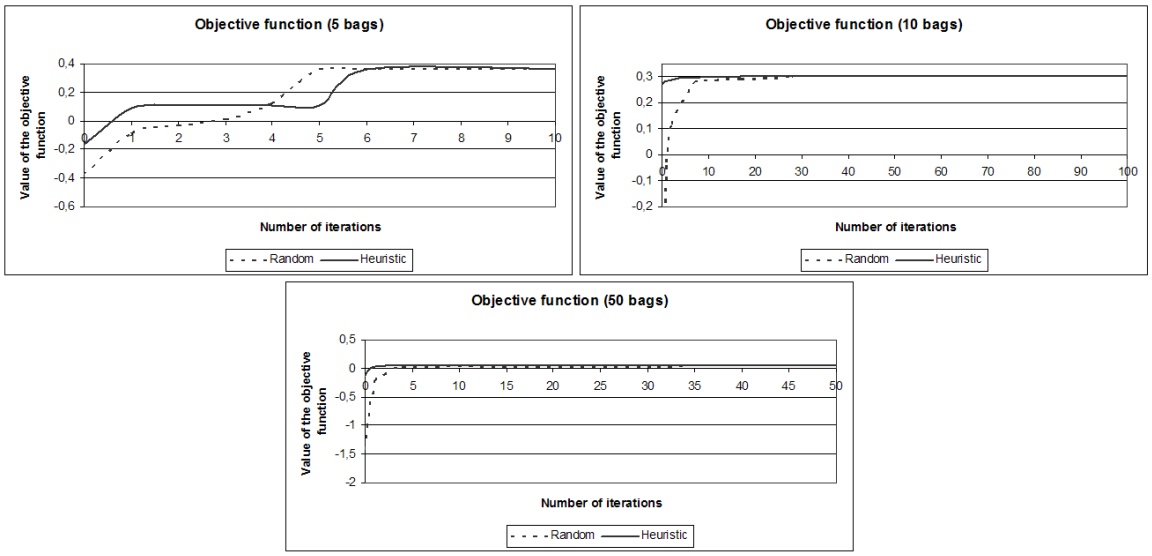

Figure 5.6: Behaviour of the objective function with the VNS algorithm (sets with 5, 10 and 50 bags)

the algorithm with the heuristic initial solution reaches, in general, the optimal solution before, since a better value to initialize the process is given.

Anyway, in this work, we are not really interested in obtaining the best solution of Problem (5.6) but rather in obtaining competitive results for the classification problem.

5.8.2 Artificial database with spherically separable sets of instances

From now on, the experiments described herein illustrate the classification problem.

In this experiment, we have built one artificial database, where the instances of G_+ and G_- are spherically separable , in the following way. For each group, 2000 instances have been generated coming from uniform distributions: the instances of G_+ via a uniform distribution with parameters -10 and 10, $U(-10, 10)$, and the instances of G_- via a uniform distribution with parameters -20 and 20, $U(-20, 20)$, and by taking into account that at least one coordinate does not belong to the interval $(-10, 10)$.

The instances of each class have been grouped in 100 bags (20 instances per bag) and we have repeated the process for different dimensions ($d = 2, 3, 10, 20, 30, 50, 100$).

In all the cases, an accuracy of 100% was found in each group for the training and the test sample in every dimension.

	Dim	d=2	d=3	d=10	d=20	d=30	d=50	d=100
Test	G_+	100	92	93	97	97	97	100
	G_-	100	90	87.69	93	97	99	100
	Total	100	91	90.35	95	97	98	100
Train	G_+	100	92.22	96.44	99.78	99.89	100	100
	G_-	100	95.57	78.86	98.56	100	100	100
	Total	100	93.9	87.65	99.17	99.95	100	100

Table 5.2: Accuracy for uniform artificial database

5.8.3 Artificial database with spherically separable sets of bags

Another separable artificial database for the classification problem has been constructed as follows. For each class (G_+ and G_-), we have generated a total of 2000 instances and 100 bags (20 instances per bag). The instances of G_- are generated as in the previous experiment, that is, via a uniform distribution $U(-20, 20)$, with at least one coordinate not belonging to the interval $(-10, 10)$. On the other hand, one instance of each bag of G_+ is generated via $U(-10, 10)$, while the rest of instances come from $U(-20, 20)$.

The average accuracy for the 10 runs for different dimensions and for the test and training samples are given in Table 5.2. The number of iterations for the VNS algorithm was fixed to 2000 (without local search).

One can remark that the accuracies are quite satisfactory. Moreover, for the highest dimension ($d = 100$), the accuracy is 100% in both cases (test and training sample). This is probably due to the fact that the higher the dimension, the more solutions are considered in the VNS algorithm, since the $k_{\max}$, that is, the maximum neighborhood radius taken into account, is fixed in our implementation to $d + 2$. Hence, the number of solutions studied depends on the dimension of the problem.

5.8.4 Artificial dataset based on a gaussian distribution

For this database, 100 bags, each one with 20 instances, have been generated for each group, G_+ and G_-, by using a gaussian distribution.

Each coordinate of the mean vector of each bag in G_+ comes from a uniform distribution $U(-1, 1)$, while the coordinates of the mean vector of the bags in G_- come from $U(-5, 5)$. The instances of each bag are generated from a multivariate normal distribution with the corresponding mean vector and the identity as the covariance matrix.

	Dim	d=2	d=3	d=5	d=10	d=20	d=30	d=50	d=100
Test	G_+	96	88	97	100	100	100	100	100
	G_-	87	89	96	100	99	100	100	100
	Total	91.5	88.5	96.5	100	99.5	100	100	100
Train	G_+	100	99.89	99.67	100	100	100	100	100
	G_-	87.56	89.33	97.45	100	100	100	100	100
	Total	93.78	94.61	98.56	100	100	100	100	100

Table 5.3: Accuracy for gaussian artificial database

Table 5.3 shows the average accuracy for the 10 runs of the cross validation process in the test and training samples, for different dimensions (with 5000 iterations in the VNS algorithm which builds the classifier and without local search).

One can observe that the accuracy for the highest dimensions is better, in both samples (training and test), because for a high dimension, the databases built herein become more easily separable.

5.8.5 Real database for image categorization

Finally, we have applied our algorithm to a real database for image categorization. Image categorization consists in labeling images into a set of predefined categories.

The image database is a set of 2000 images in JPEG format taken from 20 CD-ROMs published by the COREL Corporation, each CD-ROM containing 100 images representing a different concept. This dataset was previously used for Multi-instance Learning in [26, 27], and it is available at the webpage http://www.cs.olemiss.edu/~ychen/ddsvm.html.

A segmentation process was applied to these images to extract some properties about luminance, color and texture of the pictures and they were encoded into feature vectors. These feature vectors were grouped into clusters, representing the regions of the segmented image. Then, each image has several regions, where each region is characterized by a feature vector in dimension $d = 9$, representing the color, texture and shape properties of that region (see [26, 27] for a more detailed description).

From the Multiple Instance Learning framework, the different concepts (CD-ROMs) are the groups which the images will be assigned to, the images are the bags of the database, and the regions are the instances of each bag. In this dataset, there are 20 groups, 100 bags in each group and the average number of instances per bag for the different groups is displayed in Table 5.4 (along with the name of the groups). The dimension of the problem is $d = 9$. We have performed several experiments with only the first 10 groups (1000-Image database) and with the complete database (2000-Image

Class	Class name	Instances per bag (average)
0	African people and villages	4.84
1	Beach	3.54
2	Historical building	3.1
3	Buses	7.59
4	Dinosaurs	2.00
5	Elephants	3.02
6	Flowers	4.46
7	Horses	3.89
8	Mountain and glaciers	3.38
9	Food	7.24
10	Dogs	3.80
11	Lizards	2.80
12	Fashion models	5.19
13	Sunset scenes	3.52
14	Cars	4.93
15	Waterfalls	2.56
16	Antique furniture	2.30
17	Battle ships	4.32
18	Skiing	3.34
19	Desserts	3.65

Table 5.4: Description of the image database

database).

Since the database has more than two classes, the 1-v-1 algorithm, explained in Subsection 5.7.2, is the selected tool to solve the multi-class problem, and for every pair of groups i and j, the p-balls VNS algorithm, described in Subsection 5.7.1, is used to build the classifier.

First, we have considered the problem with only the first ten classes. For selecting the training and test samples, we have used 5-fold cross validation on the database. Different values for p (in the optimization algorithm to construct the separating balls) have been considered (from $p = 1$ to $p = 20$), although we only show the best results, which were obtained for $p = 15$. The number of iterations to obtain each ball is set equal to 50, and the solution based on the centroid (see Subsection 5.6.2) was taken as the initial solution.

Table 5.5 displays the confusion matrix for the test samples in the database with the first 10 classes. Each element (i, j) of this matrix represents the percentage of elements of the class i which has been assigned to the class j. The elements of the diagonal (in bold) represent the percentage of elements correctly labeled for each class. Then, one can observe that the class 4 (dinosaurs) is easily separable from the rest of classes, since all its elements have been correctly classified. However, some problems appear

		Assigned class									
		0	1	2	3	4	5	6	7	8	9
Real class	0	**67**	3	4	0	2	11	1	3	3	6
	1	2	**58**	7	3	1	8	1	0	17	3
	2	4	3	**75**	2	1	7	1	0	5	2
	3	1	3	10	**67**	9	2	0	1	1	6
	4	0	0	0	0	**100**	0	0	0	0	0
	5	10	4	2	0	0	**71**	0	5	8	0
	6	2	0	0	0	0	1	**94**	2	0	1
	7	3	1	0	0	0	14	0	**81**	1	0
	8	0	18	3	0	0	10	0	0	**69**	0
	9	5	4	1	2	5	4	0	3	5	**71**

Table 5.5: Confusion matrix for the 1000-Image database

Class	0	1	2	3	4	5	6	7	8	9
Test	67	58	75	67	100	71	94	81	69	71
Train	83.75	78.5	90.5	96.5	100	86.25	99.5	96.25	75.5	97.75

Table 5.6: Accuracy for the 1000-Image database

to distinguish between classes 0 and 5 (African people or villages and elephants), or especially between two kind of landscapes: images of beaches and images of mountains or glaciers (classes 1 and 8), where we obtain 17% of beaches misclassified as mountains or glaciers, and 18% in the other direction. Likewise, 14% of horses (class 7) are labeled as elephants (class 5).

Table 5.6 shows the accuracy for every class, that is, the percentage of elements of every class which has been correctly labeled into its class, in the training and the test samples. One can observe that the performance of the algorithm is quite good in most of the classes in the training sample, and, in general, a class which is easily separable from the rest in the training sample, continues being easily discriminated in the test sample. This is the case of class 4, with 100% of accuracy in both the training and the test samples. However, we can also find some classes, like class 3 (buses) with a much better accuracy in the training (96.5%) than in the test sample (only 67%).

Table 5.7 presents the classification accuracy for every class in the complete dataset (2000-Image database). The performance of our algorithm is good in the training sample in most of the classes (except for class 8), showing the power of our methodology to separate the bags of the different concepts. In the test sample, the accuracy is lower than for the 1000-Image database, although good results are obtained for separating classes such as number 4 and 6 (dinosaurs and flowers).

In Table 5.8, the accuracy for the test and training samples in the two databases are shown. One can observe that we obtain very good results for the training sample

Class	0	1	2	3	4	5	6	7	8	9
Test	52	58	69	67	97	55	88	78	42	67
Train	84.25	83.5	91	93.25	100	83.25	98.75	92	68.75	95.25
Class	10	11	12	13	14	15	16	17	18	19
Test	44	63	64	57	57	76	81	60	49	28
Train	86	82.5	96	92.25	92	92.5	99.25	95.5	84	74.75

Table 5.7: Accuracy for the 2000-Image database

	1000-Image database	2000-Image database
Test	75.3	62.6
Train	90.45	89.24

Table 5.8: Accuracy for the two databases

(even with the complete dataset), around 90%, and the results for the test sample are quite competitive.

Finally, we compare the results we have obtained, with the results obtained via other methods which have been used in this database: MILES algorithm [26], DD-SVM [27], MI-SVM [2] and k-means-SVM [32]. These other algorithms have also been tested over five test sets extracted at random from the database, but the technique is not cross validation (see [26]). Although our algorithm does not get to improve the best results obtained so far, our results are comparable with the solutions obtained for this database, and in fact, we get to improve the performance of other algorithms based on SVMs (like MI-SVM and k-means-SVM) in this multi-class problem in both datasets.

Algorithms	1000-Image database	2000-Image database
MILES	82.6	68.7
DD-SVM	81.5	67.5
MI-SVM	74.7	54.6
k-means-SVM	69.8	52.3
p-balls VNS algorithm	75.3	62.6

Table 5.9: Accuracy for different algorithms for the image database

5.9 Conclusions and extensions

In this chapter, a new tool for solving classification problems in Multiple Instance Learning has been described. The problem, which has an easy geometric interpretation, has been formulated as a nonlinear mixed integer optimization problem. The existence of finite optimal solution has been studied and necessary conditions for optimal solutions have been deduced.

These optimality conditions have been considered to develop the heuristic algorithm (a Variable Neighbourhood Search algorithm) to solve the model. The computational results for the classifier show that our tool is competitive, especially with separable databases.

The introduction of p balls in the classification rule for the benchmark datasets has improved remarkably the performance of the algorithm. The selection of the optimal value for p is a topic to be considered. In fact, a method to choose automatically the most suited value of p in each training sample during the cross validation process seems to be an interesting problem for further works.

The problem can be extended by considering different assumptions for building the classification rule, by changing the objective function of the problem, even by proposing a biobjective problem since some constraints may be relaxed.

The introduction of kernel structures in the problem and the extension to the Multi-Instance Regression problem, as described in [40, 94], are other topics which deserve further studies. Likewise, another interesting extension, which is addressed in the following chapter, is the application of this model to Location Theory (in particular, in location of semi-obnoxious facilities).

Chapter 6

An application to a semi-obnoxious location problem

Contents

6.1 Introduction

In this chapter, we adapt the classification rule described in Chapter 5 for the Multi-Instance Classification Problem to the area of Location Theory, in particular to a semi-obnoxious location problem.

For the last years, the location of semi-desirable facilities has been a widely studied topic by the researchers in location theory (see [12, 23, 25, 47, 84, 85, 86, 111]). A facility is said to be semi-desirable (or semi-obnoxious) when it gives service to certain customers in the neighborhood but, on the other hand, is felt as obnoxious to its environment. For instance, hospitals, airports or train stations are examples of semi-obnoxious facilities, since they are useful and necessary for the community, but they are a source of negative effects, such as noise, and therefore, they are considered as NIMBY (*not in my backyard*) facilities.

In our problem, a semi-obnoxious facility must be located in the plane and there are two different groups of customers to be considered. On the one hand, there exists a group of *attracting* points, whose demand must be satisfied by the facility which must be therefore as close as possible to all of them. On the other hand, there exists a set of *repelling* regions, which represent populated areas (whose shapes will be approximated via convex polygons) to be protected from the noxious effects coming from the facility, and hence, they must be as far as possible from the facility.

The problem of locating such a facility is approached via the construction of a ball separating the group of attracting points from the group of repelling regions, with maximum margin, in a similar way to the ball obtained for the classification problem in Chapter 5.

In Section 6.2, we introduce a formulation of this problem as a margin maximization model similar to Support Vector Machine methods. In Section 6.3, several structural properties are proved leading to a finite dominating set. This allows to obtain a polynomial solution method in Section 6.4, which will be tested on several artificial databases in Section 6.5. Finally, some concluding remarks are given in Section 6.6.

6.2 Modeling the problem

6.2.1 The basic aim

Consider G_+ and G_- two groups of elements in the Euclidean plane, where G_+ is a finite set of points $G_+ = \{x_1, \ldots, x_n\} \subset \mathbb{R}^2$, and G_- is a set of convex polygonal areas $G_- = \{S_1, \ldots, S_m\} \subset \mathbb{R}^2$ (with $n, m \geq 3$). The points of G_+ represent individual customers to be serviced by the facility, while the polygons represent areas to be protected

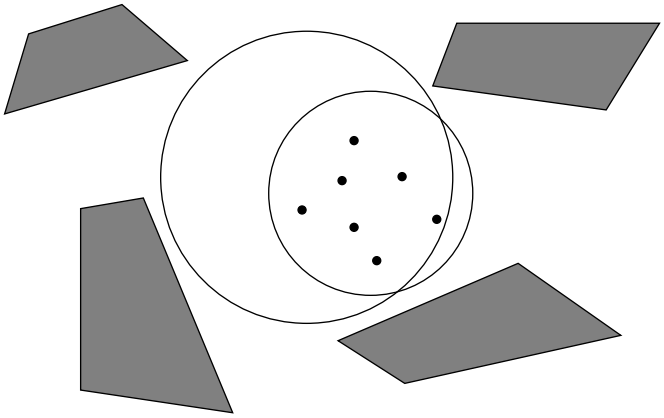

Figure 6.1: Two possible separating balls

from the inconveniences of the semi-obnoxious facility to be located. The points of G_+ are assumed not to be contained in any element of G_-. Also the polygons in G_- are assumed to have pairwise disjoint interiors. Note that this is not a restriction because any (possibly disconnected) polygonal region can be decomposed into a finite set G_- which satisfies our assumptions.

Our aim is to locate a single semi-obnoxious facility, $x_0 \in \mathbb{R}^2$, which is as near as possible to the points of G_+ (*attracting* elements) in order to receive a high-quality service, and far from the polygons of G_- (*repelling* elements).

The location of the facility is done through the construction of a ball $B(x_0, r)$, with $x_0 \in \mathbb{R}^2$ and $r \in \mathbb{R}_+$, such that every point of G_+ is strictly contained in the ball and every polygon of G_- lies outside the ball.

In Figure 6.1, an example of the problem is depicted. The black points represent the attracting points of G_+, whereas the grey-coloured areas represent the repelling elements of G_-. Our problem is to build a ball such that it contains all the points and it does not intersect the interior of any polygon.

Different solutions may exist separating the elements in G_+ and G_-. For instance, in Figure 6.1 two possible circles separating the two groups have been depicted. In order to single out one ball, we follow the strategy of maximizing a margin, similarly to the solution given for the problem of Chapter 5 (see Section 5.2). Following this strategy, the smallest circle in Figure 6.1 will be preferred as a solution.

6.2.2 The optimization problem

Given the elements of the two groups, G_+ and G_-, the following constraints must be satisfied, if possible,

$$dist^2(x_0, x_i) < r^2, \ \forall x_i \in G_+, \quad \text{i.e.,} \quad \|x_0 - x_i\|^2 < r^2, \ \forall x_i \in G_+, \tag{6.1}$$

$$dist^2(x_0, S_j) \geq r^2, \ \forall S_j \in G_-, \quad \text{i.e.,} \quad \min_{x \in S_j} \|x_0 - x\|^2 \geq r^2, \ \forall S_j \in G_-, \tag{6.2}$$

where $dist$ and $\|\cdot\|$ are the Euclidean distance and norm, respectively.

Constraints (6.1)-(6.2) can be rewritten as

$$\begin{aligned} r^2 - \|x_0 - x_i\|^2 > 0, \quad & \forall x_i \in G_+, \\ \min_{x \in S_j}(\|x_0 - x\|^2 - r^2) \geq 0, \quad & \forall S_j \in G_-, \end{aligned}$$

or equivalently,

$$\min_{x_i \in G_+} (r^2 - \|x_0 - x_i\|^2) > 0, \tag{6.3}$$

$$\min_{S_j \in G_-} \min_{x \in S_j} (\|x_0 - x\|^2 - r^2) \geq 0. \tag{6.4}$$

Following the strategy of maximizing the margin implies that we must maximize the minimum of the two positive amounts described in (6.3)-(6.4), that is, the optimization problem we want to solve is

$$\max_{x_0, r} \ \min \left\{ \min_{x_i \in G_+} (r^2 - \|x_0 - x_i\|^2), \min_{S_j \in G_-} \min_{x \in S_j} (\|x_0 - x\|^2 - r^2) \right\}. \tag{6.5}$$

Denote by Δ the margin, which is defined as the minimum between the two differences considered in Problem (6.5),

$$\Delta = \min \left\{ \min_{x_i \in G_+} (r^2 - \|x_0 - x_i\|^2), \min_{S_j \in G_-} \min_{x \in S_j} (\|x_0 - x\|^2 - r^2) \right\}. \tag{6.6}$$

Thus, our margin maximization problem can be written as

$$\begin{array}{ll} \max\limits_{x_0, r, \Delta} & \Delta \\ \text{s.t.} & \Delta \leq \min\limits_{x_i \in G_+} (r^2 - \|x_0 - x_i\|^2) \\ & \Delta \leq \min\limits_{S_j \in G_-} \min\limits_{x \in S_j} (\|x_0 - x\|^2 - r^2) \end{array} \tag{6.7}$$

or equivalently,

$$\begin{array}{ll} \max\limits_{x_0, r, \Delta} & \Delta \\ \text{s.t.} & \Delta \leq r^2 - \|x_0 - x_i\|^2, \ \forall x_i \in G_+ \\ & \Delta \leq \|x_0 - x\|^2 - r^2, \ \forall x \in S_j, \ \forall S_j \in G_-. \end{array} \tag{6.8}$$

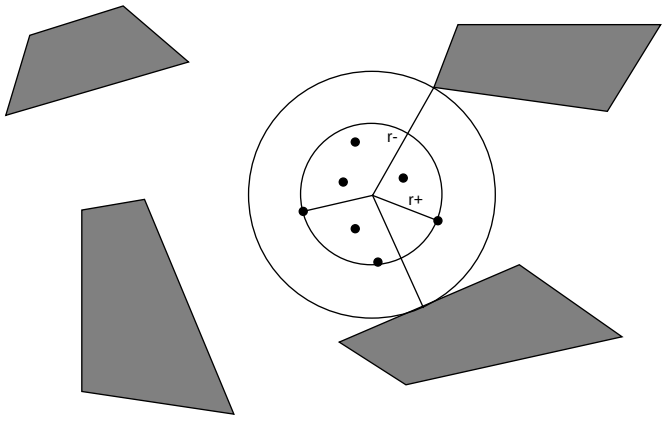

Figure 6.2: Construction of the two separating concentric balls with maximum margin

If we denote by $r_+^2 = r^2 - \Delta$ and $r_-^2 = r^2 + \Delta$, the objective function of Problem (6.8) changes into $\Delta = \dfrac{r_-^2 - r_+^2}{2}$, and the problem can be rewritten as

$$\begin{array}{ll} \max\limits_{x_0, r_+, r_-} & r_-^2 - r_+^2 \\ \text{s.t.} & \|x_0 - x_i\|^2 \leq r_+^2, \ \forall x_i \in G_+ \\ & \|x_0 - x\|^2 \geq r_-^2, \ \forall x \in S_j, \ \forall S_j \in G_- \\ & r_+, r_- \geq 0. \end{array} \tag{6.9}$$

In fact, this Problem (6.9) is more general since it also allows for situations with negative optimal values, which were unfeasible problems according to (6.7).

It will follow from Theorem 6.1 that Problem (6.9) can be reformulated by taking into account that, once the center x_0 is fixed, the optimal radii are fully defined and therefore, the objective depends on x_0 only, as follows

$$\begin{array}{ll} \max\limits_{x_0 \in \mathbb{R}^2} & f(x_0) \\ \text{s.t.} & f(x_0) = r_-^2(x_0) - r_+^2(x_0) \\ & r_+(x_0) = \max\limits_{x_i \in G_+} \|x_0 - x_i\| \\ & r_-(x_0) = \min\limits_{S_j \in G_-} \min\limits_{x \in S_j} \|x_0 - x\|. \end{array} \tag{6.10}$$

Therefore, our problem can be seen as that of obtaining two concentric balls $B(x_0, r_+)$, $B(x_0, r_-)$, where the ball $B(x_0, r_+)$ contains every point x_i belonging to G_+, the ball $B(x_0, r_-)$ does not contain strictly any points of the polygons of G_- and the difference between the squares of the radii is as large as possible, or geometrically, the area between the two circles is as large as possible. Figure 6.2 shows the graphical

idea of the problem. Our problem is thus related with the so-called *largest empty annulus problem* [37], in which an annulus of maximal area not containing points in its interior is sought, although in this problem, there is a unique set of points (instead of two groups) and no regions are considered.

In the following section, we derive some necessary optimality conditions. We will use (x_0, r_+, r_-) to either denote a finite feasible solution to Problem (6.9) or assume that $r_+ = r_+(x_0)$ and $r_- = r_-(x_0)$, as defined in Problem (6.10).

6.3 Necessary conditions for optimality

For deriving the necessary conditions for optimality of a feasible solution, the concept of active element will be necessary.

A point x_i from G_+ is an *active point* for the solution (x_0, r_+, r_-) iff the distance from x_i to the center x_0 is exactly r_+, that is, $dist(x_0, x_i) = \|x_0 - x_i\| = r_+$. Thus, the set of active points of G_+, which is denoted by $A_+(x_0)$, is formed by the points lying on the boundary of the ball $B(x_0, r_+)$.

In the same way, a polygon S_j from G_- is an *active polygon* for (x_0, r_+, r_-) iff the distance from S_j to x_0 is exactly r_-, that is, $dist(x_0, S_j) = \min_{x \in S_j} \|x_0 - x\| = r_-$. We denote by $A_-(x_0)$ the set of active polygons from G_-.

When x_0 is clear from the context, we will simply write A_+ and A_-.

In the proofs, the way to show that a feasible solution (x_0, r_+, r_-) is not optimal will be by finding another solution (x'_0, r'_+, r'_-) with a better value of the objective function or by exhibiting a direction of increase of f at x_0.

Theorem 6.1 *If (x_0, r_+, r_-) is an optimal solution, there exists at least one active element in each group G_+ and G_-, that is, the sets A_+ and A_- of active elements are non-empty.*

Proof.
Suppose that A_+ is an empty set. Since (x_0, r_+, r_-) is a feasible solution of Problem (6.9), all the points of the group G_+ must be (due to the emptiness of A_+) contained strictly in the ball $B(x_0, r_+)$, that is, $\|x_0 - x_i\| < r_+$, $\forall x_i \in G_+$.

Then, it is sufficient to take

$$r'_+ = r_+(x_0) = \max_{x_i \in G_+} \|x_0 - x_i\|,$$

which is strictly smaller than r_+, and we obtain (x_0, r'_+, r_-), a feasible solution improving strictly the value of the objective function.

On the other hand, suppose that A_- is empty. Due to the feasibility of (x_0, r_+, r_-), the distance from x_0 to every polygon of G_- is strictly greater than r_-, that is, $dist(x_0, S_j) > r_-$, $\forall S_j \in G_-$. Thus, it is sufficient to consider

$$r'_- = r_-(x_0) = \min_{S_j \in G_-} dist(x_0, S_j) = \min_{S_j \in G_-} \min_{x \in S_j} \|x_0 - x\|,$$

strictly greater than r_-, and the solution (x_0, r_+, r'_-) improves strictly the objective function.

In both cases, we conclude that the initial solution (x_0, r_+, r_-) cannot be optimal.

□

Theorem 6.2 *Let (x_0, r_+, r_-) be an optimal solution, one has that:*

1. *If $r_+ \leq r_-$, then there must exist at least two active polygons in G_-.*
2. *If $r_+ \geq r_-$, then there must exist at least two active points in G_+.*

Proof.
By Theorem 6.1, if (x_0, r_+, r_-) is an optimal solution, there must exist at least one active point a in G_+ and one active polygon S in G_-. Below, we obtain new conditions about the number of active elements in each case.

1. When $r_+ \leq r_-$, suppose there is only one polygon S in the set A_-. Let y be the projection of x_0 on S, i.e., the point in S such that $dist(x_0, S) = \min_{x \in S} dist(x_0, x) = dist(x_0, y)$ and consider the direction $p = x_0 - y$. Our aim is to prove that this vector p represents a direction of improvement for the objective function.

   If we move x_0 an amount $\epsilon > 0$, small enough (for not finding any new active element), in the direction $u = \frac{p}{\|p\|}$, we obtain that $x'_0 = x_0 + \epsilon u$ and $r'_- = r_- + \epsilon$.

   The other radius r'_+ must be measured as the maximum distance from x'_0 to the points belonging to $A_+(x_0)$.

   In case we obtain that $r'_+ \leq r_+$, because the new center is closer to all the points in $A_+(x_0)$, the radii r'_+ and r'_- will have decreased and increased respectively, and consequently the objective function will also have strictly improved.

   Otherwise, the radius r'_+ will be the distance from x'_0 to the point a of $A_+(x_0)$ which is now the furthest one (see Figure 6.3). Due to the triangle inequality on a, x_0 and x'_0, one has that $r'_+ \leq r_+ + \epsilon$, and the value of the objective function is strictly improved when $r_+ < r_-$, since

$$\begin{aligned} r'^2_- - r'^2_+ &\geq (r_- + \epsilon)^2 - (r_+ + \epsilon)^2 \\ &= r^2_- - r^2_+ + 2\epsilon(r_- - r_+) > r^2_- - r^2_+. \end{aligned}$$

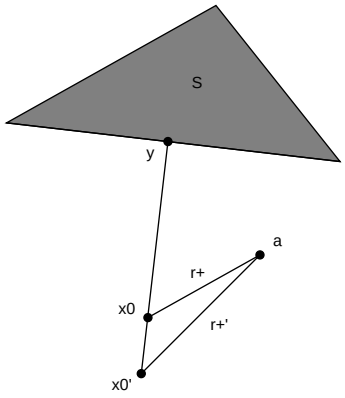

Figure 6.3: Proof of Theorem 6.2, case $r_+ \leq r_-$

In the case that $r_+ = r_-$, two cases can arise: either x_0', a and y are not collinear, or $a = y$, but this last is contrary to our assumption that no point of G_+ belongs to an element of G_-. Therefore, by strict triangle inequality $r'_+ < r_+ + \epsilon$,

$$r_-'^2 - r_+'^2 > (r_- + \epsilon)^2 - (r_+ + \epsilon)^2 = r_-^2 - r_+^2.$$

2. When $r_+ \geq r_-$, suppose there is only one active point a in A_+. Then, the vector $p = a - x_0$ will be proved to represent a direction of improvement.

   If x_0 is moved an amount $\epsilon > 0$, small enough for not having new active elements, in the direction $u = \frac{p}{\|p\|}$, we obtain that $r'_+ = r_+ - \epsilon$. The radius r'_- will be the minimum distance from $x'_0 = x_0 + \epsilon u$ to the polygons in $A_-(x_0)$. If we obtain that $r'_- \geq r_-$, because the new center is further from all the polygons candidates to become active, the two radii r'_+ and r'_- have improved and also the objective function. Otherwise, we denote by S one of the active polygons for the center x_0, which is now also the closest to the new center x'_0 (since there are not any new active polygons in G_-) and by y the projection of x_0 on S, i.e., the point of S such that $dist(x_0, S) = \min_{x \in S} dist(x_0, x) = dist(x_0, y)$, and the objective function can be expressed as follows,

   $$r_-^2 - r_+^2 \;\; = \;\; \|x_0 - y\|^2 - \|x_0 - a\|^2.$$

   Three different situations must be considered.

   - If y is a vertex of the polygon S and x_0 is strictly contained in the normal cone of S in y (denoted by $N_S(y)$), that is, x_0 satisfies that $(x_0 - y)^\top (y - s) > 0$, $\forall s \in S$, then, for $\epsilon > 0$ small enough, x'_0 will also be contained strictly in this normal cone, and $dist(x'_0, S) = \min_{x \in S} dist(x'_0, x) = dist(x'_0, y)$ (see Figure 6.4).

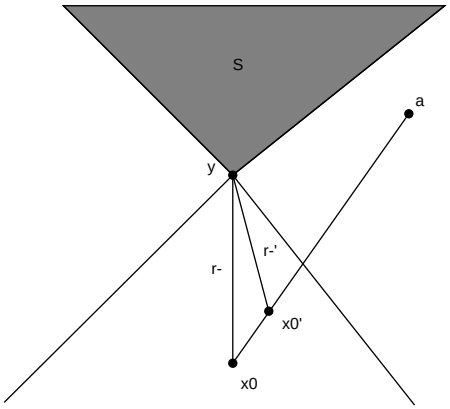

Figure 6.4: Proof of Theorem 6.2, case $r_+ \geq r_-$: The distance to the polygon is measured in the vertex

In that case, due to the triangle inequality, one has that $r_- \leq r'_- + \epsilon$ and consequently, $r'_- \geq r_- - \epsilon$, and the value of the objective function is improved in case $r_+ > r_-$, because

$$\begin{aligned} r'^2_- - r'^2_+ &\geq (r_- - \epsilon)^2 - (r_+ - \epsilon)^2 \\ &= r^2_- - r^2_+ + 2\epsilon(r_+ - r_-) > r^2_- - r^2_+ \end{aligned} \tag{6.11}$$

For $r_+ = r_-$, we know that $r_- < r'_- + \epsilon$, except for the case when x'_0, y and a are collinear. But this situation is not possible for $r_+ = r_-$, because it would mean that $a \in S$, which is not allowed by assumption. Therefore,

$$r'^2_- - r'^2_+ > (r_- - \epsilon)^2 - (r_+ - \epsilon)^2 = r^2_- - r^2_+ \tag{6.12}$$

- If the point y is on an edge of S, then, for an amount $\epsilon > 0$ small enough, to measure the distance from the new center x'_0 to the polygon S, we also have to find the point z (along the same edge of the polygon) which is the projection of x'_0 on S, i.e., the point satisfying $dist(x'_0, S) = \min_{x \in S} dist(x'_0, x) = dist(x'_0, z)$ (see Figure 6.5).
  In that case, one has that

$$r_- = \min_{x \in S} dist(x_0, x) \leq dist(x_0, z) \leq r'_- + \epsilon$$

  by using the definition of r_- and the triangle inequality on x_0, z and x'_0. Thus, we have that $r'_- \geq r_- - \epsilon$ and we can obtain again the same expression as in (6.11). Then, the objective function is improved for $r_+ > r_-$.
  And for $r_+ = r_-$, since $r'_- > r_- - \epsilon$ (except for the case in which x_0, y and a are collinear, and this situation cannot occur because it would mean that $a \in S$), we obtain again the expression (6.12), and the objective function is also improved.

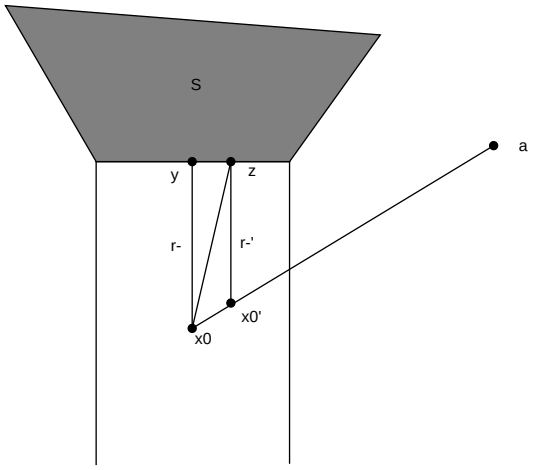

Figure 6.5: Proof of Theorem 6.2, case $r_+ \geq r_-$: The distance to the polygon is measured in the edge

- If y is a vertex of S and x_0 is on the boundary of the normal cone of S in y, then the projection of the new center x_0' on S will be either the same vertex y or a point z on an adjacent edge of S, depending on the position of a. Hence, one of the two arguments used previously applies to find a solution with a better value of the objective function.

□

Remark 6.1 *It can be proved that without the assumption that all points of G_+ lie outside the elements of G_-, Theorem 6.2 still holds in case of strict inequalities, but when $r_+ = r_-$, one may only conclude in the existence of an optimal solution with two active elements in G_+, and of an optimal solution (possibly different from the previous) with two active elements in G_-.*

Remark 6.2 *Note that only the first case in Theorem 6.2 is of interest to our original problem.*

Theorem 6.3 *If (x_0, r_+, r_-) is an optimal solution, then the intersection of the convex hulls of the two groups of active elements A_+ and A_- is a non-empty set.*

Proof.
Suppose $CH(A_+) \cap CH(A_-)$, the intersection of the convex hulls of the sets of active elements A_+ and A_-, is empty. In that case, a straight line h of equation $p^\top x = c$

can be found which strictly separates these two convex hulls, where p is a vector in $\mathbb{R}^2$ of unit length and $c \in \mathbb{R}$, such that the halfplane containing $CH(A_+)$ is defined by $\{p^\top x > c\}$. Consider the straight line $g : \{x = x_0 + \lambda p,\ \lambda \in \mathbb{R}\}$. We show now that the objective function will be improved by moving x_0 along this straight line a certain amount $\epsilon > 0$, small enough, which will terminate the proof.

Denote by S an active polygon from $A_-(x_0)$ which is the closest one to the new center $x_0' = x_0 + \epsilon p$, and by a a point from $A_+(x_0)$ which maximizes the distance from x_0' to $A_+(x_0)$. Denote by a_0 the orthogonal projection of a on g. Let y be the point of S such that $dist(x_0, S) = \min_{x \in S} dist(x_0, x) = dist(x_0, y)$ and y_0 its orthogonal projection to the straight line g. With this notation, the objective function can be expressed as follows,

$$\begin{aligned} r_-^2 - r_+^2 &= \|x_0 - y\|^2 - \|x_0 - a\|^2 \\ &= \|x_0 - y_0\|^2 + \|y_0 - y\|^2 - \|x_0 - a_0\|^2 - \|a_0 - a\|^2 \end{aligned}$$

If we move x_0 to x_0' along the straight line g, to measure the new radius r'_-, three different situations must be analyzed.

- In case the point y is a vertex of the polygon and x_0 is strictly contained in the normal cone of S in y, then, for an amount $\epsilon > 0$ small enough, the new center x_0' will also be contained strictly in the normal cone, and the distance from x_0' to S will continue being the distance from x_0' to the vertex y, that is, $dist(x_0', S) = \min_{x \in S} dist(x_0', x) = dist(x_0', y)$.

  Then, since $p = \frac{a_0 - y_0}{\|a_0 - y_0\|}$ (observe that $a_0 \neq y_0$, because g is orthogonal to the separating hyperplane and hence, a_0 and y_0 are also separated by the straight line h), the following calculation shows that the objective function improves,

$$\begin{aligned} r_-'^2 - r_+'^2 &= \|x_0 + \epsilon p - y_0\|^2 + \|y_0 - y\|^2 - \|x_0 + \epsilon p - a_0\|^2 - \|a_0 - a\|^2 \\ &= \|x_0 - y_0\|^2 + 2\epsilon (x_0 - y_0)^\top p + \|y_0 - y\|^2 \\ &\quad -\|x_0 - a_0\|^2 - 2\epsilon (x_0 - a_0)^\top p - \|a_0 - a\|^2 \\ &= r_-^2 - r_+^2 + 2\epsilon (a_0 - y_0)^\top \frac{a_0 - y_0}{\|a_0 - y_0\|} \\ &= r_-^2 - r_+^2 + 2\epsilon \|a_0 - y_0\| > r_-^2 - r_+^2 \end{aligned}$$

- In case the point y is on an edge of the polygon S, then, for an amount $\epsilon > 0$ small enough, to measure the distance from the new center x_0' to S, we also have to move along the same edge of the polygon to find the point z such that $dist(x_0', S) = \min_{x \in S} dist(x_0', x) = dist(x_0', z)$, the projection of x_0' on S (see Figure 6.6).

  Consider z_0 the orthogonal projection of z to the straight line g. Observe that $p = \frac{a_0 - z_0}{\|a_0 - z_0\|}$ ($a_0 \neq z_0$, because g is orthogonal to h and h separates a and z) and

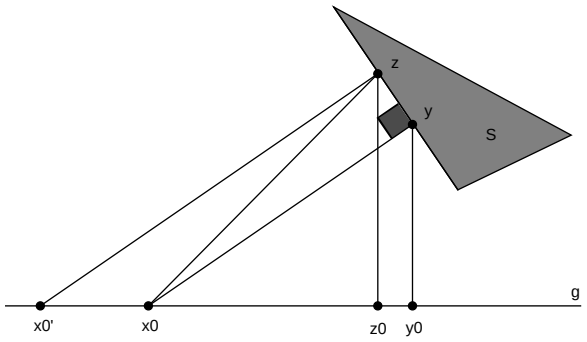

Figure 6.6: Proof of Theorem 6.3, second part

observe also that $x_0 - y$ and $z - y$ are orthogonal, because y is the projection of x_0 on the edge containing z. Therefore, by Pythagoras' Theorem, one has that

$$\begin{array}{rcl} \|x_0 - z_0\|^2 + \|z_0 - z\|^2 & = & \|x_0 - z\|^2 = \|x_0 - y\|^2 + \|y - z\|^2 \\ & \geq & \|x_0 - y\|^2 = \|x_0 - y_0\|^2 + \|y_0 - y\|^2 \end{array} \tag{6.13}$$

The objective function remains as follows,

$$\begin{array}{rcl} r_-'^2 - r_+'^2 & = & \|x_0 + \epsilon p - z_0\|^2 + \|z_0 - z\|^2 - \|x_0 + \epsilon p - a_0\|^2 - \|a_0 - a\|^2 \\ & = & \|x_0 - z_0\|^2 + 2\epsilon(x_0 - z_0)^\top p + \|z_0 - z\|^2 \\ & & -\|x_0 - a_0\|^2 - 2\epsilon(x_0 - a_0)^\top p - \|a_0 - a\|^2 \end{array}$$

And now, by using inequality (6.13), we obtain

$$\begin{array}{rcl} r_-'^2 - r_+'^2 & \geq & \|x_0 - y_0\|^2 + \|y_0 - y\|^2 \\ & & -\|x_0 - a_0\|^2 - \|a_0 - a\|^2 + 2\epsilon(a_0 - z_0)^\top p \\ & = & r_-^2 - r_+^2 + 2\epsilon\|a_0 - z_0\| > r_-^2 - r_+^2 \end{array}$$

- In case y is a vertex of S and x_0 is on the boundary of the normal cone of S in y, then the projection of x_0' on S will be either the vertex y or a point z on an edge of S, and, depending on the position of a, one of the two previous arguments applies to find a solution which improves the objective function.

□

Remark 6.3 *For the following theorem, we need the additional assumption that the data must be in general position. This means the exceptional situations described below do NOT appear. The aim of introducing this assumption is to avoid situations in which the associated solution has a slightly different behaviour to the general one:*

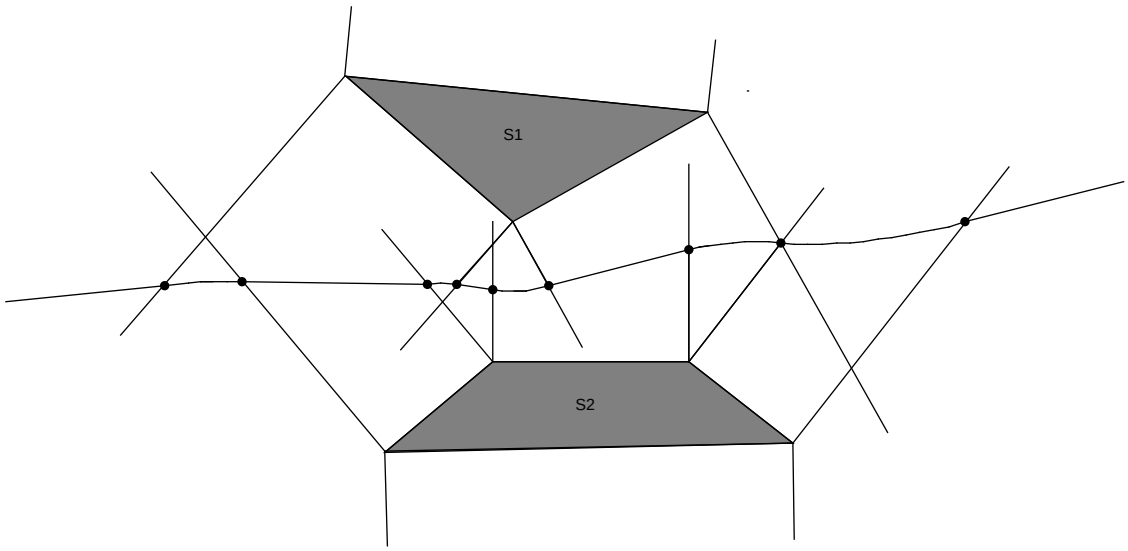

Figure 6.7: Bisector of two polygons S_1 and S_2. Breakpoints •

1. *One point of G_+ and two vertices of two polygons of G_- are collinear.*
2. *One point of G_+ and one vertex of a polygon of G_- define an orthogonal direction to an edge of a polygon of G_-.*
3. *Two points of G_+ and one vertex of a polygon of G_- are collinear.*
4. *Two points of G_+ define an orthogonal direction to an edge of a polygon of G_-.*

Likewise, the concept of bisector for two convex polygons and breakpoints will be necessary for the proof of Theorem 6.4 (see [33, 87] for a detailed description).

Definition 6.1 *The bisector of two convex polygons S_1 and S_2 is the locus of points $x \in \mathbb{R}^2$ satisfying that $dist(x, S_1) = dist(x, S_2)$. One has that this bisector is a continuous open curve consisting of linear segments and parabolic segments.*

The points at which two such segments meet will be called breakpoints (see Figure 6.7).

Theorem 6.4 *Under the assumption that the data are in general position, if (x_0, r_+, r_-) is an optimal solution, one of the following situations arises:*

1. *there exist at least four associated active elements ;*
2. *there exist at least three active elements, two polygons $S_1, S_2 \in A_-$ and one point $a \in A_+$, satisfying that y_1, a and y_2 are collinear, with y_i such that $dist(x_0, S_i) = \min_{x \in S_i} dist(x_0, x) = dist(x_0, y_i)$, $i = 1, 2$;*

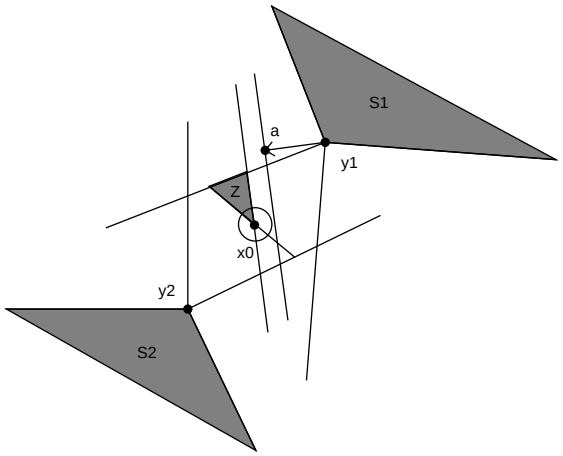

Figure 6.8: Distances from x_0 to the polygons are the distances to two vertices

3. *there exist at least three active elements, two polygons $S_1, S_2 \in A_-$ and one point $a \in A_+$, and x_0 is a breakpoint.*

Proof.
In the case in which the two radii are equal, we obtain directly the result of having four active elements associated, by Theorem 6.2. Below, we consider the remaining two cases.

1. When $r_+ < r_-$, by Theorems 6.1 and 6.2, we know that an optimal solution must have at least two distinct active polygons $S_1, S_2 \in A_-$, and one active point, $a \in A_+$. Suppose an optimal solution (x_0, r_+, r_-) has been obtained with only these three active elements.

   Since x_0 must be at the same distance from the two active polygons of A_-, it must be along their bisector which is composed of line segments and pieces of parabola. So x_0 is either a breakpoint or an 'inner point' of such a segment or piece.

   Then, in this last case, we must still show that a new better solution can be found, and the following different cases must be considered.

   - If there exist two vertices $y_1 \in S_1$ and $y_2 \in S_2$ which satisfy $dist(x_0, S_i) = \min_{x \in S_i} dist(x_0, x) = dist(x_0, y_i)$, $i = 1, 2$, x_0 lies on the mediatrix g between the vertices y_1 and y_2 (see Figure 6.8).

     Suppose that the active point $a \in A_+$ is nearer to S_1 than to S_2 (the other case is analogous by symmetry).

Define R the convex region determined by those points nearer to S_1 than to S_2 which are in the normal cone of S_1 at y_1, that is, $R = \{x : dist(x, y_1) \leq dist(x, y_2)\} \cap N_{S_1}(y_1)$. In this region, define the following function,

$$g(x) = \|x - y_1\|^2 - \|x - a\|^2 = 2x^\top(a - y_1) + C'. \tag{6.14}$$

where $C' = \|y_1\|^2 - \|a\|^2$. One has that $f(x) = g(x)$, $\forall x \in R$, with f the objective function of Problem (6.10), in particular, $f(x_0) = g(x_0)$.

In order to find a direction of improvement for the objective function in the neighbourhood of x_0, we study the directional derivatives of the objective function f at this point. Since the function g is differentiable in the region R, and $f \equiv g$ in R, we obtain that the gradient of the function at x_0 is $\nabla g(x_0) = 2(a - y_1)$, and the directional derivative along a vector v is

$$\nabla_v f(x_0) = \nabla_v g(x_0) = \nabla g(x_0)^\top \cdot v = 2(a - y_1)^\top v, \ \forall v = y - x_0, \ y \in R \tag{6.15}$$

Hence, to obtain a direction of improvement, it is sufficient to choose a vector v such that the scalar product $(a - y_1)^\top v$ is strictly larger than zero.

If we define the straight line orthogonal to the vector $(a - y_1)$ and containing the point x_0, that is, $g : (a - y_1)^\top(x - x_0) = 0$, and if we consider the region Z determined by those points in R which are also in the positive halfplane defined by the straight line g, that is, $Z = R \cap \{x : (a - y_1)^\top(x - x_0) > 0\}$, then the intersection $Z \cap B(x_0, \epsilon)$, with $\epsilon > 0$ small enough, is not empty, except for the case in which the straight line coincides with the mediatrix.

Then, we can find one point $z \in Z \cap B(x_0, \epsilon)$, and by moving the point x_0 in the direction $v = z - x_0$, the objective function is improved.

The case in which g coincides with the mediatrix is only possible if y_1, a, and y_2 are collinear, which is the exception number 1 in Remark 6.3. Anyway, in this exceptional case, if we move x_0 along the mediatrix, the value of the objective function remains constant (then, the solution is not unique).

- If there exist a vertex $y_1 \in S_1$ and a point y_2 lying on an edge of S_2 such that $dist(x_0, S_i) = \min_{x \in S_i} dist(x_0, x) = dist(x_0, y_i)$, $i = 1, 2$, x_0 lies on a parabolic piece of the bisector, this parabola being the bisector between the vertex y_1 and the edge of S_2 (see Figure 6.9).

  Suppose that the active point $a \in A_+$ is nearer to S_1 than to S_2 (for the other situation, see the reasoning described for the following case, with y_1 and y_2 lying on the edges of the polygons).

  Define $R = N_{S_1}(y_1) \cap \{x : dist(x, y_1) \leq dist(x, y_2)\}$ and the function g as in expression (6.14). One has that $f(x) = g(x)$, $\forall x \in R$, and hence, the expression (6.15) for the directional derivative of f at x_0 along any vector $v = y - x_0$, with $y \in R$, remains valid.

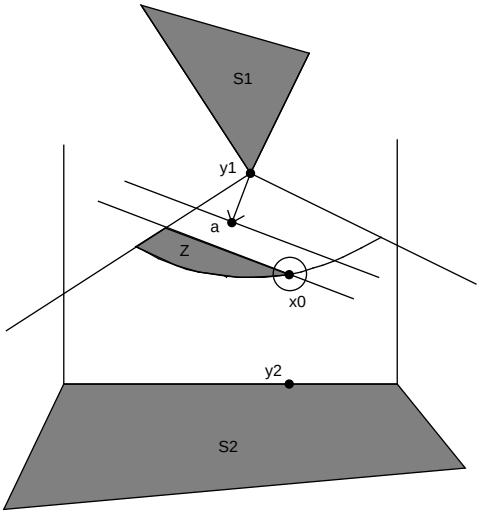

Figure 6.9: Distances from x_0 to the polygons are the distances to a vertex and to an edge

Hence, to obtain a direction of improvement, it is sufficient to choose a vector v such that the scalar product $(a-y_1)^\top v$ is strictly bigger than zero. And if we define $g : (a-y_1)^\top(x-x_0) = 0$ (the straight line containing x_0 and orthogonal to $(a-y_1)$) and $Z = R \cap \{x : (a-y_1)^\top(x-x_0) > 0\}$, the intersection $Z \cap B(x_0, \epsilon)$, with $\epsilon > 0$ small enough, is not empty, except for the case in which the straight line is tangent to the parabola. Then, a point $z \in Z \cap B(x_0, \epsilon)$ can be found, and by moving the point x_0 in the direction $v = z - x_0$, the objective function is improved.

The case in which g is tangent to the parabola is only possible when y_1, a, x_0 and y_2 are collinear, that is, when $a - y_1$ is orthogonal to the edge containing y_2, which is the exception number 2 of Remark 6.3 (in that case, a local optimum is found).

- If there exist two points y_1 and y_2, with y_i lying on an edge of S_i, such that $dist(x_0, S_i) = \min_{x \in S_i} dist(x_0, x) = dist(x_0, y_i)$, $i = 1, 2$, x_0 lies on the bisectrix of the angle formed by the two edges, which represents the bisector in this case (see Figure 6.10).

  Suppose that the active point $a \in A_+$ is nearer to S_1 than to S_2 (by symmetry, the other case is analogous).

  Denote by a_0 and y_0 the orthogonal projections of a and y_i on the bisectrix. Then, the objective function can be written in x_0 as

$$\begin{aligned} r_-^2 - r_+^2 &= \|x_0 - y_1\|^2 - \|x_0 - a\|^2 \\ &= \|x_0 - y_0\|^2 + \|y_0 - y_1\|^2 - \|x_0 - a_0\|^2 - \|a_0 - a\|^2. \end{aligned}$$

  The vector $p = a_0 - y_0$, if non-zero, will be a direction of improvement of

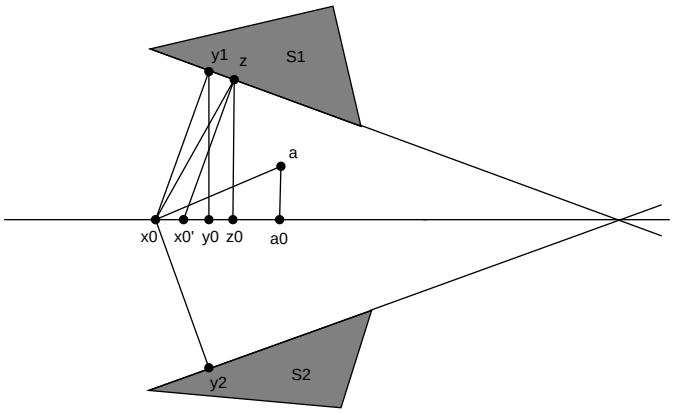

Figure 6.10: Distances from x_0 to the polygons are the distances to two edges

the objective function, for $\epsilon > 0$, small enough.

Let z be the orthogonal projection of the new center $x_0' = x_0 + \epsilon p$ on the edge, and z_0 its orthogonal projection on the bisectrix.

If we move x_0 along the direction p an amount $\epsilon > 0$, the new value of the objective function is

$$\begin{aligned} r_-'^2 - r_+'^2 &= \|x_0 + \epsilon p - z_0\|^2 + \|z_0 - z\|^2 - \|x_0 + \epsilon p - a_0\|^2 - \|a_0 - a\|^2 \\ &= \|x_0 - z_0\|^2 + 2\epsilon(x_0 - z_0)^\top p + \|z_0 - z\|^2 \\ &\quad -\|x_0 - a_0\|^2 - 2\epsilon(x_0 - a_0)^\top p - \|a_0 - a\|^2. \end{aligned}$$

By Pythagoras' Theorem, we obtain that

$$\begin{aligned} \|x_0 - z_0\|^2 + \|z_0 - z\|^2 &= \|x_0 - z\|^2 = \|x_0 - y_1\|^2 + \|y_1 - z\|^2 \\ &\geq \|x_0 - y_1\|^2 = \|x_0 - y_0\|^2 + \|y_0 - y_1\|^2 \end{aligned}$$

In fact, the inequality is strict, since $y_1 \neq z$. Then, one has that

$$\begin{aligned} r_-'^2 - r_+'^2 &= \|x_0 - z_0\|^2 + \|z_0 - z\|^2 \\ &\quad -\|x_0 - a_0\|^2 - \|a_0 - a\|^2 + 2\epsilon(a_0 - z_0)^\top p \\ &> \|x_0 - y_0\|^2 + \|y_0 - y_1\|^2 \\ &\quad -\|x_0 - a_0\|^2 - \|a_0 - a\|^2 + 2\epsilon(a_0 - z_0)^\top p \\ &= r_-^2 - r_+^2 + 2\epsilon(a_0 - z_0)^\top (a_0 - y_0) \geq r_-^2 - r_+^2 \end{aligned}$$

and the objective function has improved, since the vectors $(a_0 - y_0)$ and $(a_0 - z_0)$ are parallel and in the same sense, for $\epsilon > 0$ small enough.

In case $a_0 = y_0$, the objective function cannot be improved, thus we have obtained a local optimal solution. The result is the situation 2 of Theorem 6.4, that is, three active elements (two polygons S_1, S_2 and one point a) with the points y_1, a and y_2 being collinear.

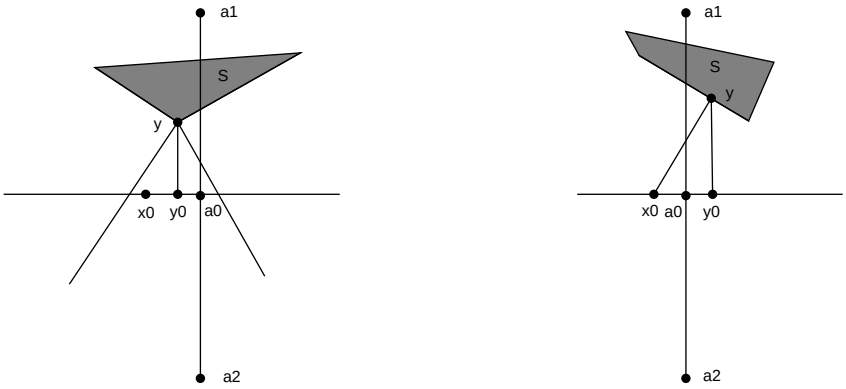

Figure 6.11: Two situations when $r_+ > r_-$

2. When $r_+ > r_-$, by Theorems 6.1 and 6.2, we know that an optimal solution of the problem must have at least two active points $a_1, a_2 \in A_+$, and one active polygon, $S \in A_-$. Suppose an optimal solution (x_0, r_+, r_-) with only these three active elements has been obtained. A new solution will be found with a better value of the objective function.

   Denote by g the mediatrix between the two active points of A_+, by y the point belonging to S such that $dist(x_0, S) = \min_{x \in S} dist(x_0, x) = dist(x_0, y)$, and by a_0 and y_0 the orthogonal projections of the points a_1 and y on the straight line g. We have to consider two situations (y is a vertex of the polygon or y lies on an edge of the polygon, see Figure 6.11), which are exactly the same as those described in the proof of Theorem 6.3.

   With a similar reasoning, we derive that a new feasible solution can be obtained which improves the objective function. The exception in this case is when a_1, y, x_0 and a_2 are collinear. This can happen because there are two points a_1, a_2 and a vertex y of a polygon S which are collinear (exception 3 in Remark 6.3) or because there are two points a_1, a_2 defining an orthogonal direction to an edge of a polygon S (exception 4 in Remark 6.3).

□

The concepts of nearest and farthest-point Voronoi diagrams (see [87, 99]) for a set of points or polygons will be necessary for the proof of Theorem 6.5.

Definition 6.2 *Given the set of points* $\{x_1, \ldots, x_n\}$ *and the set of polygons* $\{S_1, \ldots, S_m\}$, *the farthest-point (resp. nearest-polygon) Voronoi cell associated to* x_k *(resp.* S_l*) de-*

noted by V_k (resp. W_l) is defined as follows:

$$V_k = \bigcap_{i \in \{1,\dots,n\} \setminus \{k\}} \{x : dist(x, x_k) \geq dist(x, x_i)\}, \tag{6.16}$$

$$W_l = \bigcap_{j \in \{1,\dots,m\} \setminus \{l\}} \{x : dist(x, S_l) \leq dist(x, S_j)\}. \tag{6.17}$$

The sets $V = \bigcup_{k=1,\dots,n} V_k$ and $W = \bigcup_{l=1,\dots,m} W_l$ are called the farthest-point and the nearest-polygon Voronoi diagrams.

Theorem 6.5 *If the convex hulls of the two groups G_+ and G_- are disjoint, that is, $CH(G_+) \cap CH(G_-) = \emptyset$, then the solution is unbounded and the separating balls are transformed into straight lines.*

Proof.
Since $CH(G_+) \cap CH(G_-) = \emptyset$, a straight line $h : \{p^\top x = c\}$, with $p \in \mathbb{R}^2$ and $c \in \mathbb{R}$, separating the two convex hulls can be found, in the same way as done in the proof of Theorem 6.3. Let $l : \{p^\top x = c'\}$ be another straight line, parallel to h, such that every point $x_k \in G_+$ satisfies that $p^\top x_k > c$ and $p^\top x_k < c'$.

Construct the farthest-point and nearest-polygon Voronoi diagrams in the plane for G_+ and G_-, respectively, and the intersection of the two diagrams. Let V be a cell obtained as the intersection of the resulting diagram with the halfplane $\{p^\top x > c'\}$, such that there exists a point x_0 inside the cell satisfying that the semi-straight line $g : \{x = x_0 + \lambda p,\ \lambda \geq 0\}$ is completely included in the cell V.

Once x_0 is chosen, since it is inside a cell of the intersection of the two diagrams, the farthest point in G_+, say a, and the nearest polygon in G_-, say S, are known, that is, $a \in A_+$ and $S \in A_-$, and these two elements remain active for all the possible solutions in the cell, in particular for all the possible solutions in g. Then, with a similar reasoning to that done in the proof of Theorem 6.3, one has that if we move x_0 along g, for certain $\lambda' > 0$, the objective function increases linearly, thus, a new feasible solution $(x_0 + \lambda' p, r'_+, r'_-)$ with the same active elements can be found which is strictly better than the original one.

In fact, the larger the value of λ, the better the solution. Therefore, the solution is unbounded and, in that case, the concentric balls are transformed in two straight lines $\{p^\top x = b\}$ and $\{p^\top x = d\}$, with $b > d$, and such that the closed halfplane $\{p^\top x \geq b\}$ contains $CH(G_+)$ whereas $\{p^\top x \leq d\}$ contains $CH(G_-)$.

□

6.4 An algorithm to build the set of optimal solutions

With the necessary optimality conditions studied in Section 6.3, a finite dominating set of solutions has been obtained. A method to obtain an optimal solution is to perform a complete enumeration of all the candidate solutions, as is described below.

We are going to study all the local optimal solutions, and we will compute the value of the objective value for those points, and the one with the biggest value will be the global optimal solution. According to Theorems 6.1, 6.2 and 6.4, there must exist at least one active element in each set (A_+ and A_-), there must exist at least two active elements in the set associated to the biggest ball (that is, if $r_+ > r_-$, there will exist at least two active points in A_+, and if $r_- > r_+$, there will be at least two active polygons), and one of the situations described in Theorem 6.4 must be reached. That way, the finite dominating set of solutions will be formed by points x_0 whose configuration of associated active elements belongs to one of the following options:

1. three active polygons S_1, S_2, S_3 and one active point a (in this case, $r_- > r_+$);
2. two active polygons S_1, S_2 and two active points a_1, a_2 (no condition on the radii);
3. two active polygons S_1, S_2, one active point a and x_0 is a breakpoint of the bisector defined by S_1 and S_2 (in this case, $r_- > r_+$);
4. two active polygons S_1, S_2, one active point a and x_0 satisfies that y_1, y_2 and a are collinear, with y_i such that $dist(x_0, S_i) = \min_{x \in S_i} dist(x_0, x) = dist(x_0, y_i)$, $i = 1, 2$ (in this case, $r_- > r_+$);
5. three active points a_1, a_2, a_3, and one active polygon S (in this case, $r_+ > r_-$).

In the algorithm, to describe all the candidates, we will consider all the possible configurations and we will compute the solution x_0 as the intersection of the corresponding bisectors of the sets A_+ and A_-. Since the bisector of two polygons consists of segments (for two vertices, the bisector is their mediatrix, and for two edges, the bisector is the bisectrix) and pieces of parabola (for one vertex and one edge), we will study each vertex and edge of a polygon as different active elements in the algorithm.

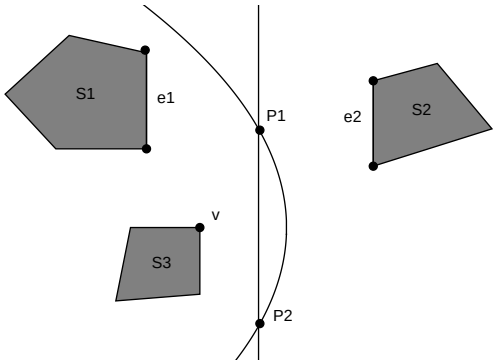

Figure 6.12: Computing a solution as the intersection of a bisectrix and a parabola

6.4.1 Case 1: card(A_+)=1 and card(A_-)=3

Let S_1, S_2 and S_3 be the three active polygons. As has been said before, every vertex and every edge of a polygon is studied as a possible active element. For a polygon S, considering a vertex v as the active element will mean that the closest point of S to the solution x_0 is v. Analogously, considering an edge e as the active element will mean that the point of S which is the closest one to x_0 lies on this edge e (not being one of the two vertices defining e).

Then, x_0 will be computed by following a different strategy depending on the number of active vertices and edges:

- Three vertices: x_0 is the circumcenter of the triangle defined by these three points (equivalently, x_0 is the intersection of the mediatrices for each pair of points).
- Two vertices and one edge: x_0 is the intersection of the mediatrix of the vertices and the parabola of one vertex and one edge.
- One vertex and two edges: x_0 is the intersection of the bisectrix of the two edges and the parabola of one vertex and one edge.
- Three edges: x_0 is the intersection of two bisectrices.

Once x_0 is computed (in some cases, more than one solution can be obtained), next step is to check if this solution is feasible, that is, given the three active elements, we must check if x_0 belongs to the intersection of the normal cones of the polygon S_i at the vertex v_i or the edge e_i respectively, for $i = 1, 2, 3$.

An example of this situation can be seen in Figures 6.12 and 6.13. In Figure 6.12, there are two active edges e_1 and e_2, and one active vertex v (belonging, respectively, to the active polygons S_1, S_2 and S_3). The bisectrix for the two edges is computed, and as well the parabola which represents the bisector of the edge e_2 and the vertex

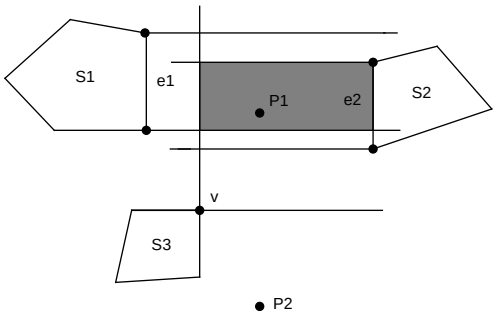

Figure 6.13: Checking the feasibility of the two points

v. There exist two points (P_1 and P_2) as the result of intersecting the bisectrix and the parabola. In Figure 6.13, we check the feasibility of these two possible solutions, and P_1 is accepted as a solution, because it belongs to the intersection of the normal cones of the three active elements (the shadowed rectangle in the picture) whereas P_2 is outside that rectangle.

If we obtain a solution x_0 with this combination of active elements, we define r_- as the distance from x_0 to any of these active elements. Observe that we must also check that the distance from x_0 to these active polygons S_i, $i = 1, 2, 3$, coincides with the distance to the active vertices or edges which have been considered, that is, the closest points from the polygons to x_0 must be the selected active vertices or must lie on the selected active edges (otherwise, the solution is not feasible).

Afterwards, we compute the distance from x_0 to the rest of polygons of G_-. If the minimum of these distances is bigger than or equal to r_- (if this minimum was smaller, the polygons S_1, S_2 and S_3 could not belong to A_-), we compute r_+ as the maximum distance from x_0 to the points of G_+, and the point a whose distance to x_0 is r_+ will be the fourth active element.

In this case, r_+ must be smaller than r_- to have the guarantee of having obtained a local optimal solution (else, according to Theorem 6.2, a better solution can be found in a neighbourhood of x_0).

6.4.2 Case 2: card(A_+)=2 and card(A_-)=2

Let a_1 and a_2 be the active points. Let S_1 and S_2 be the active polygons (in this case, we choose directly from the beginning the four active elements). We compute the mediatrix of the two active points, and we compute the bisector of the two active elements in the polygons (it will be a mediatrix if we have two active vertices, a bisectrix if there are two active edges, or a parabola if there are a vertex and an edge). The intersection of the mediatrix and the bisector is computed and we check the feasibility of this solution as done in the previous case (that is, we check if the solution x_0 belongs to the intersection of the normal cones of the polygons at the corresponding vertex or edge, and we also check that each selected vertex or edge is really active, in the sense that it is or it contains the closest point from the corresponding polygon to x_0).

Once a solution x_0 is obtained, we compute r_+ as the distance from x_0 to one of the active points and r_- as the distance to one of the active polygons. Then, we compute the maximum distance from x_0 to the rest of points of G_+ (x_0 is a candidate if this maximum distance is smaller than or equal to r_+) and the minimum distance from x_0 to the rest of polygons of G_- (x_0 is candidate to optimal solution if this minimum distance is bigger than or equal to r_-).

6.4.3 Case 3: card(A_+)=2, card(A_-)=1 and x_0 is a breakpoint

Let S_1 and S_2 be the two active polygons. In this case, for the first polygon, we can always consider as active elements in the algorithm only the vertices, since a breakpoint is built as the intersection of the bisector of two polygons with the boundary of the normal cone of one of the polygons at some of its vertices (see Figure 6.7). Then, the option of having two active edges can be ruled out (otherwise, each breakpoint would be studied twice).

Given one active vertex of S_1 and one active element of S_2 (a vertex or an edge), we compute the corresponding bisector (a mediatrix or a parabola, respectively) and we compute the intersection of this bisector with the intersection of the boundaries of the normal cones of the polygons at their active elements. That way, we obtain one (or several) breakpoint and we compute r_- as the distance from x_0 to the active polygons (if the selected vertices or edges are really active elements for the corresponding polygons).

Then, we compute the distances from x_0 to the rest of the polygons, and the minimum of these distances must be bigger than or equal to r_- (otherwise, x_0 is not a candidate optimal solution). We compute r_+ as the maximum distance from x_0 to the points in G_+ (r_+ must be smaller than r_-, otherwise, we could find a better solution in a neighbourhood of x_0).

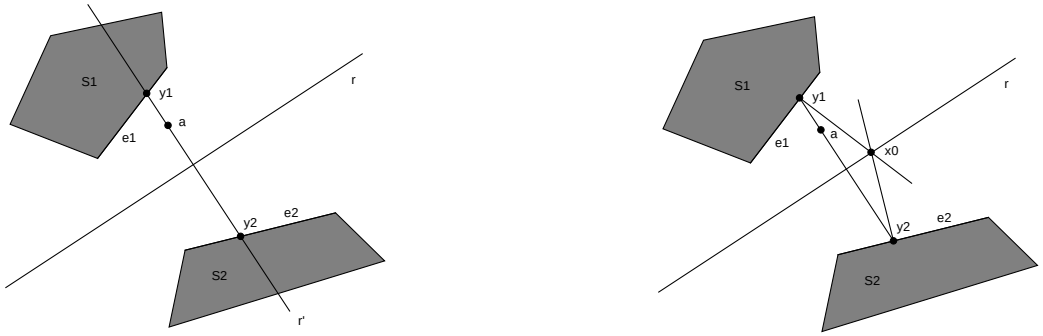

Figure 6.14: Case 5. Left: Constructing y_1 and y_2. Right: Constructing x_0

6.4.4 Case 4: card(A_+)=2, card(A_-)=1 and y_1, y_2 and a are collinear

Let a be the active point. Let S_1 and S_2 be the two active polygons. In this case, we only consider the edges of the polygons as possible active elements, since the condition we impose is that a, y_1 and y_2 are collinear, with y_i such that $dist(x_0, S_i) = \min_{x \in S_i} dist(x_0, x) = dist(x_0, y_i)$, $i = 1, 2$, and y_i not being vertices. The case of y_i being vertices cannot happen with data in general position (see Remark 6.3).

Given a and the edges e_1 and e_2, we study if there can exist two points y_1 and y_2 lying on the edges, such that the condition of collinearity is satisfied. If this is possible, we compute the bisectrix g of the two edges and the orthogonal straight line g' to the bisectrix containing the point a (see Figure 6.14, left). Let y_1 and y_2 be the intersection of g' with e_1 and e_2, respectively. Then, x_0 will be built as the intersection of the bisectrix with the orthogonal straight line to e_1 containing y_1 (see Figure 6.14, right). Symmetrically, we can do the same with e_2.

Once x_0 is built, we follow the same reasoning to build the radii as in case 2.

6.4.5 Case 5: card(A_+)=3 and card(A_-)=1

Let a_1, a_2, a_3 be the three active points, we compute x_0 as their circumcenter (equivalently, x_0 is the intersection of the mediatrices between these points), and $r_+ = dist(x_0, a_i)$, for any $i = 1, 2, 3$.

Now, we compute the distance from x_0 to the rest of points of G_+. If the maximum of these distances is smaller than or equal to r_+ (if it is bigger than r_+, the points a_i, $i = 1, 2, 3$, cannot be active), we compute r_- as the minimum distance from x_0 to the polygons of G_-. The polygon S whose distance to x_0 is equal to r_- will be the fourth active element.

Finally, r_+ must be bigger than r_- to assure that we have a local optimal solution (otherwise, a better solution can be found in a neighbourhood of x_0, according to Theorem 6.2). This implies that the value of the objective function for a candidate solution with this configuration of active elements will be negative. Hence, if we have already found a candidate solution with positive value, we do not need to compute any candidate of this type, since it cannot be a global optimum.

6.4.6 Cardinality of the set of candidates

Let us study now the size of the set of candidate points obtained this way. Denote by n the number of points in G_+, by m the number of polygons in G_- and by k the number of vertices of each polygon (in case of having polygons with different number of vertices, k would be the maximum number of vertices for these polygons).

For the candidate solutions of type 1, we need to study all the possible combinations of three polygons (and all the possible combinations of vertices and edges, which are different active elements). This yields a set of $\mathcal{O}(k^3m^3)$ points.

For the candidates of type 2, we need to select two polygons (and every possible combination of active elements, vertices and edges, of these two polygons) and two vertices. We have then $\mathcal{O}(n^2k^2m^2)$ points.

Two polygons are needed to build each candidate of type 3. We have $\mathcal{O}(k^2m^2)$ such points. For the candidates of type 4, we need to consider all possible combinations of edges of two polygons and one point, yielding $\mathcal{O}(nk^2m^2)$. Finally, if we need to compute the candidates of type 5, we need to study all the combinations of three active points, and we have $\mathcal{O}(n^3)$ points.

The overall cardinality is then $\mathcal{O}((n+km)^3 + n^2k^2m^2)$.

6.5 Computational experiment

The algorithm described in Section 6.4 to compute an optimal solution of our problem via complete enumeration of all the possible candidates has been implemented by using Matlab 6.5 on a computer with Pentium IV CPU 3.06 GHz.

Different numerical tests have been performed with artificial databases, built at random.

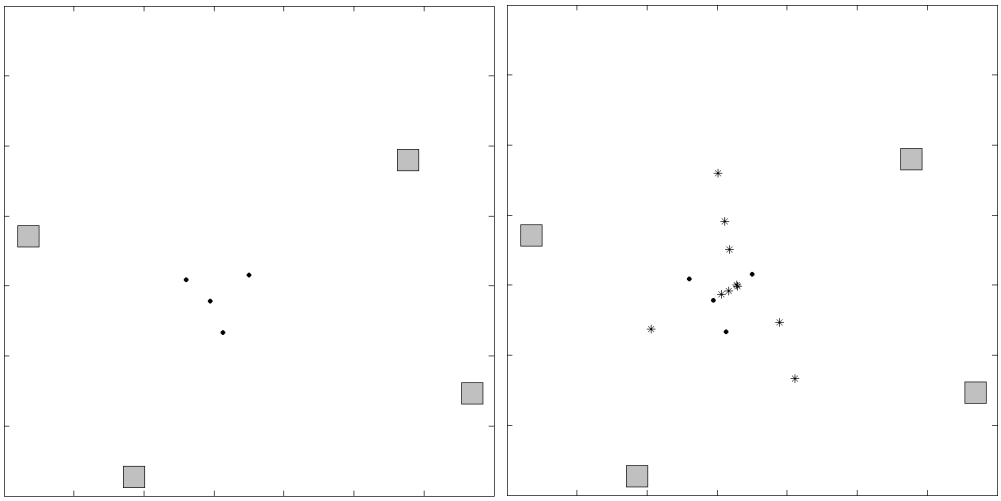

Figure 6.15: Left: Initial scenario. Rigth: Set of candidate optimal solutions

6.5.1 Small dataset: Comparing areas for all the candidates

The first example is a small dataset (4 points and 4 squares) to show the different types of candidate solutions that one can have in a problem. We have generated 4 points for the group G_+ coming from a uniform distribution (in particular, we have taken the distribution $U(-5,5)$), and other 4 points also coming from a uniform distribution, $U(-20,20)$, as the center of the squares (all the squares with the same area) which are the polygons for the group G_-. Our aim is to locate a single semi-obnoxious facility in a point $x_0 \in \mathbb{R}^2$, or equivalently, to compute two concentric balls such that $B(x_0, r_+)$ contains all the points and $B(x_0, r_-)$ does not intersect any squares. Figure 6.15 (left) shows a picture of the artificial database.

All the candidate optimal solutions have been computed via the method described in Section 6.4, by taking into account all the possible combinations of active elements. Figure 6.15 (right) shows this set of candidate locations, represented via stars.

In Figure 6.16, we show the two candidates with a configuration of type 1 (according to the previous section), that is, there are three active polygons (squares) and one active point. In the picture, the active squares are the black ones, while the active point is inside a small circle. These active elements (points and squares) lie on the boundary of the balls $B(x_0, r_+)$ and $B(x_0, r_-)$, respectively, where x_0 is represented via a star. Maximizing the objective function is equivalent to maximizing the area of the annulus defined by the boundaries of the two balls.

In Figure 6.17, the three candidate solutions have two active points and two squares. In Figure 6.18, we show five candidates with two active squares and one active element, and x_0, the location of the facility, is a breakpoint of the bisector defined by the two

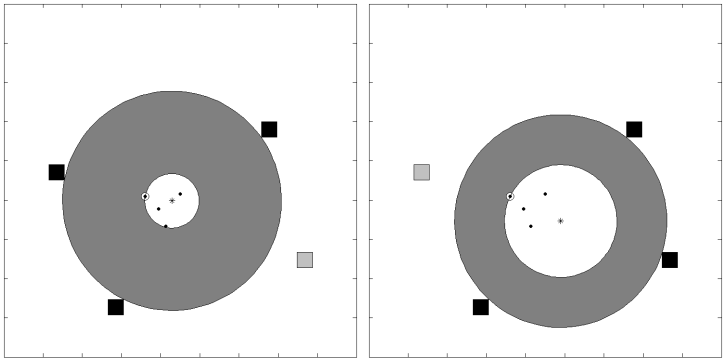

Figure 6.16: Candidates type 1. Area of the annulus: 183.27 and 132.32, respectively

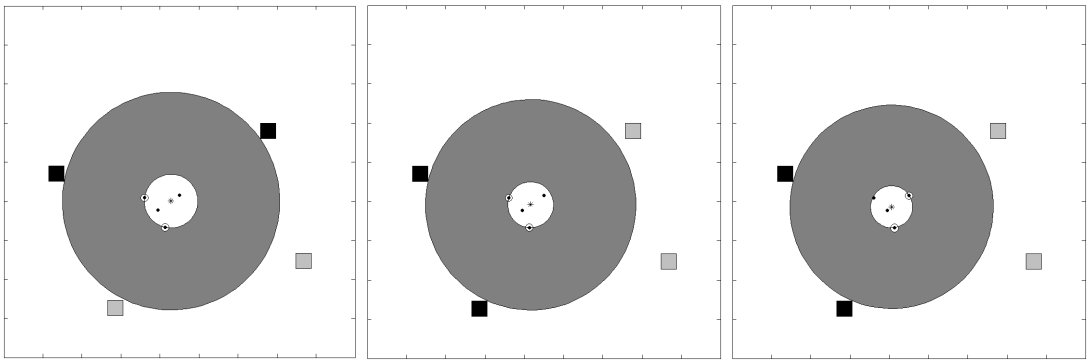

Figure 6.17: Candidates type 2. Area of the annulus: 182.32, 171.45 and 160.97, respectively

active squares. Observe that, although the active elements are the same for the three first pictures with this configuration, the solutions are different because the centers of the balls are different breakpoints of the same bisector. Due to the definition of breakpoint, one of the active squares in this kind of solutions has a vertex as the active element, but the adjacent edge touches tangently the ball $B(x_0, r_-)$. Hence, one can say that the two elements (the vertex and the edge) can be considered as active.

In this case, there are no candidate solutions with a configuration of type 4 or 5.

If we compare the ten areas (that is, the values of the objective function), we obtain that the first picture in Figure 6.16 is the optimal solution of our problem.

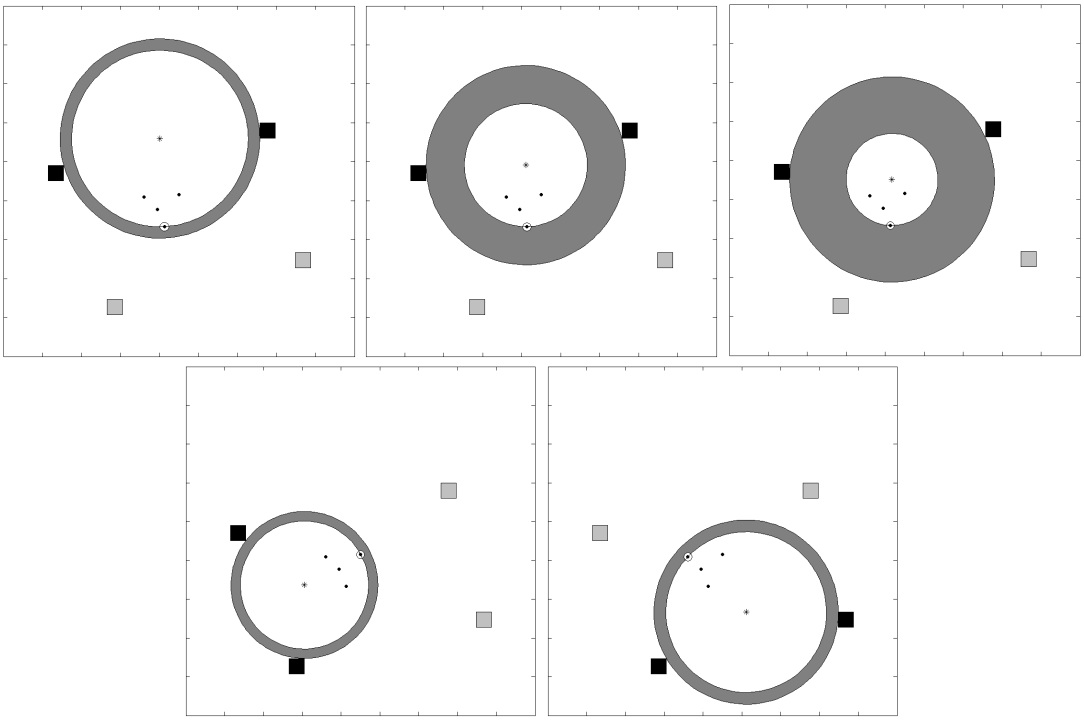

Figure 6.18: Candidates type 3. Area of the annulus: 35.648, 101.54, 138.16, 21.53 and 33.932, respectively

6.5.2 Other random datasets

Other larger databases have been generated to run the algorithm. In the next one, we have generated at random 50 points for the group G_+ and 20 points as the centers of the squares of G_- (the area for every square is the same), coming from two uniform distributions. Figure 6.19 shows a picture of the dataset.

By means of the method described in Section 6.4, all the candidate optimal solutions have been studied. Figure 6.20 shows two pictures, with different zoom levels, of all the candidate locations we have obtained, represented via stars. The stars which are far from the set of squares and points represent local optima with a negative value of the objective function. These solutions have at least two active points associated (configurations of type 2 and 5). In practice, if the dataset is spherically separable (in the sense that there exists a sphere separating the two sets of elements), the global optimum will not have a negative value of the objective function, but the formulation of our problem allows this kind of solutions as local optima.

Figure 6.21 shows the optimal solution for this dataset. The solution x_0, represented via a star, has two active points associated (those with a small circle around) and two active squares (in both of them, the point lying on the boundary of the ball is a vertex).

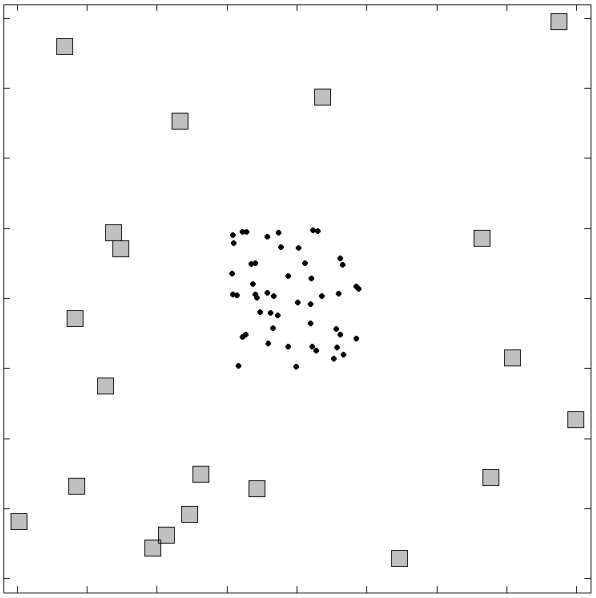

Figure 6.19: Initial scenario (50 points and 20 squares)

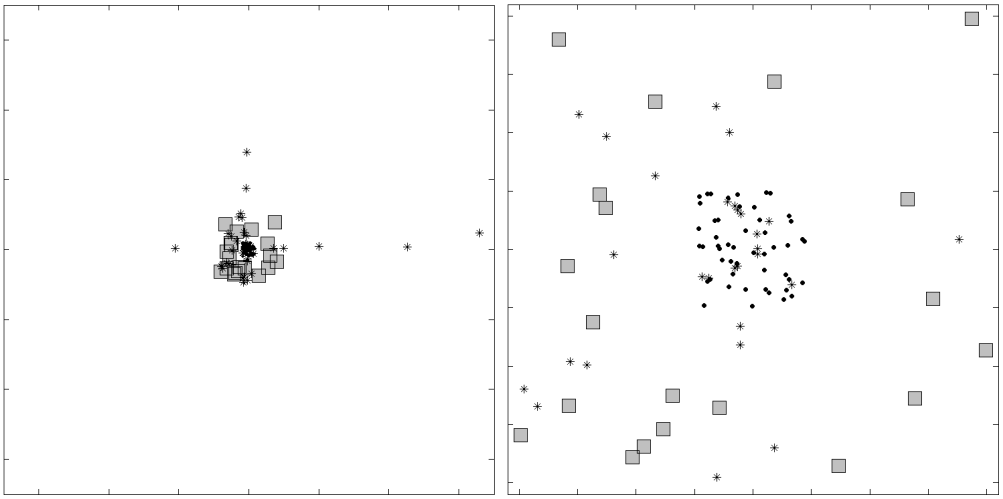

Figure 6.20: Candidates to optimal solution

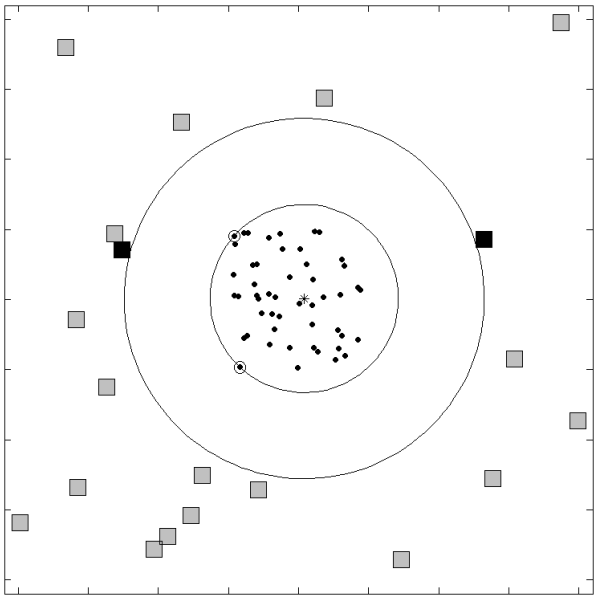

Figure 6.21: Optimal solution

The value of the objective function in this case (the area of the annulus) is 481.29.

Finally, we show a bigger random database, with 100 points generated via a uniform distribution and 50 squares (with a smaller area than in the previous cases). The initial scenario can be observed in Figure 6.22.

In Figure 6.23, one can observe the set of candidate optimal solutions (two different zoom levels). In this case, we have a lot of solutions with negative value of the objective function.

However, the database is spherically separable, hence, we have some solutions with a positive value of the objective function. All these candidates with positive value are depicted in Figure 6.24. Finally, the optimal location of the facility is depicted in Figure 6.25. The two balls also appear in the picture. The solution has four active elements associated, two active points and two active polygons. The ball of radius r_- touches these two squares on one vertex of each square.

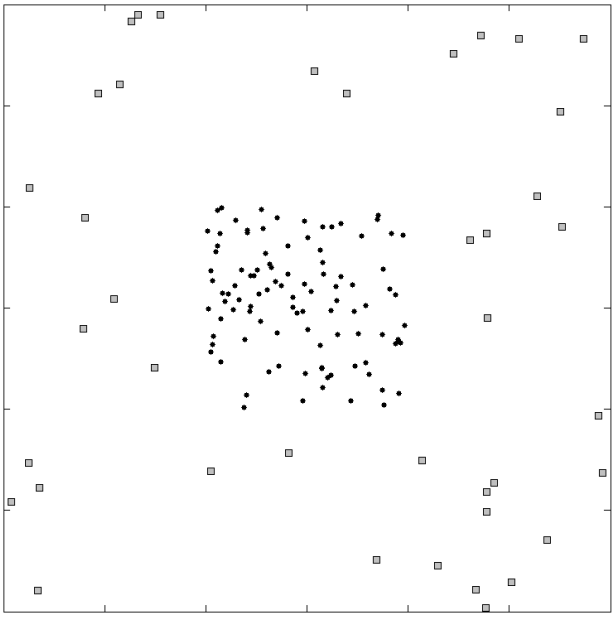

Figure 6.22: Initial scenario (100 points and 40 squares)

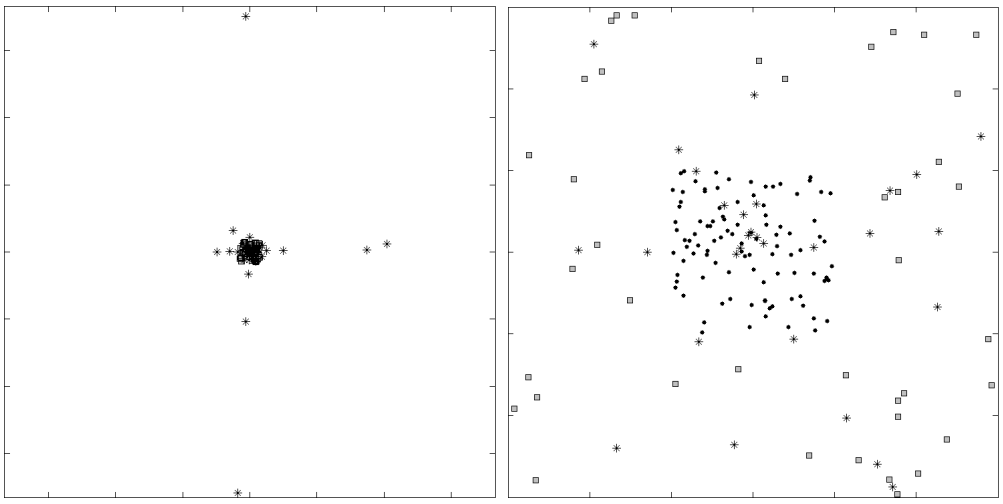

Figure 6.23: Candidates to be optimal solution

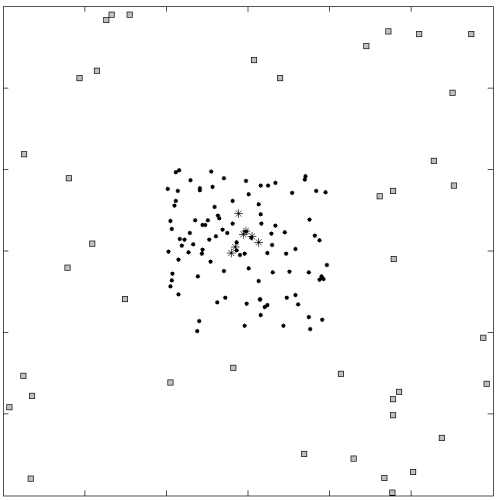

Figure 6.24: Candidates with positive value of the objective function

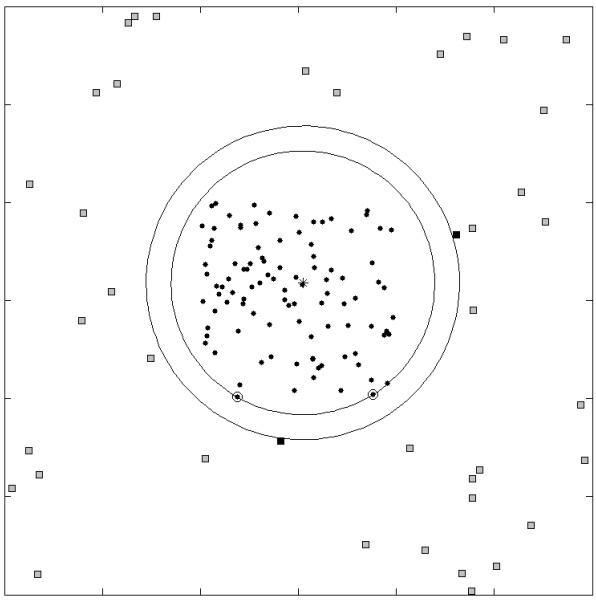

Figure 6.25: Optimal solution

6.6 Conclusions and extensions

In this chapter, the location of a single semi-obnoxious facility in the Euclidean plane with repelling areas has been solved. The idea of maximizing a margin, as done in the classification problems in Chapters 2 and 5, has been introduced to define the concept of solution.

The problem has been formulated via a nonlinear continuous optimization problem and necessary conditions for optimality have been deduced. These conditions state that every candidate solution must have at least four active elements (except for some especial cases), two of them belonging to the group whose associated ball is bigger and one of them belonging to the other group. Likewise, other conditions have been obtained by studying the intersection of the convex hulls of the sets of active elements and the sets of groups, respectively.

With these necessary conditions, it is proved that a finite dominating set of solutions can be built in order to obtain an optimal solution. This dominating set of solutions has been constructed algorithmically. This algorithm has been implemented and some numerical results have been given.

The concept of solution for this problem can be extended by considering other types of balls (such as ellipsoids, for example) and the problem can also be extended to higher dimensions.

For higher dimensions, heuristics techniques must be used to obtain a solution, and as well for large databases, if we want to decrease the CPU running time for obtaining a solution. A possibility would be to use metaheuristics, such as VNS, [54, 81], for which a neighbourhood structure can be defined by following a similar scheme to that used in the algorithm of Chapter 5.

Bibliography

[1] A. A. Afifi and R. M. Elashoff. Missing Observations in Multivariate Statistics: II. Point Estimation in Simple Linear Regression. *Journal of the American Statistical Association*, 62(317):10–29, (1967).

[2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support Vector Machines for Multiple-Instance Learning. In *Proc. Advances in Neural Information Processing Systems*, volume 15, pages 561–568, (2002).

[3] C. Angulo, D. Anguita, and L. González. Interval Discriminant Analysis using Support Vector Machines. In *Proc. of the European Symposium on Artificial Neural Networks*, pages 223–228, (2007).

[4] A. Asuncion and D. J. Newman. UCI Machine Learning Repository. http://www.ics.uci.edu/∼mlearn/MLRepository.html, (2007).

[5] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley and Sons, New York, (1993).

[6] A. Ben-Tal and A. Nemirovski. Robust Convex Optimization. *Mathematics of Operations Research*, 23(4):769–805, (1996).

[7] A. Ben-Tal and A. Nemirovski. Robust Solutions of Uncertain Linear Programs. *Operations Research Letters*, 25:1–13, (1998).

[8] S. Bhattacharyya, L. Grate, S. Mian, L. El Ghaoui, and M. Jordan. Robust Sparse Hyperplane Classifiers: Application to Uncertain Molecular Profiling Data. *Journal of Computational Biology*, 11(6):1073–1089, (2004).

[9] L. Billard and E. Diday. Regression Analysis for Interval-valued Data. In H. A. L. Kiers, J-P. Rasson, P. J. F. Groenen, and M. Schader, editors, *Data Analysis, Classification and Related Methods*, pages 369–374. Springer-Verlag, Berlin, (2000).

[10] L. Billard and E. Diday. Symbolic Regression Analysis. In K. Jajuga, A. Sokolowski, and H-H. Bock, editors, *Classification, Clustering and Data Analysis*, pages 281–288. Springer-Verlag, Berlin, (2002).

[11] L. Billard and E. Diday. From the Statistics of Data to the Statistics of Knowledge: Symbolic Data Analysis. *Journal of the American Statistical Association*, 98(462):470–487, (2003).

[12] R. Blanquero and E. Carrizosa. A D.C. Biobjective Location Model. *Journal of Global Optimization*, 23:139–154, (2002).

[13] H-H. Bock and E. Diday, editors. *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*. Springer-Verlag, Berlin, (2000).

[14] R. C. Bunescu and R. J. Mooney. Multiple Instance Learning for Sparse Positive Bags. In *Proc. 24th International Conference on Machine Learning*, pages 105–112, (2007).

[15] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, (1998).

[16] E. Carrizosa, J. Gordillo, and F. Plastria. Classification Problems with Imprecise Data through Separating Hyperplanes. Technical Report MOSI/33, MOSI Department, Vrije Universiteit Brussel, http://www.vub.ac.be/MOSI/papers/CarrizosaGordilloPlastria.pdf, (2007).

[17] E. Carrizosa, J. Gordillo, and F. Plastria. Support Vector Regression for Imprecise Data. Technical Report MOSI/35, MOSI Department, Vrije Universiteit Brussel, http://www.vub.ac.be/MOSI/papers/CarrizosaPlastriaGordillo_interval_SVR.pdf, (2007).

[18] E. Carrizosa, J. Gordillo, and F. Plastria. Building Separating Concentric Balls to Solve a Multi-instance Classification Problem. Optimization Online, http://www.optimization-online.org/DB_FILE/2008/01/1897.pdf, (2008).

[19] E. Carrizosa, J. Gordillo, and F. Plastria. Kernel Support Vector Regression with Imprecise Output. Optimization Online, http://www.optimization-online.org/DB_FILE/2008/01/1896.pdf, (2008).

[20] E. Carrizosa, J. Gordillo, and D. R. Santos-Peñate. Covering Models with Time-dependent Demand. Optimization Online, http://www.optimization-online.org/DB_FILE/2007/03/1613.pdf, (2007).

[21] E. Carrizosa and B. Martín-Barragán. Máquinas de Vector de Apoyo: Problemas de Programación Matemática. *Boletín de la SEIO*, 21(2):15–20, (2005).

[22] E. Carrizosa and F. Plastria. On Minquantile and Maxcovering Optimisation. *Mathematical Programming*, 71:101–112, (1995).

[23] E. Carrizosa and F. Plastria. Location of Semi-Obnoxious Facilities. *Studies in Locational Analysis*, 12:1–27, (1999).

[24] A. Celminš. Multidimensional Least-squares Fitting of Fuzzy Models. *Mathematical Modelling*, 9(9):669–690, (1987).

[25] P. C. Chen, P. Hansen, B. Jaumard, and H. Tuy. Weber's Problem with Attraction and Repulsion. *Journal of Regional Science*, 32:467–486, (1992).

[26] Y. Chen, J. Bi, and J. Z. Wang. MILES: Multiple-Instance Learning via Embedded Instance Selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1931–1947, (2006).

[27] Y. Chen and J. Z. Wang. Image Categorization by Learning and Reasoning with Regions. *Journal of Machine Learning Research*, 5:913–939, (2004).

[28] V. Cherkassky and Y. Ma. Selection of Meta-Parameters for Support Vector Regression. In *Proc. of the International Conference on Artificial Neural Networks*, pages 687–693, (2002).

[29] P-M. Cheung and J. T. Kwok. A Regularization Framework for Multiple-Instance Learning. In *Proc. 23rd International Conference on Machine Learning*, pages 193–200, (2006).

[30] C. Cortes and V. Vapnik. Support Vector Networks. *Machine Learning*, 20:273–297, (1995).

[31] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, (2000).

[32] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual Categorization with Bags of Keypoints. In *Proc. ECCV'04 Workshop Statistical Learning in Computer Vision*, pages 59–74, (2004).

[33] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, (1997).

[34] F. A. T. De Carvalho, E. A. Lima-Neto, and C. P. Tenorio. A New Method to Fit a Linear Regression Model for Interval-valued Data. In *Proc. of the 27th Annual German Conference on Artificial Intelligence*, pages 295–306, (2004).

[35] L. Devroye and T. J. Wagner. Distribution-free Performance Bounds with the Resubstitution Error Estimate. *IEEE Transactions on Information Theory*, 25(2):208–210, (1979).

[36] P. Diamond. Fuzzy Least Squares. *Information Sciences*, 46:141–157, (1988).

[37] J. M. Díaz-Báñez, F. Hurtado, H. Meijer, D. Rappaport, and T. Sellares. The Largest Empty Annulus Problem. *International Journal of Computational Geometry and Applications*, 13(4):317–325, (2003).

[38] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the Multiple Instance Problem with Axis-Parallel Rectangles. *Artificial Intelligence*, 89:31–71, (1997).

[39] T-N. Do and F. Poulet. Kernel Methods and Visualization for Interval Data Mining. In *Proc. of the Conference on Applied Stochastic Models and Data Analysis*, pages 345–354, (2005).

[40] D. R. Dooly, Q. Zhang, S. A. Goldman, and R. A. Amar. Multiple-Instance Learning of real-valued Data. *Journal of Machine Learning Research*, 3:651–678, (2002).

[41] N. R. Draper and H. Smith. *Applied Regression Analysis*. John Wiley, New York, (1998).

[42] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik. Support Vector Regression Machines. In *Proc. Advances in Neural Information Processing Systems*, volume 9, pages 155–161, (1997).

[43] A. P. Duarte-Silva and P. Brito. Linear Discriminant Analysis for Interval Data. *Computational Statistics*, 21:289–308, (2006).

[44] P. D'Urso. Linear Regression Analysis for Fuzzy/Crisp Input and Fuzzy/Crisp Output Data. *Computational Statistics and Data Analysis*, 42:47–72, (2003).

[45] J. Eichborn and O. Chapelle. Object Categorization with SVM: Kernels for Local Features. Technical report, Max Planck Institute for Biological Cybernetics, (2004).

[46] L. El Ghaoui, G.R.G. Lanckriet, and G. Natsoulis. Robust Classification with Interval Data. Technical Report CSD-03-1279, Division of Computer Science, University of California, Berkeley, (2003).

[47] E. Erkut and S. Neuman. Analytical Models for Locating Undesirable Facilities. *European Journal of Operational Research*, 40:275–291, (1989).

[48] E. Frank and X. Xu. Applying Propositional Learning Algorithms to Multi-instance Data. Working Paper. University of Waikato, (2003).

[49] J. H. Friedman. Another Approach to Polychotomous Classification. Technical report, Stanford University, (1996).

[50] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola. Multi-Instance Kernels. In *Proc. 19th International Conference on Machine Learning*, pages 179–186, (2002).

[51] J. Gordillo, F. Plastria, and E. Carrizosa. Locating a Semi-Obnoxious Facility with Repelling Polygonal Regions. Technical Report MOSI/20, MOSI Department, Vrije Universiteit Brussel, http://www.vub.ac.be/MOSI/papers/GordilloPlastriaCarrisoza2006_Location.pdf, (2006).

[52] S. Gunn. Support Vector Machines for Classification and Regression. Technical Report ISIS-1-98, Department of Electronics and Computer Science, University of Southampton, (1998).

[53] D. J. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. MIT Press, Cambridge, (2001).

[54] P. Hansen and N. Mladenovic. Variable Neighborhood Search: Principles and Applications. *European Journal of Operational Research*, 130:449–467, (2001).

[55] J. A. Hartigan. *Clustering Algorithms*. John Wiley and Sons, New York, (1975).

[56] T. Hastie and R. Tibshirani. Classification by Pairwise Coupling. *The Annals of Statistics*, 26(2):451–471, (1998).

[57] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, New York, (2001).

[58] R. Herbrich. *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press, Cambridge, (2002).

[59] J-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer-Verlag, Berlin, (1993).

[60] D. H. Hong and C. Hwang. Support Vector Fuzzy Regression Machines. *Fuzzy Sets and Systems*, 138(2):271–281, (2003).

[61] D. H. Hong and C. Hwang. Extended Fuzzy Regression Models Using Regularization Method. *Information Sciences*, 164:31–36, (2004).

[62] C-W. Hsu and C-J. Lin. A Comparison of Methods for Multi-class Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, (2002).

[63] C. Hwang, D. H. Hong, E. Na, H. Park, and J. Shim. Interval Regression Analysis Using Support Vector Machine and Quantile Regression. In L. Wang and Y. Jin, editors, *Fuzzy Systems and Knowledge Discovery*, pages 100–109. Springer-Verlag, Berlin, (2005).

[64] C. Hwang, D. H. Hong, and K. H. Seok. Support Vector Interval Regression Machine for Crisp Input and Output Data. *Fuzzy Sets and Systems*, 157:1114–1125, (2006).

[65] J-T. Jeng, C-C. Chuang, and S-F. Su. Support Vector Interval Regression Networks for Interval Regression Analysis. *Fuzzy Sets and Systems*, 138:283–300, (2003).

[66] R. Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1137–1143, (1995).

[67] R. Kondor and T. Jebaram. A Kernel between Sets of Vectors. In *Proc. 21th International Conference on Machine Learning*, pages 361–368, (2003).

[68] G. R. G. Lanckriet, L. El Ghaoui, C. Bhattacharyya, and M. I. Jordan. A Robust Minimax Approach to Classification. *Journal of Machine Learning Research*, 3:555–582, (2002).

[69] H. Lee and H. Tanaka. Upper and Lower Approximation Models in Interval Regression Using Regression Quantile Techniques. *European Journal of Operational Research*, 116:653–666, (1999).

[70] E. A. Lima-Neto and F. A. T. De Carvalho. Centre and Range Method for Fitting a Linear Regression Model to Symbolic Interval Data. *Computational Statistics and Data Analysis*, 52(3):1500–1515, (2008).

[71] E. A. Lima-Neto, F. A. T. De Carvalho, and E. S. Freire. Applying Constrained Linear Regression Models to Predict Interval-valued Data. In *Proc. of the 28th Annual German Conference on Artificial Intelligence*, pages 92–106, (2005).

[72] R. J. A. Little. Regression with Missing X's. *Journal of the American Statistical Association*, 87(420):1227–1237, (1992).

[73] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, New Jersey, (2002).

[74] W. Z. Liu, A. P. White, S. G. White, S. G. Thompson, and M. A. Bramer. Techniques for Dealing with Missing Values in Classification. In *Proc. of the Second International Symposium on Intelligent Data Analysis*, pages 527–536, (1997).

[75] M. Magnani. Techniques for Dealing with Missing Data in Knowledge Discovery Tasks. http://magnanim.web.cs.unibo.it/data/pdf/missingdata.pdf, (2004).

[76] O. L. Mangasarian and E. W. Wild. Multiple Instance Classification via Successive Linear Programming. *Journal of Optimization Theory and Applications*, To appear, (2008).

[77] O. Maron and T. Lozano-Pérez. A Framework for Multiple-Instance Learning. In *Proc. Advances in Neural Information Processing Systems*, volume 10, pages 570–576, (1998).

[78] O. Maron and A. L. Ratan. Multiple-Instance Learning for Natural Scene Classification. In *Proc. 15th International Conference on Machine Learning*, pages 341–349, (1998).

[79] B. Martín-Barragán. *Mathematical Programming for Support Vector Machines*. PhD thesis, Universidad de Sevilla, (2006).

[80] D. Mattera and S. Haykin. Support vector machines for dynamic reconstruction of a chaotic system. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods. Support Vector Learning*, pages 211–242. MIT Press, Cambridge, (1998).

[81] N. Mladenovic and P. Hansen. Variable Neighborhood Search. *Computers and Operations Research*, 24:1097–1100, (1997).

[82] J. M. Moguerza and A. Muñoz. Support Vector Machines with Applications. *Statistical Science*, 21(3):322–336, (2006).

[83] NEOS. Server for Optimization. http://www-neos.mcs.anl.gov/.

[84] S. Nickel and E. M. Dudenhoffer. Weber's Problem with Attraction and Repulsion under Polyhedral Gauges. *Journal of Global Optimization*, 11:409–432, (1997).

[85] Y. Ohsawa. Bicriteria Euclidean Location Associated with Maximin and Minimax Criteria. *Naval Research Logistics*, 47:581–592, (2000).

[86] Y. Ohsawa, F. Plastria, and K. Tamura. Euclidean Push-Pull Partial Covering Problems. *Computers and Operations Research*, 33:3566–3582, (2006).

[87] A. Okabe, B. Boots, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley and Sons, Chichester, (1992).

[88] A. Okabe and A. Suzuki. Locational Optimization Problems Solved through Voronoi Diagrams. *European Journal of Operational Research*, 98:445–456, (1997).

[89] H. T. Pao, S. C. Chung, Y. Y. Xu, and H-C Fu. An EM based Multiple Instance Learning Method for Image Classification. *Expert Systems with Applications*, To appear, (2008).

[90] E. Périnel and Y. Lechevallier. Symbolic Discrimination Rules. In H-H. Bock and E. Diday, editors, *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*, pages 244–265. Springer-Verlag, Berlin, (2000).

[91] F. Plastria and E. Carrizosa. Undesirable Facility Location with Minimal Covering Objective. *European Journal of Operational Research*, 119(1):158–180, (1999).

[92] F. Plastria and E. Carrizosa. Gauge-Distances and Median Hyperplanes. *Journal of Optimization Theory and Applications*, 110:173–182, (2001).

[93] J. C. Platt, N. Cristianini, and J. Shawe-Taylor. Large Margin DAGs for Multiclass Classification. In *Advances in Neural Information Processing Systems 12*, pages 547–553, (2000).

[94] S. Ray and D. Page. Multiple Instance Regression. In *Proc. of the 18th International Conference on Machine Learning*, pages 425–432, (2001).

[95] F. Rossi and B. Conan-Guez. Multi-layer Perceptron on Interval Data. In K. Jajuga, A. Sokolowski, and H-H. Bock, editors, *Classification, Clustering and Data Analysis*, pages 427–434. Springer-Verlag, Berlin, (2002).

[96] F. J. Ruíz, N. Agell, and C. Angulo. A Kernel Intersection Defined on Intervals. In *Proc. of the Catalan Conference in Artificial Intelligence*, (2004).

[97] J. Scheffer. Dealing with Missing Data. *Research Letters in the Information and Mathematical Sciences*, 3:153–160, (2002).

[98] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, (2002).

[99] M. I. Shamos and D. Hoey. Closest-Point Problems. In *Proc. of the 16th Annual IIEE Symposium on Foundations of Computer Science*, pages 151–162, (1975).

[100] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, (2004).

[101] J. Síma. Neural Expert Systems. *Neural Networks*, 8(2):261–271, (1995).

[102] A. J. Smola and B. Schölkopf. A Tutorial on Support Vector Regression. *Statistics and Computing*, 14:199–222, (2004).

[103] H. Tanaka and H. Ishibuchi. Possibilistic Regression Analysis Based on Linear Programming. In J. Kacprzyk and M. Fedrizzi, editors, *Fuzzy Regression Analysis*, pages 47–60. Omnitech Press, Warsow, (1992).

[104] H. Tanaka and H. Lee. Interval Regression Analysis by Quadratic Programming Approach. *IEEE Transactions on Fuzzy Systems*, 6:473–481, (1998).

[105] H. Tanaka, S. Uejima, and K. Asai. Linear Regression Analysis with Fuzzy Model. *IEEE Transactions on Systems, Man and Cybernetics*, 12:903–907, (1982).

[106] D. M. J. Tax and R. P. W. Duin. Using Two-class Classifiers for Multiclass Classification. In *Proceedings of the 16th International Conference on Pattern Recognition*, pages 124–127, (2002).

[107] T. B. Trafalis and S. A. Alwazzi. Support Vector Regression with Noisy Data: a Second Order Cone Programming Approach. *International Journal of General Systems*, 36(2):237–250, (2007).

[108] T. B. Trafalis and R. C. Gilbert. Robust Classification and Regression Using Support Vector Machines. *European Journal of Operational Research*, 173(3):893–909, (2006).

[109] T. B. Trafalis and R. C. Gilbert. Robust Support Vector Machines for Classification and Computational Issues. *Optimization Methods and Software*, 22(1):187–198, (2007).

[110] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, Reading, MA, (1977).

[111] H. Tuy, F. Al-Khayal, and F. Zhou. A D.C. Optimization Method for Single Facility Location Problems. *Journal of Global Optimization*, 7:209–227, (1995).

[112] R. J. Vanderbei. LOQO User's Manual - Version 4.05. Technical Report ORFE-99, Operations Research and Financial Engineering, Princeton University, New Jersey, (2000).

[113] V. N. Vapnik. *The Nature of Statistical Learning*. Springer-Verlag, New York, (1995).

[114] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, (1998).

[115] J. Wang and J. D. Zucker. Solving the Multiple-Instance Problem: A Lazy Learning Approach. In *Proc. 17th International Conference on Machine Learning*, pages 1119 – 1126, (2000).

[116] N. Weidmann, E. Frank, and B. Pfahringer. A Two-level Learning Method for Generalized Multi-instance Problems. In *Proc. 14th European Conference on Machine Learning*, pages 468–479, (2003).

[117] J. Weston and C. Watkins. Multi-class Support Vector Machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, (1998).

[118] X. Xu. Statistical Learning in Multiple Instance Problems. Master's thesis, University of Waikato, (2003).

[119] C. Zhang, X. Chen, M. Chen, S. C. Chen, and M. L. Shyu. A Multiple Instance Learning Approach for Content Based Image Retrieval Using One-Class Support Vector Machine. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 1142– 1145, (2005).

[120] Q. Zhang and S. A. Goldman. EM-DD: An Improved Multiple-Instance Learning Technique. In *Proc. Advances in Neural Information Processing Systems*, volume 14, pages 1073–1080, (2002).

[121] Z-H. Zhou. Multi-Instance Learning: A Survey. Technical report, AI Lab, Department of Computer Sciences and Technology, Nanging University, (2004).

[122] Z-H. Zhou, K. Jiang, and M. Li. Multi-Instance Learning Based Web Mining. *Applied Intelligence*, 22(2):135–147, (2005).