Discretization of Continuous Features by Using a Kernel

L. González[1], F. Velasco[1], F.J. Cuberos[2], J.A. Ortega[3] and C. Angulo[4]

[1] COSDE Research Group, Dept. Applied Economics I, University of Seville (Spain) {luisgon, velasco}@us.es
[2] Dept. Planificación-Radio Televisión de Andalucía, Seville (Spain) fjcuberos@rtva.es
[3] Dept. Computer Science, University of Seville (Spain) ortega@lsi.us.es
[4] GREC Research Group, Technical University of Catalonia, Vilanova i Geltrú. (Spain) cangulo@esaii.upc.es

Keywords: Qualitative knowledge, Time series, TV-viewing share

1 Motivation

The interoperability between the different systems of an company constitutes a fundamental aspect to foment the competitiveness. This competitiveness is maximum in the telecommunication sector, because in this case it must take into account not only the competence between companies but also the involved interests of the holdings which manage them. The increasing regulations of the European Commission which implies the entry of the new competitive television operators have hugely increased the necessity in this sector of offering quality services to the users and auspicious economic results to its management committee. It is necessary to define new techniques for intra-systems interoperability of any company and specifically, in the TV sector. Several works may be found in [6].

In this paper, we introduce a new tool to improve the decision support systems of an company. It helps us to compare time series in an efficient way and with a low computational cost. These techniques are applied to improve the interoperability of two intra-systems of a TV enterprise. They are the programming and merchandising systems.

On the other hand, automated processing and knowledge extraction from data is an important task performed by machine learning algorithms. Hence, the generation of classification rules from class-labelled examples is possible. Instances can be described by a set of numerical, nominal, or continuous features. Several of these algorithms are expressly designed to handle numerical or nominal data; other algorithms perform better with discrete-values features, despite the fact that they can also handle continuous features [13]. Meanwhile a certain number of algorithms developed in the machine learning community focus on learning from nominal feature spaces. Real-world classification includes patterns with continuous features where such algorithms can not be applied, unless the continuous features are firstly discretized. Discretization is the process of transforming a continuous attribute into a finite number of intervals associated with a discrete, numerical value –a number, symbol or letter. This is the usual approach for learning tasks that use mixed-mode – continuous and discrete- data. The Discretization process is developed in two stages: given the range of values for the continuous attribute, first the number of discrete intervals is found; then, the width or boundaries for the intervals.

In [14] it was shown than even on purely numerical-valued data, results for text classification on the derived text-like representation outperforms the more naive numbers-as-tokens representation and, more importantly, is competitive with mature numerical classification methods such as C4.5[15], Ripper[2] and SVM[1, 3, 8, 17]. The most straightforward way is to treat each number that a feature may take on as a distinct "word", and proceed with the use of a text classification method using the combination of true words and tokens-for-numbers words. However, this makes the numbers 1 and 2 as dissimilar as the numbers 1 and 100 –all three values are unrelated tokens to the classification methods. An approach to applying text-classification methods problems with numerical-valued features would be desirable so that the distance between such numerical values can be discerned by the classification method. Most of the methods translating a continuous feature into symbols –letters– in order to deal with texts –letters chains– lose part of their efficient since they are not designed for this task.

The kernel proposed in this paper is specifically designed to work with letters chains coming from a discretization process of a continuous feature and it highlights the properties of these features. To cope the effectiveness of this kernel, it will be used on words from a dictionary where a distance exists between letters of the alphabet. The kernel was firstly proposed to compare among time series that had been converted into symbol chains –words– [4, 5]. Thus, the similarity measure between words quantified a distance between original time series.

The rest of this paper is structured as follows: first, both a kernel and a distance between finite intervals are defined. Distance is used to define a real function measuring the similarity between two words and if words have the same length, this function is a Kernel because it fulfills the Mercer con-

dition. Next, one example about classification rules is developed. Finally, the conclusions and ideas for future works are enumerated.

2 Interval distance from a kernel

In essence, the goal in the construction of kernel functions is to guarantee the existence of an application ϕ, defined from the working set, $\mathcal{X}$ to a vectorial space endowed with a dot product, $\mathcal{F}$. From this function ϕ the kernel function is defined, denoted $k(\cdot,\cdot)$, over pairs of elements of the working set as the dot product of their transformations into the feature space, $k(\cdot,\cdot) = \langle\phi(\cdot),\phi(\cdot)\rangle_{\mathcal{F}}$, where $\langle\cdot,\cdot\rangle$ is denoted a dot product. The kernel function let us $k(\cdot,\cdot)$ establish similarities between the original elements from their transformed ones, so a distance between the points in the input space can be defined. It must be considered, when elaborating a similarity and distance measure, that the ϕ application must be able to highlight the essential characteristics of the initial set of elements[7].

Following the ideas presented in [11], let

$$\mathcal{I} = \left\{(c-r, c+r) \subset \mathbb{R} : c \in \mathbb{R}, r \in \mathbb{R}^+\right\}$$

be the family of all the open intervals contained in the real line of finite dimension (in default, we are working with open intervals, but it is posible to translate the study to closed intervals naturally). A function $\phi_1 : \mathcal{I} \to \mathbb{R}^2$ is defined as: $\phi_1(I) = P(c,r)^t$ and the kernel k and a distance d_1^2 between intervals are:

$$k(I_1,I_2) = \begin{pmatrix} c_1 & r_1 \end{pmatrix} S \begin{pmatrix} c_2 \\ r_2 \end{pmatrix} \qquad d_1^2(I_1,I_2) = \begin{pmatrix} \Delta c & \Delta r \end{pmatrix} S \begin{pmatrix} \Delta c \\ \Delta r \end{pmatrix}$$

where $I_1 = (c_1 - r_1, c_1 + r_1)$, $I_2 = (c_2 - r_2, c_2 + r_2)$, $\Delta c = c_2 - c_1$ and $\Delta r = r_2 - r_1$, and P must be a non singular matrix ($S = P^t P$). Thus, the discretization of a continuous feature in symbols representing different intervals, allows us to use as a distance between symbols the distances defined between intervals as it will be showed.

3 Kernel

From this point, we always consider that the symbols are letters (A, B,$\cdots$) because the ordinal scale is reflected in the alphabetical order. Let $\mathcal{A} = \{A_1, A_2, \cdots, A_\ell\}$ be an alphabet of ℓ letters and let $\mathcal{P}$ be a set of the words obtained from this alphabet. Let $P1 = P1_1 P1_2 \cdots P1_n$ and $P2 = P2_1 P2_2 \cdots P2_m$ be words from $\mathcal{P}$ where $P1_i$, $P2_j \in \mathcal{A}$ and $n \geq m$. A map K_λ is defined as follow:

$$K_\lambda(P1, P2) = \max\left\{\sum_{i=1}^{m} \lambda^{d^2(P1_{i+k}, P2_i)},\ k = 0, \cdots, n-m\right\}$$

where $0 < \lambda < 1$ and $d(\cdot, \cdot)$ is a distance between letters.

Property 1. For all $P1$, $P2 \in \mathcal{P}$ and $0 < \lambda_1 < \lambda_2 < 1$, then: $K_{\lambda_1}(P1, P2) \leq K_{\lambda_2}(P1, P2)$.

Property 2. For all $P1$, $P2 \in \mathcal{P}$ and $0 < \lambda < 1$, then: $K_\lambda(P1, P2) \leq m$. This upper bound is attained: If $P2 = P1_1 P1_2 \cdots P1_m$, and $K_\lambda(P1, P2) = m$.

Property 3. Let $r = \max_{ij} d(A_i, A_j)$, with $A_i, A_j \in \mathcal{A}$. For all $P1$, $P2 \in \mathcal{P}$ and $0 < \lambda < 1$ then: $m\lambda^{r^2} \leq K_\lambda(P1, P2)$. This lower bound is attained: Let $A = A_i$ and $B = A_j$ be such that $d(A, B) = r^2$. If $P1 = AA \cdots A$ and $P2 = BB \cdots B$ with size of $P1$, n, and size of $P2$, m, then $K_\lambda(P1, P2) = m\lambda^{r^2}$.

Thereby, for all $0 < \lambda < 1$:

$$m\,\lambda^{r^2} \leq K_\lambda(P1, P2) \leq m, \quad \forall P1,\ P2 \in \mathcal{P}$$

Property 4. Let $\mathcal{A}$ be an alphabet obtained from a discretization process of a continuous feature and $\mathcal{P} = \{P_1 P_2 \cdots P_n,\ P_i \in \mathcal{A}\}$ the set of all the words having length n, then:

$$K_\lambda(P1, P2) = \sum_{i=1}^{n} \lambda^{d^2(P1_i, P2_i)}$$

is a Kernel.

The proof of these properties are in [9].

3.1 Generalized similarity

Let $P1$ and $P2$ be two words of the same length n from the set $\mathcal{P}$. In the definition of similarity between words, $K_\lambda(P1, P2) = \sum_{i=1}^{n} \lambda^{d^2(P1_i, P2_i)}$, all the letters have the same interest. It is possible to generalize this similarity by weighting each letter in such a form that the sum of the weights is equal to n.

Let $w_1, w_2, \cdots, w_n \in \mathbb{R}$ be scalar numbers accomplishing $w_i \geq 0$ and $\sum_{i=1}^{n} w_i = n$. The generalized similarity can be defined in two different ways:

$$K^1_\lambda(P1, P2) = \sum_{i=1}^{n} \lambda^{w_i \cdot d^2(P1_i, P2_i)} \qquad K^2_\lambda(P1, P2) = \sum_{i=1}^{n} w_i \cdot \lambda^{d^2(P1_i, P2_i)}$$

It is no difficult to prove that both are kernels (the sum and the product of kernels is a kernel [3]); however the second one has a more intuitive meaning for the weights. Also, using the properties of the exponential function we have:

$$K_\lambda^1(P1, P2) = \sum_{i=1}^{n} \lambda^{w_i \cdot d^2(P1_i, P2_i)} = \sum_{i=1}^{n} w_i' \cdot \lambda^{d^2(P1_i, P2_i)}$$

where $w_i' = \lambda^{(w_i - 1)\, d^2(P1_i, P2_i)}$. Although it is no necessarily true that $\sum_{i=1}^{n} w_i' \neq n$. For this we propose as a generalization of similarity the function $K_\lambda^2(\cdot, \cdot)$.

4 Implementation

In the current television, the programming is implemented taking into account the response of the audience according to the inversion carried out. This is known as "share". The interoperability between the programming and exploitation systems is a fundamental aspect in these enterprises. Therefore, it is necessary to dispose of good tools which allow to identify the response of the audience according to the executed programming. The comparison must be done with the responses obtained by the channel in the previous weeks in order to prove if the expected results have been achieved. Besides, the achieved results must be compared with those obtained by the competitive channels. These decision support systems have been designed taking into account that they may be defined by means of a mathematical base. The proposed techniques and methods verify this requirement. The techniques allow to optimize the exploitation of the information systems, by providing comparison mechanisms between the different channels. In particular, this comparison has been made between opened broadcasting channels in Andalucia (Spain).
An example of the classification rule is developed. Data to be considered is a set of television shares from the seven main television stations in Andalusia, Spain. It has been provided by Canal Sur Televisión and it has been collected from [18]. Time series represent the average share for 15 minutes blocks, so the daily series are 96 elements length.

We are going to use several discretization methods and will see that the results are good in all them. A variety of discretization methods can be found in the literature. From the unsupervised algorithms: equal interval width, equal frequency interval, k-means clustering or unsupervised MCC; to supervised algorithms like *ChiMerge*, *CADD*, *1RD*, *D-2* or maximum entropy. An extensive list can be found in [13]. The methods to be evaluated in this work are: i) *Equal Width Intervals* or *EWI*, ii) *Equal Frequency Intervals* or *EFI*, iii) *CAIM* (Class-Attribute Interdependence Maximization) [13], iv) *Ameva* [12], v) *CUM* [10], and vi) *DTW* [16].

In the following step several related task are accomplished: i) The discretization methods are applied over the learning subset producing a set of landmarks, ii) The landmarks are used as the limits of intervals and a symbol is assigned to each one, and iii) the series are translated into symbol chains.

The series are labelled with the name of the corresponding television station. We have selected the first 32 Wednesdays of year 2003 ($32 \cdot 7 = 224$

series) as the input set of series. Other 20 Wednesdays are used as work set (140 series) to be predicted.

In the Equal Width, Equal Frequency and *CUM* methods, the user must specify the number of intervals to be computed. As no rule for an optimal value exist, all those methods will be calculated from 2 to 9 intervals. All the methods are applied to the learning subset and a list of interval boundaries are obtained. Individual letters are assigned in alphabetical order to each interval.

The learning system evaluates (a complete study can be found in [5]) the number of successful identifications on the test subset using the k-neighbours algorithm for each discretization method. The application of the presented methodology achieves a 95% correct identification rate for the work set series, 133 over 140. The best discretization method for this data set was *Equal Frequency Interval* with 3 labels. Table 1 shows the average percentage and variance for all the methods in 200 draws for 1, 3 and 5 neighbours.

In Table 1 can be observed that, although the discretización methods build the intervals following different approaches, except for some anomalous case, the results are similar, that is, the kernel is very robust in front of the discretization methods.

With respect to the parameter λ used in the kernel, it does not significantly affect to the average of correct identification. Table 2 shows that only the *CAIM* method is affected by the variance of λ.

5 Conclusions and further work

A new similarity function for symbol chains has been proposed, generating in some cases a kernel. This function measures similarities between words in a dictionary when a distance measure between symbols is defined. In the near future, we will focus on the extension of this methodology to time series with multiple attributes and other kinds of data. At the same time, we will use new data sets to extend its validation. Finally, it must be mentioned that this kernel has certain implications in the type of considered similarity that will be studied in future researches. The small influence of the λ parameter in identification tasks must also be argued.

6 Acknowledgements

This work was partially supported by the the Junta de Andalucía grants PAI-(2004-2005/SEJ-442). Moreover, it has been partly supported by the Spanish Interministerial Committee of Science and Technology by means of the programs TIN2004-07246-C03-03 and DPI2003-07146-C02-01.

Table 1. Identification Average (%) and Standard Deviation in Test Subset (200 Draws) vs. Number of neighbours

		Neighbours					
		1		3		5	
Method	Labels	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
CAIM	7	90.5	4.26	89.4	4.56	89.1	4.74
AMEVA	3	91.6	2.74	89.4	2.77	89.7	2.81
CUM	2	90.7	2.86	88.4	2.91	89.0	2.98
	3	85.9	4.04	85.1	4.20	86.1	3.88
	4	75.9	6.01	71.3	5.29	70.9	5.59
	5	73.2	5.41	71.0	5.42	72.3	5.52
	6	82.4	4.21	80.8	4.03	80.8	4.95
	7	83.2	3.56	80.0	3.69	80.0	4.28
	8	85.2	3.33	82.8	2.95	82.1	3.36
	9	86.4	3.15	84.9	2.60	84.6	3.13
EFI	2	91.1	2.88	90.9	2.65	90.7	2.87
	3	**95.5**	2.13	95.4	1.98	95.1	2.02
	4	88.8	3.07	87.6	3.15	87.4	3.40
	5	85.2	3.87	85.1	4.14	85.4	3.85
	6	80.2	4.11	77.6	4.71	76.4	4.90
	7	74.6	4.78	71.7	5.31	71.0	5.37
	8	75.7	4.32	71.2	4.91	70.6	5.01
	9	74.7	5.26	70.4	5.27	69.1	6.20
EWI	2	71.0	11.5	65.2	13.2	66.5	12.9
	3	46.0	8.08	36.3	8.26	35.0	8.90
	4	71.9	11.9	67.3	14.2	68.8	14.3
	5	74.9	10.7	71.0	13.0	71.9	11.9
	6	72.3	10.9	68.3	13.7	70.3	13.6
	7	85.8	7.76	84.7	8.22	85.9	8.28
	8	75.3	9.32	73.3	10.3	74.2	11.0
	9	88.1	4.90	87.4	5.76	88.0	5.27
DTW	-	80,2	3,74	78,0	4,44	76,4	4,27

Table 2. Percentage of correct identifications in the Work Set for each method vs. value of λ.

	Lambda								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
CAIM	**0.88**	0.88	0.88	0.86	0.86	0.84	0.84	0.82	**0.80**
AMEVA	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.87
CUM02	0.88	0.88	0.88	0.88	0.88	0.88	0.86	0.86	0.88
EFI03	0.91	0.91	0.91	0.92	0.92	0.92	0.92	0.92	0.90
EWI09	0.85	0.85	0.85	0.85	0.84	0.84	0.85	0.87	0.85

7 References

[1]. C. Angulo and L. González. 1-v-1 Tri-Class SV Machine. In *Proc. 11th European Symposium on Artificial Neural Networks, ESANN*, pages 355–360, 2003.
[2]. W. Cohen. Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123, 1995.
[3]. N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University press 2000, 2000.
[4]. F.J. Cuberos, J.A. Ortega, F. Velasco, and L. González. Qsi - Alternative Labelling and Noise Sensitivity. In *17 International Workshop on Qualitative Reasoning.*, volume 17, pages 229–239, 2003. http://www.unb.br/ib/necbio/QR03/pdfs/QR03posterCuberos.pdf.
[5]. F.J. Cuberos, J.A. Ortega, F. Velasco, and L. González. A methodology for qualitative learning in time series. In *18 International Workshop on Qualitative Reasoning.*, volume 2, pages 147–153, 2004. http://www.qrg.cs.northwestern.edu/QR04/papers/FJC_QR044.pdf.
[6]. Konstantas D., Bourrières J.-P., Léonard M., and Boudjlida N. *Interoperability of Enterprise Software and Applications.* Springer, 2006.
[7]. L. González, , F. Velasco, and R. M. Gasca. A study of the similarities between topics. *Computational Statistics*, 20(3):465–479, 2005.
[8]. L. González, C. Angulo, F. Velasco, and M. Vilchez. Máquina ℓ-SVCR con salidas probabilísticas. *Inteligencia Artificial. Revista Iberoamericana de IA*, (17):72–82, 2002. In Spanish.
[9]. L. González, F.J. Cuberos, F. Velasco, and J.A. Ortega. Un núcleo entre literales. Tech. Report 02, Dept. of Applied Economy I, University of Seville (Spain), 2003.
[10]. L. González and J.M. Gavilán. Una metodología para la construcción de histogramas. Aplicación a los ingresos de los hogares andaluces. *XIV Reunión ASEPELT-Spain*, 2001.
[11]. L. González, F. Velasco, C. Angulo, J.A. Ortega, and F. Ruiz. Sobre núcleos, distancias y similitudes entre intervalos. *Inteligencia Artificial. Revista Iberoamericana de IA*, (23):111–117, june 2004. In Spanish.
[12]. L. González, F. Velasco, F.J. Cuberos, and J.A. Ortega. Ameva: A discretization algorithm. *Machine Learning*, in Revision:–, 2006.
[13]. L. Kurgan and K.J. Cios. Caim discretization algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 16(2):145–153, 2004.
[14]. A.A. Macskassy, H. Hirsh, A. Banerjee, and A. Dayanik. Converting numerical calssification into text classification. *Artificial Inteligence*, (143):51–77, 2003.
[15]. J.R. Quinlan. *C 4.5 Programs for Machine Learning.* Morgan Kaufmann, 1993.
[16]. H. Sakoe and S. Chiba. Dynamic-programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(1):43–49, 1978.
[17]. B. Schölkopf and A. J. Smola. *Learning with Kernel.* MIT Press, 2002.
[18]. TNS Audiencia de Medios. A service of Sofres AM company. www.sofresam.com, year 2003.