Empirical Evaluation of the Difficulty of Finding a Good Value of k for the Nearest Neighbor

Francisco J. Ferrer–Troyano, Jesús S. Aguilar–Ruiz, and José C. Riquelme

Department of Computer Science, University of Sevilla
Avenida Reina Mercedes s/n, 41012 Sevilla, Spain
{ferrer,aguilar,riquelme}@lsi.us.es

Abstract. As an analysis of the classification accuracy bound for the Nearest Neighbor technique, in this work we have studied if it is possible to find a *good value* of the parameter k for each example according to their attribute values. Or at least, if there is a pattern for the parameter k in the original search space. We have carried out different approaches based on the Nearest Neighbor technique and calculated the prediction accuracy for a group of databases from the UCI repository. Based on the experimental results of our study, we can state that, in general, it is not possible to know a priori a specific value of k to correctly classify an unseen example.
Keywords: Nearest Neighbor, Local Adaptive Nearest Neighbor.

1 Introduction

In Supervised Learning, systems based on examples (CBR, Case Based Reasoning) have been object of study and improvement from their introduction at the end of the fifties. These algorithms extract knowledge through inductive processes from the partial descriptions given by the initial set of examples or instances. Machine learning process is usually accomplished in two functionally different phases. In the first phase of Training a model of the hyperspace is created by the labelled examples. In the second phase of Classification the new examples are labelled based on the constructed model. In the Nearest Neighbor algorithm (from here on *NN*) the training examples are the model itself. *NN* assigns to each new query the label of its nearest neighbor among those that are remembered from the phase of Training. In order to improve the accuracy with noise present in data, the *k-NN* algorithm introduces a parameter k so that for each new example q to be classified the classes of the k nearest neighbors of q are considered: q will be labelled with the majority class or, in case of tie, it is randomly broken. Another alternative consists in assigning that class whose average distance is the smallest one or introducing a heuristically obtained threshold $k_1 < k$ so that the assigned class will be that with a number of associated examples greater than this threshold [10]. Extending the classification criterion, the *k-NN*$_{wv}$ algorithms (Nearest Neighbor Weighted Voted) assign weights to the prediction made by each example. These weights can be inversely proportional to the distance with respect to the example to be classified [4, 6]. Therefore, the

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2658, pp. 766–773, 2003.

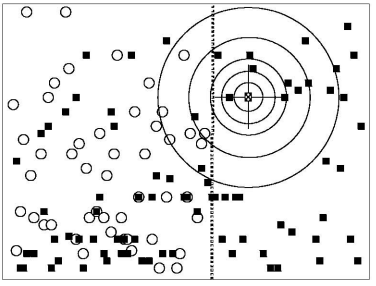

Fig. 1. Horse Colic database. If the new example to be classified is a central point, the k value slightly determines the assigned label.

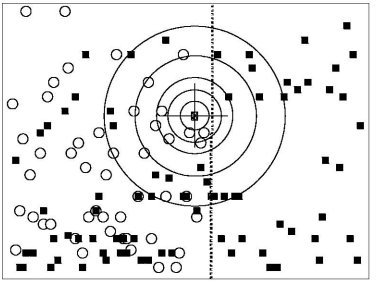

Fig. 2. Horse Colic database. If the new query is a border point, the k value can be decisive in the classification.

k number of examples observed and the metric used to classify a test example are decisive parameters. Usually k is heuristically determined by the user or by means of cross-validation [9]. The usual metrics of these algorithms are the Euclidean distance for continuous attributes and the Overlap distance for nominal attributes (both metrics were used in our experiments).

In the last years have appeared interesting approaches that test new metrics [13] or new data representations [2] to improve accuracy and computational complexity. Nevertheless, in spite of having a wide and diverse field of application, to determine with certainty when *k-NN* obtains higher accuracy than *NN* [1] and viceversa [8] is still an open problem. In [5] it was proven that when the distance among examples with the same class is smaller than the distance among examples of different class, the probability of error for *NN* and *k-NN* tends to 0 and $\frac{1}{2}$, respectively. But, not always this distribution for input data appears, reason why *k-NN* and *k-NN*$_{wv}$ can improve the results given by *NN* with noise present in the data. In [12] a study of the different situations in which *k-NN* improves the results of *NN* is exposed, and four classifiers are proposed (Locally Adaptive Nearest Neighbor, *localKNN*$_{ks}$) where for each new example q to be classified the parameter k takes a value k_q which is near to the values that classified the M (an extra parameter) nearest neighbors e_q of q.

In this work we intend to study the limits that the *k-NN* algorithm presents even when the value of k is not fixed but variable for each example. When an example as the Figure 1 illustrates is interior to a region of examples with its same label (it is a central point), the assigned label will depend little on the value of k. However, with an example near the decision boundaries (a border point, see Figure 2) the choice such parameter can be decisive. In the following section we explain several results obtained after applying the standard and weighted k–Nearest Neighbor algorithm (from now *k-NN* and *k-NN*$_{wv}$) with databases from the UCI repository [3]. In principle it seems logical to consider that the classification accuracy can improve when the k value is adjusted locally. In a previous work [7] we introduced a local nearest neighbor classifier which evaluates

Table 1. Percentage of examples that are correctly classified by kNN and kNN$_{wv}$ where k is an odd number belongs to the interval [1,11].

	k-NN						*k-NN*$_{wv}$					
DB	k=1	k=3	k=5	k=7	k=9	k=11	k=1	k=3	k=5	k=7	k=9	k=11
An	92.53	88.97	87.63	83.96	85.30	85.52	92.53	91.2	91.87	91.42	91.42	92.09
Aud	74.33	66.81	63.71	59.73	58.40	60.61	74.33	76.1	73.45	72.56	69.02	69.46
Aut	75.60	66.82	60.97	57.07	57.07	57.56	75.6	77.07	78.04	75.6	73.17	71.7
BS	79.03	79.84	80.32	86.4	88.96	88.80	79.03	79.84	80.32	87.03	89.6	89.28
BC	70.27	69.58	72.02	74.12	74.12	74.82	70.27	68.53	71.32	72.72	74.12	74.47
CHD	74.58	81.84	81.51	82.17	81.51	81.18	74.58	80.19	80.85	82.17	81.51	81.51
CR	81.88	85.94	86.52	86.52	86.81	86.37	81.88	84.63	85.21	85.65	85.94	85.94
GC	72.60	73.00	73.30	72.89	72.89	73.20	72.6	73.0	73.0	73.1	73.1	73.7
Gl	70.09	68.22	64.01	61.21	58.87	57.47	70.09	71.49	72.89	70.56	68.69	68.22
HS	75.55	79.25	80.00	81.11	80.37	81.48	75.55	78.88	80.0	81.85	80.74	81.11
He	80.64	82.58	83.87	83.87	84.51	84.51	80.64	82.58	82.58	83.22	82.58	81.93
HC	68.47	69.29	69.02	70.38	69.83	69.02	68.47	70.65	71.46	73.64	73.09	70.92
Io	86.89	86.03	85.47	84.04	84.33	84.04	86.89	86.03	85.75	84.04	84.33	84.04
Ir	95.33	95.33	95.33	96.66	95.33	94.66	95.33	95.33	95.33	96.0	94.66	94.66
PD	70.57	74.08	74.08	75.26	73.82	73.43	70.57	73.95	73.69	74.86	73.95	73.56
PT	34.21	29.20	33.03	35.69	34.21	35.39	34.21	27.13	30.38	31.56	31.85	32.15
Son	87.50	83.65	82.21	80.28	75.96	72.59	87.5	83.65	82.69	82.69	81.25	77.4
Soy	91.80	91.80	91.06	90.48	90.19	89.31	91.8	91.8	91.94	91.21	91.36	91.06
Ve	69.85	68.43	67.73	68.91	67.49	66.90	69.85	71.04	71.63	71.74	69.85	70.21
Vot	91.03	91.49	92.64	93.56	93.10	93.56	91.03	91.26	91.95	92.87	92.87	93.1
Vow	99.39	97.97	94.24	89.89	83.53	42.72	99.39	98.08	97.27	96.06	94.94	94.74
Wi	95.50	96.62	96.06	96.06	96.06	95.50	95.5	96.62	96.06	96.62	96.62	96.06
WBC	95.27	96.56	96.99	96.85	96.85	96.70	95.27	96.56	96.99	96.85	96.7	96.7
Zoo	96.03	92.07	93.06	91.08	89.10	89.10	96.03	92.07	95.04	95.04	93.06	92.07
Av.	80.37	79.81	79.36	79.09	78.27	76.43	80.37	80.74	81.24	81.63	81.02	80.67

several k values to decide the label for a new query. But the results obtained in the really interesting domains were very similar to the results given by *k-NN*, so the added computational complexity (the calculation of this local k value) can not be worth. In the next empirical analysis we show that local classifiers based on geometric proximity present a limit for classifying very near the Nearest Neighbor prediction ability, and we try to find somehow an empirical measure for that limit.

2 Empirical Evaluation

In first place we obtained the error rates by leave-one-out validation increasing the value of k. The chosen limits for the maximum values of k were three: 11, 31 and 51. The results obtained for the odd numbers in the interval [1,11] applying *k-NN* and *k-NN*$_{wv}$ are showed in Table 1. Table 2 shows the average values for k in [1,11], [1,31] and k in [1,51]. Observing both Tables we can state that

Table 2. Average percentage of examples that are classified by *k-NN* and *k-NN*$_{wv}$ with an odd value of k in the intervals [1,11], [1,31] and [1,51].

	k-NN			*k-NN*$_{wv}$		
DB	k in [1,11]	k in [1,31]	k in [1,51]	k in [1,11]	k in [1,31]	k in [1,51]
Anneal−	87.32	84.51	83.18	91.75	91.34	90.77
Audiology−	63.93	58.51	54.11	72.49	70.46	67.75
Autos−	62.52	56.28	53.45	75.20	73.32	71.76
Balance-Scale+	83.89	87.56	88.06	84.18	87.72	88.44
Breast-Cancer	72.49	73.68	73.60	71.91	73.51	73.66
Cleveland-HD+	80.47	81.47	82.07	80.14	81.82	82.40
Credit-Rating+	85.67	85.90	86.18	84.87	85.74	86.34
German-Credit	72.98	72.93	72.78	73.08	73.38	73.39
Glass−	63.31	60.57	59.99	70.32	67.31	65.79
Heart-Statlog+	79.62	81.20	82.16	79.69	81.06	81.92
Hepatitis	83.33	82.86	82.03	82.25	82.54	82.03
Horse-Colic−	69.33	67.45	66.96	71.37	69.98	69.34
Ionosphere−	85.13	83.26	80.11	85.18	83.60	81.10
Iris	95.44	95.58	95.25	95.22	95.41	95.56
Pima-Diabetes+	73.54	74.20	74.58	73.43	74.51	74.73
Primary-Tumor+	33.62	37.62	38.88	31.21	33.99	34.78
Sonar−	80.36	73.58	72.61	82.53	75.72	74.90
Soybean−	90.77	86.91	80.23	91.53	90.60	88.39
Vehicle−	68.22	67.20	65.97	70.72	69.49	68.36
Vote	92.56	92.39	91.94	92.18	92.35	91.90
Vowel−	84.62	34.40	21.17	96.75	95.63	95.63
Wine	95.97	96.34	96.45	96.25	96.52	96.65
Wisconsin-BC	96.54	96.54	96.35	96.51	96.55	96.38
Zoo−	91.74	86.26	80.69	93.89	92.69	90.74
Averages	78.89	75.72	74.12	80.94	80.63	80.11

the performance of both algorithms is very similar, although there is a slight tendency in favor of *k-NN*$_{wv}$ with regard to the obtained accuracy. From Tables 1 and 2 we can also observe that some databases have a high difficulty to be classified, for instance, *Primary-Tumor* or *Glass*. Aiming for obtaining a priori the best value of k for each example, we wonder: "what would the gain be if it is possible to find such a value for k?" , i.e., "how many examples are correctly classified for some value of k?". In other words: "how many examples are not correctly classified for any value of k by the Nearest Neighbor algorithm?".

That is, there is not a value of k for which most of the k nearest neighbors of an example has the same label as such an example. This value provides an interesting rate because it gives an error bound for the Nearest Neighbor algorithm, and generally, for any classification method based on geometric proximity.

To answer this question, we measured for each example all those values of k (among 1 and 51) that classified it correctly. If there was not value of k which classified a certain example correctly, this example was indicated as non-classifiable.

Table 3. Percentage of examples that are not able to be correctly classified by *k-NN* and *k-NN*$_{wv}$ for any k in the intervals [1,11], [1,31] and [1,51].

	k-NN			*k-NN*$_{wv}$		
DB	k in [1,11]	k in [1,31]	k in [1,51]	k in [1,11]	k in [1,31]	k in [1,51]
Anneal	2.78	2.56	2.45	4.90	4.56	4.34
Audiology	17.69	15.92	15.92	17.69	16.37	15.92
Autos	12.68	9.27	9.27	17.07	14.14	13.65
Balance-Scale	8.32	8.16	8.16	8.32	8.16	8.16
Breast-Cancer	14.68	12.93	12.58	17.83	16.78	16.43
Cleveland-HD	11.22	9.90	8.58	12.87	11.55	10.89
Credit-Rating	9.13	7.82	7.68	11.15	9.71	8.55
German-Credit	13.0	10.50	10.39	13.40	11.29	10.90
Glass	19.15	13.55	13.08	19.15	16.35	15.42
Heart-Statlog	9.63	9.26	8.89	10.0	9.26	8.89
Hepatitis	9.68	7.74	7.10	12.90	10.96	9.03
Horse-Colic	17.11	14.94	14.13	17.39	15.21	14.13
Ionosphere	8.55	6.84	6.84	8.55	7.69	7.69
Iris	3.33	2.67	2.0	4.0	3.33	2.0
Pima-Diabetes	14.19	11.19	10.28	14.71	11.97	11.19
Primary-Tumor	48.37	42.18	40.70	57.52	53.39	51.91
Sonar	6.25	4.33	4.33	6.25	4.33	4.33
Soybean	5.12	4.10	4.10	5.42	4.25	4.25
Vehicle	14.53	11.58	10.04	15.24	12.41	11.46
Vote	4.14	4.14	4.14	4.83	4.83	4.83
Vowel	0.50	0.50	0.50	0.50	0.50	0.50
Wine	1.68	1.12	1.12	1.68	1.12	1.12
Wisconsin-BC	2.0	1.72	1.72	2.0	1.72	1.72
Zoo	2.97	1.98	1.98	2.97	1.98	1.98
Averages	10.67	8.95	8.58	11.93	10.49	9.97

Table 3 indicates the percentage of non-classifiable points for both techniques according to the fixed limits. Let's observe *Horse-Colic*. The 17.11% of examples does not correctly classify with any value of k in [1,11], the 14.94% of examples does not correctly classify with any k in [1,31] and the 14.14% is not correctly classified with k in [1,51]. Thus, we can state that there is not significant difference among the limits 31 and 51, as well as between *k-NN* and *k-NN*$_{wv}$.

From Table 3 a maximum bound of the classification ability of *k-NN* can be obtained, still knowing a priori the value of k. That is, although *k-NN* could adapt locally so that for each example to be classified, according to the values of its attributes, we calculated the *best* k, the error rates given in Table 3 can not be avoided. However, there are some databases in which the improvement in the accuracy can be worth the computational effort (the calculation of that local k). So, taking again *Horse-Colic*, we can observe in Tables 1 and 3 that we would have an error rate of 14.14% instead of 30.77%, i.e., an improvement of around

Table 4. Percentage of examples that have at least a number cvk of common values of k which classify it correctly and classify its nearest neighbor correctly by means of *k-NN*, when $k \in [1, 51]$ and $cvk \in \{1, 3, 5, 7, 9, 11, 31, 51\}$.

DB/cvk	1	3	5	7	9	11	31	51
Anneal	93.65	91.09	88.86	86.41	85.30	84.18	78.06	63.91
Audiology	78.31	68.58	62.83	61.94	61.06	59.73	39.38	23.89
Autos	79.51	70.24	63.90	60.0	57.56	57.07	35.60	15.12
Balance-scale	84.0	82.24	81.92	81.44	81.12	81.12	78.88	53.12
Breast-cancer	76.92	68.53	66.78	66.43	66.08	65.73	61.18	29.37
Cleveland-HD	79.86	76.56	75.24	74.25	73.26	71.94	68.31	53.46
Credit-rating	84.20	82.6	81.44	80.28	80.28	80.0	75.79	60.43
German-credit	77.10	69.19	66.0	64.50	63.70	62.70	56.49	32.80
Glass	71.49	64.95	61.68	57.0	55.14	54.20	46.72	31.30
Heart-statlog	80.37	73.33	73.33	72.59	71.11	71.11	69.62	55.18
Hepatitis	85.16	82.58	81.93	77.41	76.77	76.77	69.67	59.35
Horse-colic	70.65	63.58	58.96	56.79	55.97	54.89	51.63	31.52
Ionosphere	89.17	85.18	83.19	82.90	82.05	81.48	69.80	61.53
Iris	96.0	96.0	95.33	94.66	94.66	94.66	94.66	86.66
Pima-diabetes	75.52	70.44	68.75	67.57	66.53	66.01	57.94	38.28
Primary-tumor	36.87	34.21	31.26	30.97	30.08	29.79	23.0	9.44
Sonar	91.34	87.01	83.17	79.32	74.03	69.23	61.05	39.42
Soybean	92.38	91.36	90.04	88.72	87.84	87.70	73.20	54.02
Vehicle	76.71	69.26	65.13	62.17	60.28	59.33	50.35	33.68
Vote	92.41	90.80	90.11	90.11	89.19	89.19	88.04	84.13
Vowel	99.49	96.66	90.70	85.65	77.07	35.75	0	0
Wine	98.87	98.31	97.75	97.75	96.62	96.62	94.94	87.07
Wisconsin-BC	95.56	94.27	94.13	94.13	94.13	94.13	93.13	89.98
Zoo	96.03	94.05	91.08	89.10	86.13	85.14	78.21	60.39
Averages	83.39	79.20	76.81	75.08	73.58	71.18	63.15	48.08

50%. In general, logically, the highest increment is given for those databases that we point out previously as difficult to be classified by means of *k-NN*.

Related to our initial objective that was to find a relationship among the attributes values of any example and some *correct value* of k to classify it correctly, we chose two databases with a significant gain, *Glass* and *Horse-Colic*. Next, for each domain we built a new database, where the label of each example was replaced by the minimum value of k for which such example was correctly classified. Then different approaches were attempted to predict the parameter k: lineal and quadratic regression through traditional statistical methods, *C4.5* (where the leaves in the decision tree obtained are possible values of k, and the Nearest Neighbor algorithm itself in a similar form to Locally Adaptive NN methods [11]. None of these techniques was able to improve the average error rate obtained by the standard *k-NN* by applying ten-folds cross-validation. Note that it is not necessary applying again the Nearest Neighbor algorithm to vali-

date this last prediction approach because for each point we had calculated if a certain k value gave a correct classification.

In a second approach we consider that maybe the problem could be in the choice of the minimum k as the label of the database, due to the possible relationship between the original space of attributes and the k value could be formed by a set of different k values. These new values might not necessarily coincide with the smallest k. In order to solve this problem and to obtain more exact information, a second database was built. In this new database, the label of each example was replaced with a set of values that classified correctly such example. Given the special features of these data sets (multi-labelled), we carried out different approaches through regressions (lineal, quadratic, and quotient of polynomials). In this manner, the adjustment was correct if for each point the value obtained by means of regression was some of the *k-labels* associated the point. However, like the previous case, this type of regressions presented some results that did not improve the basic Nearest Neighbor algorithm.

Finally, in order to measure what extent reaches the relationship between the k obtained for each example and the region of the space where this example is located, it was calculated the number of common k values that classified an example and its nearest neighbor. The results are shown in Table 4. In this Table we can observe that, again for *Horse-Colic*, only the 70.65% of examples have at least a k value shared with its nearest neighbor. In addition, this percentage decreases quickly when increasing the requirement that the number of shared k values must be higher. This means that if we tried to predict the k that classifies a point correctly according to the k that classified its nearest neighbors correctly, we would have a minimum error rate of 29.35%, that is, the points for which its nearest neighbor have not any value that classifies it correctly. It is important to notice that the values in Table 4 provide a superior bound of the probability to *guess* the parameter k in function of the nearest neighbor of an example, but it does not mean that this probability will be reached easily. In fact, we can observe that for the databases that we have denominated difficult, with 3 or 4 common values, the percentage is so low that it seems complicated to determine the correct k by means of the Nearest Neighbor.

3 Conclusions and Future Directions

A priori, we could consider that the value of k to classify a new query through the Nearest Neighbor must depend on the space region in which the such example is located. Thus, when the point is central, the value of k can be low, and when it is near to the decision boundaries, such value must be higher. Nevertheless, after our study, we can conclude that it is not possible to determine with certainty the relationship between the attribute values for a particular point and the values of k that classifies it correctly through *k-NN*. To reach this observation, different tests have been carried out on databases from the UCI repository trying to establish which are the accuracy that *k-NN* gives as classification method. In this sense, we infer that to find a space distribution of the values of k in the

attributes space is not an easy task. As sample, it is enough with verifying that the percentage of common values of k between a point and its nearest neighbor falls quickly and, therefore, the disposition in concrete regions of the values of k for a later correct estimation of the same one does not seem feasible. At least, by applying the traditional tools as regression, 1-NN or C4.5.

For future works we are trying to predict the correct value of k for each point from the original search space using genetic programming, which provides a capacity for obtaining regressions that are not bound by previous conditions. Another possible approach that we are studying is to consider the value of k based on the *enemy* instead of the nearest neighbor, since this can provide us a measurement of the proximity from a point to the decision bound of the region in which it is.

Acknowledgment.

The research was supported by the Spanish Research Agency CICYT under grant TIC2001-1143-C03-02.

References

1. D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.
2. S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for nearest neighbor searching. In *Proceedings of 5th ACM SIAM Symposium on discrete Algorithms*, pages 573–582, 1994.
3. C. Blake and E. K. Merz. Uci repository of machine learning databases, 1998.
4. S. Cost and S. Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10:57–78, 1993.
5. T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IT-13(1):21–27, 1967.
6. S.A. Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-6, 4:325–327, 1975.
7. F. J. Ferrer, J. S. Aguilar, and J. C. Riquelme. Nonparametric nearest neighbor with local adaptation. In *Proceedings of the* 10^{th} *Portuguese Conference on Artificial Intelligence*, Porto, Portugal, December 2001.
8. R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11:63–91, 1993.
9. M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B*, 36:111–147, 1974.
10. I. Tomek. An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics*, 6(6):448–452, June 1976.
11. D. Wettschereck and T. G. Dietterich. An experimental comparison of nearest neighbor and nearest hyperrectangle algorithms. *Machine Learning*, 19(1):5–28, 1995.
12. D. Wettschereck and T.G. Dietterich. Locally adaptive nearest neighbor algorithms. *Advances in Neural Information Processing Systems*, (6):184–191, 1994.
13. D. R. Wilson and T. R. Martinez. Improved heterogeneous distance functions. *Journal of Artificiall Intelligence Research*, 6(1):1–34, 1997.