Searching for a pattern of k in the Nearest–Neighbor Algorithms *

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

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

Abstract. In this work we have empirically investigated the general dataset conditions that make possible to find a value of the parameter k which classifies each test example correctly by means of the Nearest Neighbor algorithm. In this search, we have compared different approaches based on decision trees, regression trees, and geometric proximity. In addition, we have measured the difficulty of classifying a set of UCI databases as a function of the values of k that classify the nearest neighbor of an example correctly. At this stage of the investigation, we can state that, in general, determining a priori the k–values which classify each test example correctly presents a high computational cost and improves NN's accuracy scarcely.

1 Introduction

Since their introduction in the 1950's, important studies about the error bounds for the Nearest Neighbor have been published [4, 1, 5, 7]. Researchers have also developed very interesting approaches which investigated new metrics [10] or new data representations [2] for improving accuracy and computational complexity. In [4] it was shown that the error of the Nearest Neighbor is bounded by twice the optimal Bayes probability of error. In addition, it was proven that when the distance among same label examples is smaller than the distance among different label examples, the probability of error for *NN* and *k-NN* tends to 0 and $\frac{1}{2}$, respectively. But this condition is not always satisfied, which is the reason that *k-NN* and *k-NN*$_{wv}$ can improve the accuracy given by *NN*. In [8] the behavior of *NN* and *k-NN* is studied in depth and the experiments carried out with six synthetic data sets confirm the two following hypotheses: a) Noisy data need large values for k; b) The performance of *k-NN* is less sensitive to the choice of a metric. In addition, four classifiers are proposed (*Locally Adaptive Nearest Neighbor*), where the value of k can be different for each new example q to be classified. In the two first methods (*localkNN*$_{ks}$) the parameter k takes a value k_q which is the most frequent value among those that classified the M nearest neighbors of q [9]. The

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

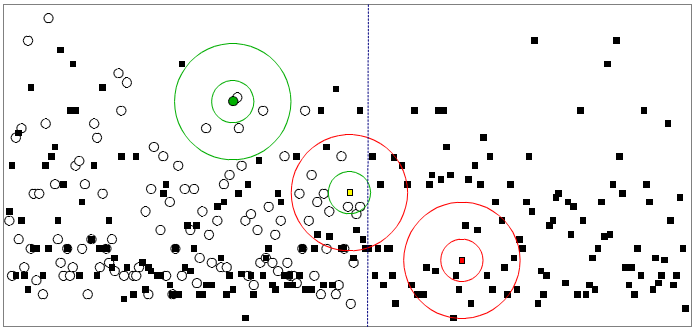

Fig. 1. Horse Colic database. If the new example to be classified is a central point, the majority class by *k-NN* and *k-NN*$_{wv}$ is not highly sensitive to the chosen value of k. If the new example to be classified is a border point, the chosen value of k can be determinant.

third method (*localkNN*$_{onekperclass}$) computes and stores for each class c_i the single k_{c_i} value that more examples it classified. The new query q is evaluated as many times as there are labels in training set, so that the used value every time is the k_{c_i} associated to each label. The fourth method (*localkNN*$_{onekpercluster}$) uses the unsupervised clustering algorithm RPCL [6] to determine different clusters from the training set. Then a single k value is associated to each cluster by leave-one-out cross-validation. The new query is classified according to the k value of the cluster it is assigned to. However, experiments with UCI data sets [3] shows that these local Nearest Neighbor methods do not improve significantly the performance of *k-NN*. So it my be difficult to justify the added computational complexity. Nevertheless, determining with certainty when local NN learners are a beneficial decision is still an open problem.

In this work we intend to study empirically the performance of Nearest Neighbor learners even when the value of k is not constant but variable for each example. In principle, when a new query is interior to a region (it is a central point), the classification by proximity does not depend on the number of observed neighbors. However, when a query is near a decision boundary, the value of k can be critical (see Fig. 1). That is, it might be possible to improve the classification accuracy by using different values of k for each example. From such premise, the reason for our study is knowing if the parameter k can be tuned in continuous regions of the search space. In other words, we have tried to find all the correct values of k that classify an example correctly when it is near the decision boundaries. In the following section we present several results obtained by applying the *k-NN* and *k-NN*$_{wv}$ algorithms to datasets from the UCI repository. The weight used in our experiments was $\frac{1}{d}$. These results show that, using the original space of attributes, local Nearest Neighbor learners do not significantly improve NN's accuracy.

Table 1. Percentage of examples that are correctly classified in *k-NN* and *k-NN*$_{wv}$ with k in [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

2 Empirical Evaluation

We begin our study observing how the classification accuracy given by *k-NN* and *k-NN*$_{wv}$ changes as the value of the parameter k increases. So in the first place we obtained the accuracy rates by leave-one-out cross-validation with the value of k restricted to the odd numbers (to avoid ties) in the interval [1,11]. Note that there is no need to restrict the parameter k to an odd number in *k-NN*$_{wv}$. Table 1 shows the results obtained. Although there is significant variability in the case of *k-NN*, the behavior of *k-NN*$_{wv}$ is more stable empirically. For instance, we can observe *Autos* and *Vowel* where the accuracy decreases from 75.60% and 99.39% to 57.56% and 42.72% respectively by *k-NN*. However, in *k-NN*$_{wv}$ the corresponding decreases in accuracy are less than 5% in both data sets. So it seems correct to consider that *k-NN*$_{wv}$ is more robust than *k-NN* with either noise present in data or similar distributions for each attribute. Nevertheless, with the interval [1,11] for the values of k in the studied domains, these results are not sufficient to determine if the accuracy is an increasing or decreasing function of k.

To investigate the behavior of the algorithms more fully, we extended the range of the values of k and calculated the accuracy for each new odd number by both techniques. In experiments with 51 as the maximum value of k, we noted that the predominant behavior of the accuracy as a function of k tended to be decreasing or increasing, depending on each database. Due to the limitations of space, Table 2 shows only the average accuracy obtained for three intervals: [1,11], [1,31] and [1,51]. Databases signed with + are classified with greater accuracy as the value of k increases for the three intervals and both methods,

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

whereas databases signed with − are classified with decreasing accuracy as the value of k increases. In both Tables, the data obtained for *Vowel* database are very meaningful with respect to the sensitivity of *k-NN* and *k-NN*$_{wv}$ to the chosen value of k. The loss of accuracy in *k-NN*$_{wv}$ reaches about 5% whereas in *k-NN* it can exceed 50%. Thus we can state that, in the most studied domains, the performances of both techniques are similar, although there is a slight average tendency in favor of *k-NN*$_{wv}$. In addition, *k-NN*$_{wv}$ gives generally a more robust performance. From Tables 1 and 2 we can also observe that some databases have low levels of accuracy, e.g. *Audiology*, *Glass*, *Horse-Colic*, *Primary-Tumor*, and *Vehicle*. In these *difficult databases* the accuracy changes smoothly as a function of the values of k. Such data can be taken as an approximate indicator of the distribution of the examples in the search space.

Aiming for the correct values of k, we wonder: *"What gain we would obtain if such values were known?"*, that is, *"How many examples are correctly classified for some value of k?"*. Posing the question in another form: *"How many examples are not correctly classified for any value of k by the Nearest Neighbor?"* i.e., there is no value of k for which most of the k nearest neighbors of an example has the same label as such an example. This is an important question because the answer can provide an error bound for the Nearest Neighbor and generally, for any classifier based on geometric proximity. To answer the question, we measured for each example all the odd values of k (between 1 and 51) that classified it correctly. If there was no value of k for an example, then it was indicated as non-classifiable. Table 3 shows the percentage of examples for which there is no value of k that classifies them correctly by means of *k-NN* and *k-NN*$_{wv}$. This data give an approximated indicator of the degree of difficulty to classify a database

Table 3. Percentage of examples that can not 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

by means of the Nearest Neighbor. We can observe that for certain databases, increasing the possible values of k has a little effect (*Anneal, Iris, Vote, Vowel, Wine, Wisconsin, Zoo*) whereas in other *difficult databases* (*Glass, Horse-Colic, Pima-Diabetes, Primary-Tumor, Vehicle*) this increase can condition strongly the classification accuracy. Although in the first group of domains, the difference between restricting k to 6 values and allowing 26 values is scarcely appreciable, in the second group it can have considerable improvements to classify a new query. *Primary-Tumor* shows clearly this phenomenon decreasing from 67% to 40%, i.e., the relative improvement of examples that find a correct value of k is relatively about 50%.

In all the studied databases, the value of k that more examples classified correctly was 1 for both methods. Table 4 shows (in Columns 2 and 6 for *k-NN* and *k-NN*$_{wv}$ respectively) the percentage of examples classified correctly using $k = 1$. These percentages are given with respect to only the examples that were classified by a greater value of k. Columns 1 and 3 show the number of different values of k that *k-NN* found in the interval [1,51] and the highest of these values. Columns 5 and 7 are the same for *k-NN*$_{wv}$. Columns 4 and 8 show the mean and the standard deviation for the different values found in each database. After calculating such data, we observed that the highest values classified few examples in comparison with the number of examples that were classified by low values of k. In addition, as k was extended, the difference between the two last values of k which classified some examples increased. It seems logical to consider that such examples, although classified, could be outliers of the database. If we also observe the mean and the standard deviation for each database, it seems sufficient to use few and low values for k.

Table 4. Number of different values of k that classified correctly some example by k-*NN* and k-NN_{wv}.

	k-*NN*				k-NN_{wv}			
DB	values	% (k=1)	k-max	mean±sd	values	% (k=1)	k-max	mean±sd
An	9	95.2	49	1.3±2.1	12	97.2	30	1.2±1.8
Aud	11	86.0	45	1.9±4.1	11	85.6	46	2.0±4.9
Aut	14	84.7	29	2.3±4.3	13	86.6	51	2.5±6.4
BS	12	88.4	23	1.6±1.8	12	84.7	23	1.6±1.8
BC	9	77.6	27	1.9±2.5	10	79.9	29	1.8±2.6
CHD	13	81.9	43	2.3±5.2	10	84.1	46	1.8±3.4
CR	11	88.8	41	1.5±2.7	22	89.4	49	1.9±4.5
GC	17	81.0	37	2.2±3.7	23	81.6	48	2.2±4.1
Gl	15	80.6	45	2.9±5.8	12	82.9	36	2.5±5.5
HS	8	82.9	45	1.8±3.3	11	82.9	51	1.9±3.7
He	9	86.8	39	2.1±4.8	10	88.1	48	2.5±6.8
HC	13	79.7	41	2.4±4.8	16	79.7	42	2.5±4.9
Io	9	93.2	31	1.6±3.1	7	94.2	17	1.3±1.6
Ir	5	97.2	35	1.4±3.1	5	97.3	40	1.7±4.7
PD	23	78.7	51	2.7±5.7	28	79.5	50	2.5±5.2
PT	29	55.7	47	5.8±9.3	22	69.1	49	4.0±7.1
Son	8	91.5	23	1.7±2.9	8	91.5	23	1.7±2.8
Soy	14	95.1	25	1.3±2.0	14	95.2	25	1.3±2.1
Ve	36	77.5	49	3.0±6.3	33	78.8	49	2.7±5.5
Vot	4	95.7	7	1.1±0.6	5	96.1	8	1.1±0.6
Vow	2	99.9	9	1.0±0.3	2	99.9	9	1.0±0.3
Wi	5	96.6	29	1.3±2.2	5	96.6	29	1.3±2.2
WBC	6	97.2	25	1.1±1.2	6	97.2	26	1.1±1.2
Zoo	2	99.0	11	1.1±0.9	2	99.0	10	1.1±0.9
Av.	11.8	87.4	33.6	1.6±3.4	12	88.5	34.2	1.9±3.5

Returning to Table 3, consider the *Horse-Colic* database. The 17.11% of examples does not correctly classify with any value of k in [1,11], 14.94% does not correctly classify with any k in [1,31] and 14.14% are not correctly classified when k belongs to interval [1,51]. Thus, we can state that there is not significant difference between the limits 31 and 51, or between k-*NN* and k-NN_{wv}. From this Table a maximum bound of the classification ability of k-*NN* can be obtained, even if the value of k is known a priori. That is, although k-*NN* could adapt locally so that we could hit a correct value of k for each new example to be classified, the error rates of 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* as an example we can observe in Tables 2 and 3, that we would have an error rate of 14.13% (Column 6 in Table 3) instead of 30.77% (Column 1 in Table 2), i.e., an improvement of around 50%. Logically, in general the highest increment is given for those databases that we point out previously as *difficult* to be classified using a technique based on the Nearest Neighbor.

Related to our initial objective, that was to find a relationship among the values of the attributes of an example and a value of k to classify it correctly, we built two new databases, where the label of each example was substituted by the minimum value of k for which such an example was correctly classified by k-*NN* and k-NN_{wv}. All non-classifiable examples were removed. Different approaches were attempted for predicting the value of k: regression trees by *M5*, decision trees by *J4.8* (*C4.5*) and the Nearest Neighbor itself in a similar form to *Locally Adaptive k-NN* [9]. None of these techniques improved the average accuracy rate obtained by the standard k-*NN* using ten-fold cross-validation. Note that it is

Table 5. Prediction of the value of k by decision trees, regression trees, and geometric proximity. The value of k was calculated by *k-NN*.

	C4.5			*M5*		*1-NN*	
DB	PA	RE	NR	RE	NR	PA	RE
An	95.2	92.0	1	99.1	17	93.7	72.1
Aud	86.1	84.2	3	113.3	1	84.0	68.3
Aut	82.8	87.0	1	113.3	1	80.1	74.0
BS	84.7	95.5	1	100.5	1	84.0	84.2
BC	77.6	95.5	1	98.2	1	74.8	81.0
CHD	79.8	94.3	1	102.9	1	80.1	68.4
CR	88.8	94.5	1	94.0	2	87.7	63.3
GC	81.0	96.7	1	97.2	1	71.4	86.9
Gl	76.3	81.0	12	105.4	1	75.8	77.0
HS	80.9	88.1	17	101.9	2	81.3	66.3
He	86.8	83.1	1	103.6	1	84.0	74.2
HC	79.7	94.0	1	107.3	3	72.5	80.8
Io	91.7	89.9	6	96.8	1	90.5	79.0
Ir	97.3	68.8	1	99.0	2	97.3	70.4
PD	75.2	93.1	1	97.6	1	73.0	76.0
PT	55.7	94.9	1	99.9	1	51.2	79.7
Son	90.9	71.4	5	112.6	15	88.9	75.5
Soy	95.1	84.5	1	102.3	2	93.6	78.0
Ve	74.0	91.4	36	98.0	1	71.1	77.5
Vot	95.7	93.0	1	97.1	2	95.2	68.9
Vowl	99.9	66.8	1	100.0	1	99.9	70.4
Wi	96.6	76.9	1	97.9	5	96.6	66.1
WBC	97.2	89.0	1	93.9	7	96.2	70.7
Zoo	99.0	68.4	1	2247.7	1	99.0	55.6
Av.	86.17	86.42	4.04	190.81	2.95	84.24	73.51

not necessary to apply again the *k-NN* method to validate it because for each example we calculated if a certain k produced a correct classification. The results obtained for the new databases with the *k-label* calculated by *k-NN* and *k-NN*$_{wv}$ are shown in Tables 5 and 6 respectively. We used the WEKA Environment [11] with the default options for *M5* and *J4.8*. Since determining the value of k is a prediction task rather than a classification task, the accuracy rate is no longer appropriate: errors are not simply present or absent, they come in different sizes. So we also observed the *relative-absolute-error*, defined as:

$$\Sigma_{i=1}^{n} \frac{|p_i - k_i|}{|k_i - \overline{k}|} \tag{1}$$

where n is the number of examples (see Table 3), p_i is the prediction of k for each example i, k_i is the new label assigned to each example i and $\overline{k}$ the mean of k. Columns PA indicate the prediction accuracy given by *NN* and *C4.5*. Columns RE indicate the *relative-absolute-error* given by the three predictor method. Finally, Columns NR indicate the number of rules or leaves generated by *C4.5* and *M5*. We can observe that, in both cases, the best prediction method is the Nearest Neighbor itself. Although the average accuracy in *C4.5* is about 2% greater than in *1-NN*, the average *relative-absolute-error* is smaller by means of *1-NN*. When the value of k is calculated by *k-NN* (Table 5), the average *relative-absolute-error* of the prediction given by *C4.5* and *1-NN* is 86.42% and 73.51% respectively. That is to say, *1-NN* offers an improvement with respect to *C4.5* of about 17%. When the value of k is calculated by *k-NN*$_{wv}$ (Table 6), the average error in the prediction given by *C4.5* and *1-NN*$_{wv}$ is 85.68% and 61.17% respectively. In this case, *1-NN*$_{wv}$ offers an improvement with respect to

Table 6. Prediction of the value of k by decision trees, regression trees, and geometric proximity. The value of k was calculated by k-NN_{wv}.

	C4.5			*M5*		*1-NN$_{wv}$*	
DB	PA	RE	NR	RE	NR	PA	RE
Anneal	97.2	82.3	1	99.5	1	96.0	59.5
Audiology	83.6	90.8	3	113.8	1	83.6	60.5
Autos	86.6	84.5	1	108.6	1	84.9	53.8
Balance-scale	84.7	95.5	1	100.5	1	84.0	83.3
Breast-cancer	79.9	94.1	1	101.0	7	77.8	75.3
Cleveland-HD	84.1	92.8	1	105.1	1	79.6	68.2
Credit-rating	89.4	88.6	1	104.2	1	87.8	56.4
German-credit	81.6	95.4	1	95.6	1	72.9	81.3
Glass	78.4	85.7	9	101.3	3	79.6	59.7
Heart-statlog	78.9	93.6	14	93.3	2	81.3	58.4
Hepatitis	86.6	85.7	1	100.5	4	85.2	59.2
Horse-colic	79.4	92.7	1	108.0	3	72.5	75.7
Ionosphere	93.5	85.9	6	104.6	3	92.3	61.5
Iris	97.3	68.8	1	98.5	1	97.3	36.2
Pima-diabetes	77.7	92.7	23	91.8	2	73.0	71.3
Primary-tumor	69.1	90.9	1	93.0	2	63.6	69.4
Sonar	90.9	71.4	5	112.6	15	88.9	60.6
Soybean	95.2	84.0	1	101.4	2	93.7	70.6
Vehicle	77.0	90.2	1	97.4	9	73.2	68.1
Vote	96.1	89.7	1	91.7	2	95.9	58.6
Vowel	99.9	66.8	1	100.1	1	99.9	34.5
Wine	96.6	76.9	1	97.9	5	96.6	41.9
Wisconsin-BC	97.2	89.0	1	93.9	7	96.2	64
Zoo	99.0	68.4	1	2247.7	1	99.0	40.1
Average	87.49	85.68	3.25	190.08	3.12	85.62	61.17

C4.5 about 40%. Even so, the added computational complexity for predicting the correct k is not worthwhile. On the other hand, the number of rules generated by *C4.5* and *M5* was 1 in most of the databases, in which the only value predicted was *k=1*.

In a second approach we considered that perhaps the problem could be in the choice of the minimum k as the class of the databases, because the possible relationship between the space of attributes and the value of k could be determined for a set of several values. That is, the best k might not necessarily coincide with the minimum k. In order to solve this problem and to obtain more exact information, a second set of databases was built. In these new domains, the label of each example was replaced with a set formed by several values of k that classified such an example correctly. With the mean and the standard deviation for the values of k obtained in a previous experiment (see Table 4), we restricted the size of the set of k values to 5 for all databases. Due to the special features of these data sets (multi-labelled), we attempt different approaches by means of regressions (linear and quadratic). In this manner, the adjustment was correct if for each example the value obtained by means of regression was some label assigned that example. However, as in the previous experiment, the results obtained did not improve significantly the accuracy obtained with a single k value.

To examine the extent of the relationship between the value of k obtained for each example and the region of the space where this example is found, we calculated the number of common values of k for each example and its nearest neighbor. The results are shown in Table 7. In this table we can observe that, again for the database *Horse-Colic*, only 70.65% of the examples have at least one

Table 7. 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

common k value with its nearest neighbor, and this percentage decreases quickly when the number of common k values required is higher. This means that if we tried to predict the k that classifies an example correctly according to the k that classified its nearest neighbors correctly, we would have a minimum error rate of 29.35%. This percentage represents the examples for which there is no value of k such that the example and its nearest neighbor are both correctly classified. It is necessary to point out that the values in Table 6 provide a superior bound of the probability to *guess* the value of k as a function of their nearest neighbors. But it does not mean that this probability will be reached easily. In fact, we can observe that for the domains that we have identified as *difficult databases*, with 3 or 4 common values of k, the percentage is so low that it seems complicated to determine the correct k by means of the Nearest Neighbor algorithm.

3 Conclusions and Future Directions

A priori, we might consider that the value of k in the Nearest Neighbor must depend on the region of the original attribute space in which each example is located. Thus, when the example is a central point the value of k would be low, and when it is a border point the value of k would be higher. In order to verify this assumption, we have carried out different tests on a set of UCI databases in an attempt to bound the classification accuracy given by the Nearest Neighbor. After our experimental study, we must conclude that it is not possible to identify a relationship between the values of the attributes for border point and the values of k that classify it correctly by means of k-*NN*.

In this sense, we can infer that to find a distribution of the values of k in the original attribute space is not an easy task. Verifying that the percentage of common values of k that classify an example and its nearest neighbor decrease quickly in *difficult databases* is a sufficient test. In addition, the added computational complexity can be *prohibitive*. Therefore, the location of the k values in different regions for obtaining a later correct estimation of them it does not seem feasible. At least by traditional techniques such as regression trees (*M5*), decision trees (*C4.5*), and geometric proximity (*NN*). In our current research, we are trying to predict the k values by data transformation, using prototypes and feature construction. We are also investigating another approach based on the *nearest enemy* instead of the nearest neighbor. This can provide us with a measurement of the proximity of an example to the decision boundaries which define the region in which it is located.

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. T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IT-13(1):21–27, 1967.
5. R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11:63–91, 1993.
6. A. Krzyzak L. Xu and E. Ola. Rival penalized competitive learning for clustering analysis, rbf net, and curve detection. *IEEE Transactions on Neural Networks*, 4(4):636–649, 1993.
7. D. Heath S. Salzberg, A. Delcher and S Kasif. Best-case results for nearest neighbor learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):599–610, 1995.
8. C. Wettschereck. *A Study of Distance-Based Machine Learning Algorithms*. PhD thesis, Oregon State University, 1995.
9. D. Wettschereck and T.G. Dietterich. Locally adaptive nearest neighbor algorithms. *Advances in Neural Information Processing Systems*, (6):184–191, 1994.
10. D. R. Wilson and T. R. Martinez. Improved heterogeneous distance functions. *Journal of Artificiall Intelligence Research*, 6(1):1–34, 1997.
11. Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementatios*. Morgan Kauffman, 1999.