Data Characterization for Effective Prototype Selection

Ramón A. Mollineda, J. Salvador Sánchez, and José M. Sotoca

Dept. Llenguatges i Sistemes Informàtics, Universitat Jaume I,
Av. Sos Baynat s/n, E-12071 Castelló de la Plana, Spain
{mollined,sanchez,sotoca}@uji.es

Abstract. The Nearest Neighbor classifier is one of the most popular supervised classification methods. It is very simple, intuitive and accurate in a great variety of real-world applications. Despite its simplicity and effectiveness, practical use of this rule has been historically limited due to its high storage requirements and the computational costs involved, as well as the presence of outliers. In order to overcome these drawbacks, it is possible to employ a suitable prototype selection scheme, as a way of storage and computing time reduction and it usually provides some increase in classification accuracy. Nevertheless, in some practical cases prototype selection may even produce a degradation of the classifier effectiveness. From an empirical point of view, it is still difficult to know a priori when this method will provide an appropriate behavior. The present paper tries to predict how appropriate a prototype selection algorithm will result when applied to a particular problem, by characterizing data with a set of complexity measures.

1 Introduction

One of the most widely studied non-parametric classification approaches corresponds to the k-Nearest Neighbor (k-NN) decision rule [3]. Given a set of n previously labeled instances (training set, TS), the k-NN classifier consists of assigning an input sample to the class most frequently represented among the k closest instances in the TS, according to a certain dissimilarity measure. A particular case of this rule is when $k = 1$, in which an input sample is assigned to the class indicated by its closest neighbor.

The asymptotic classification error of the k-NN rule (i.e., when n grows to infinity) tends to the optimal Bayes error rate as $k \to \infty$ and $k/n \to 0$. Moreover, if $k = 1$, the error is bounded by approximately twice the Bayes error [3]. The optimal behavior of this rule in asymptotic classification performance along with a conceptual and implementational simplicity make it a powerful classification technique capable of dealing with arbitrarily complex problems, provided that there is a large enough TS available.

Nevertheless, this theoretical requirement of large TS size is also the main problem using the 1-NN rule because of the seeming necessity of a lot of memory and computational resources. This is why numerous investigations have been concerned with finding new approaches that are efficient with computations. Within this context, many fast algorithms to search for the NN have been proposed. Alternatively, some prototype selection techniques [1, 4, 6] have been directed to reduce the TS size by selecting only the most relevant instances among all the available ones, or by generating new prototypes in locations accurately defined.

J.S. Marques et al. (Eds.): IbPRIA 2005, LNCS 3523, pp. 27–34, 2005.

On the other hand, in many practical situations the theoretical accuracy can hardly be achieved because of certain inherent weaknesses that significantly reduce the effective applicability of k-NN classifiers in real-world domains. For example, the performance of these rules, as with any non-parametric classification approach, is extremely sensitive to data complexity. In particular, class-overlapping, class-density, and incorrectness or imperfections in the TS can affect the behavior of these classifiers. Other prototype selection methods [5, 10, 13, 14] have been devoted to improve the 1-NN classification performance by eliminating outliers (i.e., noisy, atypical and mislabeled instances) from the original TS, and by reducing the possible overlapping between regions from different classes.

Despite the apparent benefits of most prototype selection algorithms, in some domains these techniques might not achieve the expected results due to certain data characteristics. For this reason, it seems interesting to know a priori the conditions under which the application of a prototype selection scheme can become appropriate. A set of data complexity measures [7, 8] are used in this paper to predict when a prototype selection technique leads to an improvement with respect to the plain 1-NN rule.

2 Data Complexity Measures

The behavior of classifiers is strongly dependent on data complexity. Usual theoretical analysis consists of searching accuracy bounds, most of them supported by impractical conditions. Meanwhile, empirical analysis is commonly based on weak comparisons of classifier accuracies on a small number of unexplored data sets. Such studies usually ignore the particular geometrical descriptions of class distributions to explain classification results. Various recent papers [7, 8] have introduced the use of measures to characterize the data complexity and to relate such descriptions to classifier performance.

In [7, 8], authors define some complexity measures for two classes. For our purposes, a generalization of such measures for the n-class problem is accomplished. The ideal goal is to represent classification problems as points in a space defined by a number of measures, where clusters can be related to classification performances. Next paragraphs describe the measures selected for the present study (the same short notation as in the original paper [7] is here used).

Generalized Fisher's Discriminant Ratio (F1). The plain version of this well-known measure computes how separated are two classes according to a specific feature. It compares the difference between class means with the sum of class variances. A possible generalization for C classes, which also considers all feature dimensions, can be stated as follows:

$$F1 = \frac{\sum_{i=1}^{C} n_i \cdot \delta(m, m_i)}{\sum_{i=1}^{C} \sum_{j=1}^{n_i} \delta(x_j^i, m_i)} \qquad (1)$$

where n_i denotes the number of samples in class i, δ is a metric, m is the overall mean, m_i is the mean of class i, and x_j^i represents the sample j belonging to class i.

Volume of Overlap Region (F2). The original measure computes, for each feature, the length of the overlap range normalized by the length of the total range in which all values of both classes are distributed. The volume of the overlap region for two classes is the product of normalized lengths of overlapping ranges for all features. Our generalization sums this measure for all pairs of classes, that is,

$$F2 = \sum_{(c_i,c_j)} \prod_k \frac{\min\{\max(f_k,c_i),\max(f_k,c_j)\} - \max\{\min(f_k,c_i),\min(f_k,c_j)\}}{\max\{\max(f_k,c_i),\max(f_k,c_j)\} - \min\{\min(f_k,c_i),\min(f_k,c_j)\}} \tag{2}$$

where (c_i, c_j) goes through all pair of classes, k takes feature index values, while $\min(f_k, c_i)$ and $\max(f_k, c_i)$ compute the minimum and maximum values of feature f_k in class c_i, respectively.

Feature Efficiency (F3). In [7], the feature efficiency is defined as the fraction of points that can be separated by a particular feature. For a two-class problem, the original measure takes the maximum feature efficiency. This paper considers the points in the overlap range (instead of those separated points as in the original formulation). The measure value for C classes is the overall fraction of points in some overlap range of any feature for any pair of classes. Obviously, points in more than one range are counted once. This measure does not take into account the joint contribution of features.

Non-parametric Separability of Classes (N2, N3). The first measure (N2) is the ratio of the average distance to intraclass nearest neighbor and the average distance to interclass nearest neighbor. It compares the intraclass dispersion with the interclass separability. Smaller values suggest more discriminant data. The second measure (N3) is simply the estimated error rate of the 1-NN rule by the leaving-one-out scheme.

Density Measure (T2). This measure does not characterize the overlapping level, but contributes to understand the behavior of some classification problems. It describes the density of spatial distributions of samples by computing the average number of instances per dimension.

3 Prototype Selection

Prototype Selection (PS) techniques have been proposed as a way of minimizing the problems related to the k-NN classifier. They consist of selecting an appropriate reduced subset of instances and applying the 1-NN rule using only the selected examples. Two different families of PS methods exist in the literature: editing and condensing algorithms.

Editing [5, 10, 13–15] eliminates erroneous cases from the original set and "cleans" possible overlapping between regions from different classes, what usually leads to significant improvements in performance. Thus the focus of editing is not on reducing the set size, but on defining a high quality TS by removing outliers. Nevertheless, as a by-product these algorithms also obtain some decrease in size and consequently, a reduction of the computational burden of the 1-NN classifier.

Wilson [14] introduced the first editing proposal. Briefly, this consists of using the k-NN rule to estimate the class of each instance in the TS, and removing those whose class label does not agree with that of the majority of its k neighbors. Note that this algorithm tries to eliminate mislabeled instances from the TS as well as close border instances, smoothing the decision boundaries.

On the other hand, condensing [1, 4, 6, 9, 11, 12] aims at selecting a sufficiently small set of training instances that produces approximately the same performance than the 1-NN rule using the whole TS. It is to be noted that many condensing schemes make sense only when the classes are clustered and well-separated, which constitutes the focus of the editing algorithms.

Hart's algorithm [6] is the earliest attempt at minimizing the number of stored instances by retaining only a *consistent* subset of the original TS. A consistent subset, say S, of a set of instances, T, is some subset that correctly classifies every instance in T using the 1-NN rule. Although there are usually many consistent subsets, one generally is interested in the *minimal* consistent subset (i.e., the subset with the minimum number of instances) to minimize the cost of storage and computing time. Unfortunately, Hart's algorithm cannot guarantee that the resulting subset is minimal in size.

4 Experimental Results and Discussion

As already stated in Sect. 1, in some cases PS algorithms may produce an effect different from the one theoretically expected, that is, they may even degrade the performance of the plain 1-NN classifier. A way of characterizing the problems could be by using the data complexity measures introduced in Sect.2. Thus the experiments reported in this paper aim at describing the databases in terms of such measures and analyzing the conditions under which PS methods can perform better than the plain 1-NN rule.

In our experiments, we have included a total number of 17 data sets taken from the UCI Machine Learning Database Repository (`http://www.ics.uci.edu/~mlearn`) and from the ELENA European Project (`http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/`). The 5-fold cross-validation error estimate method has been employed for each database: 80% of the available instances have been used as the TS and the rest of instances for the test set. The main characteristics of these data sets and their values for the complexity measures previously described are summarized in Table 1.

For the PS methods, we have tested Wilson's editing, Hart's condensing, and the *combining* edited and condensed set. In this latter case, we have firstly applied Wilson's editing to the original TS in order to remove mislabeled instances and smooth the decision boundaries, and then Hart's algorithm has been used over the Wilson's edited set to further reduce the number of training examples. After preprocessing the TS by means of some PS scheme, the 1-NN classifier has been applied to the test set.

Table 2 reports the error rate and the percentage of original training instances retained by each method for each database. Typical settings for Wilson's editing algorithm (i.e., number of neighbors) have been tried and the ones leading to the best performance have been finally included. The databases are sorted by the value of F1. By means of the data complexity measures, we have tried different orderings which could give us an indication of the relation between the complexity of a data set and the particular method applied to it. From all those measures, it seems that F1 is the one that better discriminates between the cases in which an editing has to be firstly applied and those in which one could directly employ the plain 1-NN rule.

Table 1. Experimental data sets: characteristics and complexity measures.

	Classes	Dim	Samples	F1	F2	F3	N2	N3	T2
Cancer	2	9	683	1.315	0.319	0.902	0.220	0.950	76
Clouds	2	2	5000	0.245	0.380	0.877	0.019	0.846	2500
Diabetes	2	8	768	0.032	0.252	0.994	0.839	0.679	96
Gauss	2	2	5000	0.000	0.309	0.960	0.060	0.650	2500
German	2	24	1000	0.026	0.664	0.992	0.794	0.664	42
Glass	6	9	214	0.474	0.013	0.963	0.452	0.734	24
Heart	2	13	270	0.041	0.196	0.985	0.838	0.567	21
Liver	2	6	345	0.017	0.073	0.968	0.853	0.623	58
Phoneme	2	5	5404	0.082	0.271	0.878	0.067	0.912	1081
Satimage	6	36	6435	2.060	0.000	0.883	0.215	0.909	179
Segment	7	19	2310	0.938	0.000	0.583	0.072	0.967	122
Sonar	2	60	208	0.029	0.000	0.947	0.544	0.827	3
Texture	11	40	5500	3.614	0.000	0.726	0.119	0.992	138
Vehicle	4	18	846	0.259	0.169	0.968	0.273	0.653	47
Vowel	11	10	528	0.536	0.482	0.962	0.129	0.991	53
Waveform	3	21	4999	0.410	0.007	0.997	0.769	0.780	238
Wine	3	13	178	2.362	0.000	0.315	0.018	0.770	14

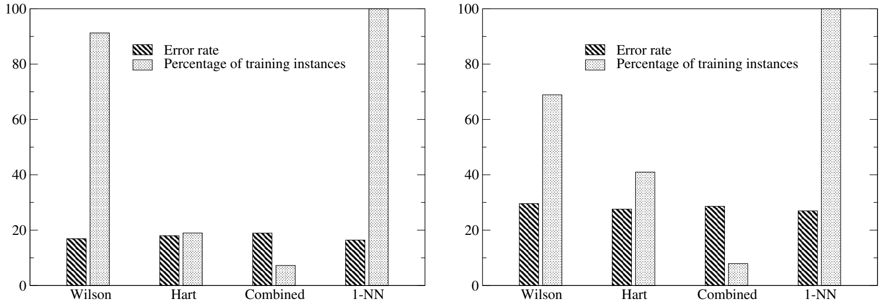
As can be seen in Table 2, Wilson's editing outperforms the 1-NN rule when F1 is under 0.410 (that is, when regions from different classes are strongly overlapped). Consequently, for a particular problem, one could decide to apply an editing to the original TS or directly to employ the plain 1-NN classifier according to the value of F1. For data sets with no (or weak) overlapping (in Table 2, those with F1 $>$ 0.410), the use of an editing can become even harmful in terms of error rate: it seems that editing removes some instances that are defining the decision boundary and therefore, this produces a certain change in the form of such a boundary. Another important result in Table 2 refers to the percentage of training instances given by Hart's condensing: in general, the reductions in TS size for databases with high overlap are lower than those in the case of data sets with weak overlapping.

From the results included in Table 2, it is possible to distinguish between two situations. First, for domains in which the classes are strongly overlapped, one has to employ an editing algorithm in order to obtain a lower error rate (in these cases, benefits in size reduction and classification time are also obtained). Second, for databases with weak overlapping (i.e., F1 is high enough), in which error rate given by the 1-NN rule can be even lower than that achieved with an editing, one should still decide when to apply a PS scheme (reducing time and storage needs) and when to directly use the 1-NN classifier without any preprocessing. In many problems, differences in error rate are not statistically significant (for example, in Satimage database, the error rates for Wilson's editing and 1-NN rule are 16.90% and 16.40%, respectively) and in such cases, savings in memory requirements and classification times can become the key issues for deciding which method to employ.

Table 2. 1-NN error rate and percentage of training instances (in brackets), sorted by F1 (values in italics indicate the lowest error rate for each database).

	F1	Wilson		Hart		Combined		1-NN	
Gauss	0.000	*30.24*	(68.93)	35.86	(54.07)	30.76	(8.08)	35.06	(100.00)
Liver	0.017	*32.18*	(66.59)	37.68	(59.13)	34.17	(17.46)	34.50	(100.00)
German	0.026	*30.60*	(68.10)	38.50	(53.45)	30.49	(10.73)	34.69	(100.00)
Sonar	0.029	43.03	(82.04)	50.40	(34.49)	*40.42*	(17.25)	47.89	(100.00)
Diabetes	0.032	*27.21*	(71.66)	35.29	(51.47)	27.34	(10.78)	32.68	(100.00)
Heart	0.041	*32.61*	(58.06)	42.14	(59.54)	35.20	(13.52)	41.83	(100.00)
Phoneme	0.082	*26.43*	(89.42)	34.07	(21.55)	28.17	(9.28)	29.74	(100.00)
Clouds	0.245	*11.52*	(88.06)	17.28	(27.25)	11.80	(4.07)	15.34	(100.00)
Vehicle	0.259	36.54	(64.15)	36.76	(53.43)	37.36	(18.65)	*35.59*	(100.00)
Waveform	0.410	*18.96*	(82.01)	26.01	(38.96)	21.84	(17.09)	22.04	(100.00)
Glass	0.474	32.37	(70.69)	31.35	(47.01)	32.74	(18.74)	*28.60*	(100.00)
Vowel	0.536	5.23	(96.69)	4.57	(23.40)	8.51	(21.96)	*2.10*	(100.00)
Segment	0.938	5.28	(96.09)	5.88	(13.73)	6.88	(9.90)	*3.72*	(100.00)
Cancer	1.315	*4.25*	(95.54)	6.43	(11.44)	4.39	(3.00)	4.54	(100.00)
Satimage	2.060	16.90	(91.24)	17.94	(18.96)	18.93	(7.23)	*16.40*	(100.00)
Wine	2.362	29.57	(68.89)	27.59	(40.97)	28.60	(7.92)	*26.95*	(100.00)
Texture	3.614	1.22	(98.97)	2.91	(8.01)	2.86	(6.86)	*1.04*	(100.00)

Fig. 1 illustrates the situation just described, comparing the error rate and the percentage of training instances for two databases with a high value of F1. For the Satimage database, differences in error rate are not statistically significant but, in terms of percentage of training instances, the combined approach is clearly the best option: it stores only 7.23% of the original samples and provides an error rate approximately 2% higher than the plain 1-NN rule with the whole TS (100% of instances). Results for the Wine database are similar to those of the Satimage domain, although now differences in error rate are more important when comparing Wilson's editing and 1-NN classifier.



(a) Satimage

(b) Wine

Fig. 1. Comparing error rate and percentage of the original instances retained by each method for several databases with high F1.

As a conclusion, for these cases with high F1, one has to decide whether it is more important to achieve the lowest error rate but without any reduction in storage or to attain a moderate error rate with important savings in memory requirements (and also, in classification times).

Despite F1 results in the complexity measure with the highest discrimination power in the specific framework of PS, it is to be noted that other measures can become especially useful for other different tasks. For example, F2 and F3 (conveniently adapted) could be particularly interesting in the case of feature selection because they could be used as objective functions to pick subsets of relevant features. On the hand, other measures constitute a complement in the analysis of certain problems. In this sense, T2 can help to understand why the plain 1-NN classifier does not perform well in problems with weak overlapping. For example, the 1-NN error rate in Wine database, which corresponds to a problem with almost no overlapping (F1 = 2.362), is high enough (26.95%); this can be explained by the fact that there exists a very small number of training instances per dimension (T2 = 14).

5 Concluding Remarks and Further Extensions

The primary goal of this paper has been to analyze the relation between data complexity and efficiency for the 1-NN classification. More specifically, we have investigated on the utility of a set of complexity measures as a tool to predict whether or not the application of some PS algorithm results appropriate in a particular problem.

After testing different data complexity measures, from the experiments carried out over 17 databases, it seems that F1 can become especially useful to distinguish between the situations in which a PS technique is clearly needed and those in which a more extensive study has to be considered. While in the former case the PS approach achieves the lowest error rate and some savings in memory storage, for the later it is not clear the significance of gains in error rate and therefore, other measures should be employed because even the application of a method with a higher error rate could be justified according to other benefits in computational requirements.

It is worth noting that for those situations in which PS degrades the 1-NN accuracy, one could still reduce the (high) computing time associated to the plain 1-NN rule by means of *fast search* algorithms [2]. However, it is known that fast search algorithms can lessen the number of computations during classification but they still maintain the memory requirements.

Future work is mainly addressed to extend the data complexity measures employed in the same framework of the present paper, trying to better characterize the conditions for an appropriate use of PS techniques. A larger number of PS algorithms, both from selection and abstraction perspectives, has also to be tested in order to understand the relation between data complexity and performance of the 1-NN classifier. Finally, a more exhaustive study will help to categorize the use of several complexity measures for different pattern recognition tasks.

Acknowledgments

This has work been supported in part by grants TIC2003-08496 from the Spanish CICYT, GV04A/705 from Generalitat Valenciana, and P1-1B2004-08 from Fundació Caixa Castelló-Bancaixa.

References

1. Chang, C.-L.: Finding prototypes for nearest neighbor classifiers, IEEE Trans. on Computers 23 (1974) 1179-1184.
2. Chavez, E., Navarro, G., Baeza-Yates, R.A., Marroquin, J.L.: Searching in metric spaces, ACM Computing Surveys 33 (2001) 273-321.
3. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification, IEEE Trans. on Information Theory 13 (1967) 21-27.
4. Dasarathy, B.V.: Minimal consistent subset (MCS) identification for optimal nearest neighbor decision systems design, IEEE Trans. on Systems, Man, and Cybernetics 24 (1994) 511-517.
5. Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach, Prentice Hall, Englewood Cliffs, NJ (1982).
6. Hart, P.E.: The condensed nearest neighbor rule, IEEE Trans. on Information Theory 14 (1968) 515-516.
7. Ho, T.-K., Basu, M.: Complexity measures of supervised classification problems, IEEE Trans. on Pattern Analysis and Machine Intelligence 24 (2002) 289-300.
8. Bernardo, E., Ho, T.-K.: On classifier domain of competence, In: Proc. 17th. Int. Conf. on Pattern Recognition 1, Cambridge, UK (2004) 136-139.
9. Kim, S.-W., Oommen, B.J.: Enhancing prototype reduction schemes with LVQ3-type algorithms, Pattern Recognition 36 (2003) 1083-1093.
10. Kuncheva, L.I.: Editing for the k-nearest neighbors rule by a genetic algorithm, Pattern Recognition Letters 16 (1995) 809-814.
11. Mollineda, R.A., Ferri, F.J., Vidal, E.: An efficient prototype merging strategy for the condensed 1-NN rule through class-conditional hierarchical clustering, Pattern Recognition 35 (2002) 2771-2782.
12. Ritter, G.L., Woodruff, H.B., Lowry, S.R., Isenhour, T.L.: An algorithm for a selective nearest neighbour decision rule, IEEE Trans. on Information Theory 21 (1975) 665-669.
13. Tomek, I.: An experiment with the edited nearest neighbor rule, IEEE Trans. on Systems, Man and Cybernetics 6 (1976) 448-452.
14. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data sets, IEEE Trans. on Systems, Man and Cybernetics 2 (1972) 408-421.
15. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms, Machine Learning 38 (2000) 257-286.