Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 36 (2009) 3587–3595
Expert Systems with Applications
www.elsevier.com/locate/eswa

Pseudo nearest neighbor rule for pattern classification

Yong Zeng *, Yupu Yang, Liang Zhao

Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China

Abstract

In this paper, we propose a new pseudo nearest neighbor classification rule (PNNR). It is different from the previous nearest neighbor rule (NNR), this new rule utilizes the distance weighted local learning in each class to get a new nearest neighbor of the unlabeled pattern-pseudo nearest neighbor (PNN), and then assigns the label associated with the PNN for the unlabeled pattern using the NNR. The proposed PNNR is compared with the k-NNR, distance weighted k-NNR, and the local mean-based nonparametric classification [Mitani, Y., & Hamamoto, Y. (2006). A local mean-based nonparametric classifier. *Pattern Recognition Letters,* 27, 1151–1159] in terms of the classification accuracy on the unknown patterns. Experimental results confirm the validity of this new classification rule even in practical situations.

Keywords: The k-nearest neighbor classification rule (k-NNR); Pseudo nearest neighbor classification rule (PNNR); Distance weighted k-nearest neighbor rule; The local mean-based learning; Pseudo nearest neighbor (PNN)

1. Introduction

In pattern recognition, and in situations where a concise representation of the underlying probability density distributions is difficult to obtain, the use of nonparametric techniques to classify an unknown pattern as belonging to one of a set of M classes is necessary. One of the most popular nonparametric techniques is the k-nearest neighbor classification rule (k-NNR). According to this rule, an unclassified sample is assigned to the class represented by a majority of its k-nearest neighbors in the training set. Cover and Hart have shown that, as the number N of samples and the number k of neighbors both tend to infinity in such a manner that $k/N \to 0$, the error rate of the k-NNR approaches the optimal Bayes error rate (Cover & Hart, 1967). The k-NNR generally achieves good results when the available number of prototypes is large, relative to the intrinsic dimensionality of the data involved. However, in most real situations, the number of available prototypes is usually very small, which often leads to dramatic degradations of k-nearest neighbor (k-NN) classification accuracy. That clearly explains why there are many variants of the k-NNR to improve the k-NN classification performance in small data set situations. Most of these variants are more concerned with the choice of the distance measures or metrics (Hastie & Tibshirani, 1996; Kohavi, Langley, & Yung, 1997; Paredes & Vidal, 2006; Ricci & Avesani, 1999; Wilson & Martinez, 1996), or prototype editing techniques (Ferri, Albert, & Vidal, 1999; Paredes & Vidal, 2000; Penrod & Wagner, 1977). Instead of focusing on distance training and prototype editing, here we concentrate on finding a new distance weighted local learning method to improve the classification performance of the k-NNR.

The k-NNR implicitly assumes the k-nearest neighbors of an unlabeled point x to be contained in a region of relatively small volume, so that sufficiently good resolution in the estimates of the different conditional densities can be obtained. The radius of the region is determined by the distance of the k-furthest neighbor. If k is very small and the nearest neighbors are not nearby due to data sparseness, or the nearest neighbor classes are unreliable due to noise, the "local" estimate tends to be very poor. This has two consequences. The first is that the k-NN classifiers usually

* Corresponding author. Tel.: +86 21 3420 4261; fax: +86 21 3420 4427.
E-mail address: zeng_yong@sjtu.edu.cn (Y. Zeng).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.02.003

suffer from the existing outliers (Fukunaga, 1990). The second is that the k neighbors are implicitly assumed to have an equal weight in decision in the standard k-NNR, regardless of their distances to the pattern x to be classified. To overcome the influence of the outliers, Mitani and Hamamoto have proposed a local mean-based nonparametric classifier (Mitani & Hamamoto, 2006). With regard to the second consequence, Dudani proposed to give different weights to the k neighbors based on their distances to the unlabeled pattern x, with close neighbors having greater weights (Dudani, 1976).

Motivating by the distance weighted k-nearest neighbor rule and the nearest local mean learning method, we proposed a pseudo nearest neighbor rule (PNNR). In this new rule, instead of the original nearest neighbor of unclassified pattern x, a new nearest neighbor (we shall name it the pseudo nearest neighbor, PNN) is decided according to the value of the weighted sum of distances of the local k-nearest neighbors of unclassified pattern in each class of the prototypes, and the unlabeled pattern is assigned to the class for which the pseudo nearest neighbor (PNN) belongs to. The experimental results show the effectiveness of the proposed PNNR in such practical situations.

This paper is organized as follows: we present the PNNR in Section 2. In Section 3, we present an algorithmic view of the PNNR. The difference between the PNNR and the k-NNR is given in Section 4. The experiment results both on six benchmark data sets and artificial data sets, and their analysis are reported in Section 5. Section 6 concludes the paper and presents future work.

2. The pseudo nearest neighbor classification rule

2.1. Related work

2.1.1. The k-NN rule and distance weighted k-NN rule

The k-nearest neighbor rule (k-NNR) (Fix & Hodges, 1951) is well known in the pattern recognition literature. According to this rule, an unclassified pattern is assigned to the class represented by a majority of its k-nearest neighbors (k-NNs). This rule is nowadays usually called the voting k-NNR. Cover and Hart have shown that as the number N of patterns and k both tend to infinity in such a manner that $k/N \to 0$ the error rate of the k-NNR approaches the optimal Bayes error rate (Cover & Hart, 1967). In fact, the NNR is the simplest version of the k-NNR for $k = 1$.

In voting k-NNR, the k neighbors are implicitly assumed to have an equal weight in decision, regardless of their distances to the pattern x to be classified. It is intuitively appealing to give different weights to the k neighbors based on their distances to unclassified point x, with closer neighbors having greater weights.

Let d be a distance measure, and $x_1, x_2, \ldots, x_k$ be the k-NNs of x arranged in increasing order of $d(x, x_i)$. So x_1 is the first nearest neighbor of x. Dudani (1976) proposes to assign to the i-th nearest neighbor, x_i, a weight w_i defined as

$$w_i = \begin{cases} \frac{d(x_k,x)-d(x_i,x)}{d(x_k,x)-d(x_1,x)} & \text{if } d(x_k,x) \neq d(x_1,x), \\ 1 & \text{if } d(x_k,x) = d(x_1,x). \end{cases} \quad (1)$$

Pattern x is assigned to the class for which the weights of the representatives among the k-NNs sum to the greatest value. This rule has been shown by Dudani to yield lower error rates than those obtained using the voting k-NNR. However some other researchers have arrived at less optimistic conclusion (Dasarathy, 1991; Morin & Raeside, 1981). Denoeux provides an excellent and detailed review of distance weighted k-NNR (Denoeux, 1995).

2.1.2. The local mean learning

To overcome the influence of the existing outliers in the nonparametric classifiers, Mitani and Hamamoto have proposed a local mean learning method (Mitani & Hamamoto, 2006). Let $X_j = \left\{x_j^i | i = 1, \ldots, n_j\right\}$ be a training sample set from the class ω_j, where n_j is the number of training samples from the class ω_j. First, to compute the local mean vector y_j of the j-th class prototypes using the r nearest neighbor training samples $\left\{x_j^{(1)}, \ldots, x_j^{(r)}\right\}$ in the j-th class prototypes:

$$y_j = \frac{1}{r}\sum_{i=1}^{r} x_j^{(i)}. \quad (2)$$

Classify the unlabeled pattern x into class ω_c if

$$(x - y_c)^{\mathrm{T}}(x - y_c) = \min(x - y_j)^{\mathrm{T}}(x - y_j). \quad (3)$$

2.2. The proposed pseudo nearest neighbor rule

For the N available labeled prototypes, let $n_1, \ldots, n_M$ is the number of them belonging to the class $1, 2, \ldots, M$, respectively. Let $x_j^{(1)}, \ldots, x_j^{(k)}$ denote the k-nearest neighbors in the j-th class prototypes of unlabeled pattern x, and $d_j^{(1)}, \ldots, d_j^{(k)}$ be the corresponding distances arranged in increasing order. Motivated by the distance weighted k-NN rule: to give different weights to the k neighbors based on their distances to x, with closer neighbors having greater weights, however, different from the previous distance weighted k-NN rule, we assign to the i-th nearest neighbor in the j-th class prototypes of unlabeled pattern x, $x_j^{(i)}$, a weight w_i defined as:

$$w_i = \frac{1}{i} \quad i = 1, \ldots, k. \quad (4)$$

It is obvious that w_i decreases with the increasing of i, the less w_i is, and the corresponding pattern $x_j^{(i)}$ has less impact to the classification of unlabeled pattern.

Motivated by while different from the local mean learning method as mentioned above, we let y_j be the weighted sum of distances of the k-nearest neighbors of the pattern x in the j-th class, then

$$y_j = w_1 \times d_j^{(1)} + w_2 \times d_j^{(2)} + \cdots + w_k \times d_j^{(k)} \quad j = 1, \ldots, M. \tag{5}$$

We define the pseudo nearest neighbor of the unlabeled pattern x by

$$x_{\mathrm{PNN}} = \arg\min\{y_j\} \quad j = 1, 2, \ldots, M. \tag{6}$$

We propose the pseudo nearest neighbor classification rule (PNNR) as follow: a new pair (x, θ) is given, the pseudo nearest neighbor classification rule decides x belongs to category θ of its pseudo nearest neighbor (PNN)x_{PNN}. In case of ties for the pseudo nearest neighbors, we can choose randomly in those pseudo nearest neighbors.

3. Difference between the PNNR and the standard *k*-NNR

Although the proposed PNNR is similar to the standard NNR and k-NNR, it assigns the label for the unlabeled pattern in a distinct manner.

3.1. Difference between the PNNR and the NNR

Being an extension of the nearest neighbor rule, the pseudo nearest neighbor classification rule for classifying the unclassified pattern is to assign it the label associated with the pseudo nearest neighbor. In fact, the pseudo nearest neighbor is not a true nearest neighbor, but is the associated class to which the unclassified pattern belongs. The difference between the two rules is illustrated in Fig. 1. The unlabeled pattern (hollow point) should be assigned to class 2 using the nearest neighbor rule; however, with the pseudo nearest neighbor classification rule, the test pattern is assigned to class 1. Where $k = 3$, the pseudo nearest neighbor is decided by the three nearest neighbors of the test pattern in class 1.

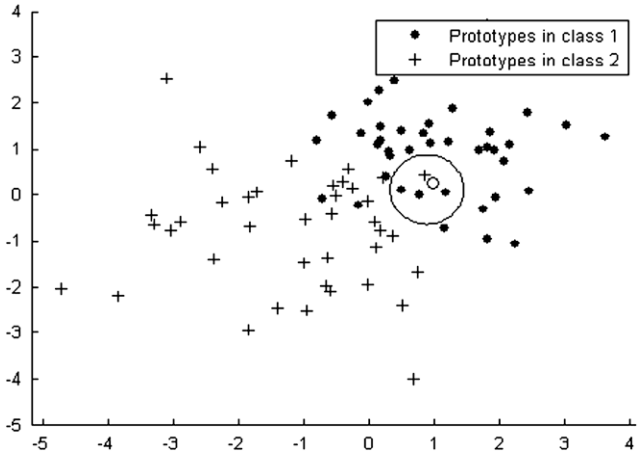

Fig. 1. The difference between the PNNR and the NNR. In the NNR, the nearest neighbor of the unlabeled pattern (hollow point) is cross point in the class 2. In this $k = 3$ case, with the generalized nearest neighbor rule, the pseudo nearest neighbor which is decided by three black points represents the class 1.

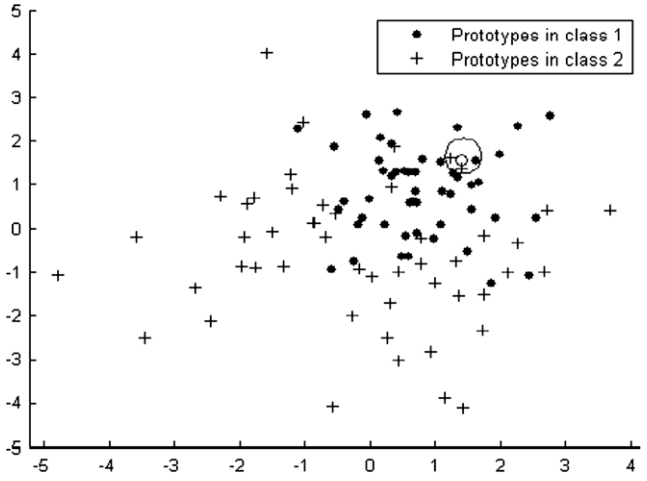

Fig. 2. Difference between the PNNR and the k-NNR. For $k = 3$, the k-NNR assigns the unclassified pattern (hollow point) the label class 2 but the GNNR assigns it, the class 1. In this special case, it is reasonable that the unlabeled pattern is labeled the class 1.

3.2. Difference between the PNNR and the k-NNR

In a sense, the PNNR is similar to the k-NNR. The common for two rules is, both two rules must get the nearest neighbors of the unlabeled pattern; the difference is, for the latter, the nearest neighbors is with respect to all prototypes in all classes, nevertheless, for the former, with respect to the prototypes in a certain class. Sometimes, the different classification with respect to the same unlabeled pattern results from their difference; it is shown in Fig. 2.

4. An algorithmic view

We summarize the generalized nearest neighbor rule in the algorithmic form. Assuming there are the N labeled prototypes available, let n_j be the number of the prototypes belonging to the j-th class, $j = 1, \cdots, M$. This then gives the algorithm for the parameter k:

1. for ($j = 1, \ldots, M$)
2. for ($i = 1, \ldots, k$)
3. calculate the distance between the k-nearest neighbors of the test pattern in the jth class prototypes and the test pattern $d_j^1, \ldots, d_j^k$, $d_j^i = ||x_j^i - x||_2$
4. end
5. get the arranged distances in creasing order, $d_j^{(1)}, \ldots, d_j^{(k)}$
6. $y_j = \sum_{i=1}^{k} w_i * d_j^{(i)}$
7. end
8. get the pseudo nearest neighbor $x_{\mathrm{PNN}} = \arg\min\{y_j\}$
9. if ($x_{\mathrm{PNN}} = y_r$, $r \in \{1, \ldots, M\}$)
10. then (the test pattern is assigned to the class r)
11. else if ($x_{\mathrm{PNN}} \in \{y_r, \ldots, y_{r+s}\}$, $r, \ldots, (r+s) \in \{1, \ldots, M\}$)
12. then (choose one randomly in the collection $\{r, \ldots, r+s\}$ and assign it to the test pattern as its label)

13. end
14. end

The parameter k is optimized by the cross-validation (CV) approach and we adopt the same approach which has been described in (Mitani & Hamamoto, 2006).

5. Experimental results

We tested the performance of the proposed PNNR, and compared it with the classical k-NNR, the distance weighted k-NNR (Dudani, 1976) and the local mean-based learning method (Mitani & Hamamoto, 2006). For simplicity, we will refer to the previous four classification methods as PNN, k-NN, WNN and LM, respectively later. In order to test the performance of the different classification methods, we used two different types of data sets. In the first, six artificial data sets were used. In the second, six standard data sets of which only data sets with numeric features were selected from the UCI (Merz & Murphy, 1996) were considered. The distance metric used is the Euclidean distance. The classification function adopted for the k-NNR is function *knnclassify* which can be found in Bioinformatics Toolbox in *MATLAB 2006a* and the rule being specified is majority rule with nearest point tie-break. Performance is evaluated by looking at three figures of merit: (a) classification performance with varying the neighborhood size k, (b) classification performance with varying feature space dimension, and (c) classification performance with varying the number of the training set.

Table 1
Some characteristics of the data sets used

Database	Features	Samples	Classes	Test set
Letter	16	20,000	26	4000 cases
Pen	16	10,992	10	3498 cases
Thyroid	21	7200	3	3428 cases
Pima	8	768	2	–
Wine	13	178	3	–
Iris	4	150	3	–

5.1. Experimental data

5.1.1. Artificial data sets

Four of the artificial data sets we adopted are same to that Mitani and Hamamoto adopted (Mitani & Hamamoto, 2006). The class-conditional probability density function in each of the classes is singular normal density function in these four data sets. The other two artificial data sets are different from the previous four, for these two artificial data sets, the class-conditional probability density function in each of the classes is a linear combination of multiple normal density functions. We named the former four artificial

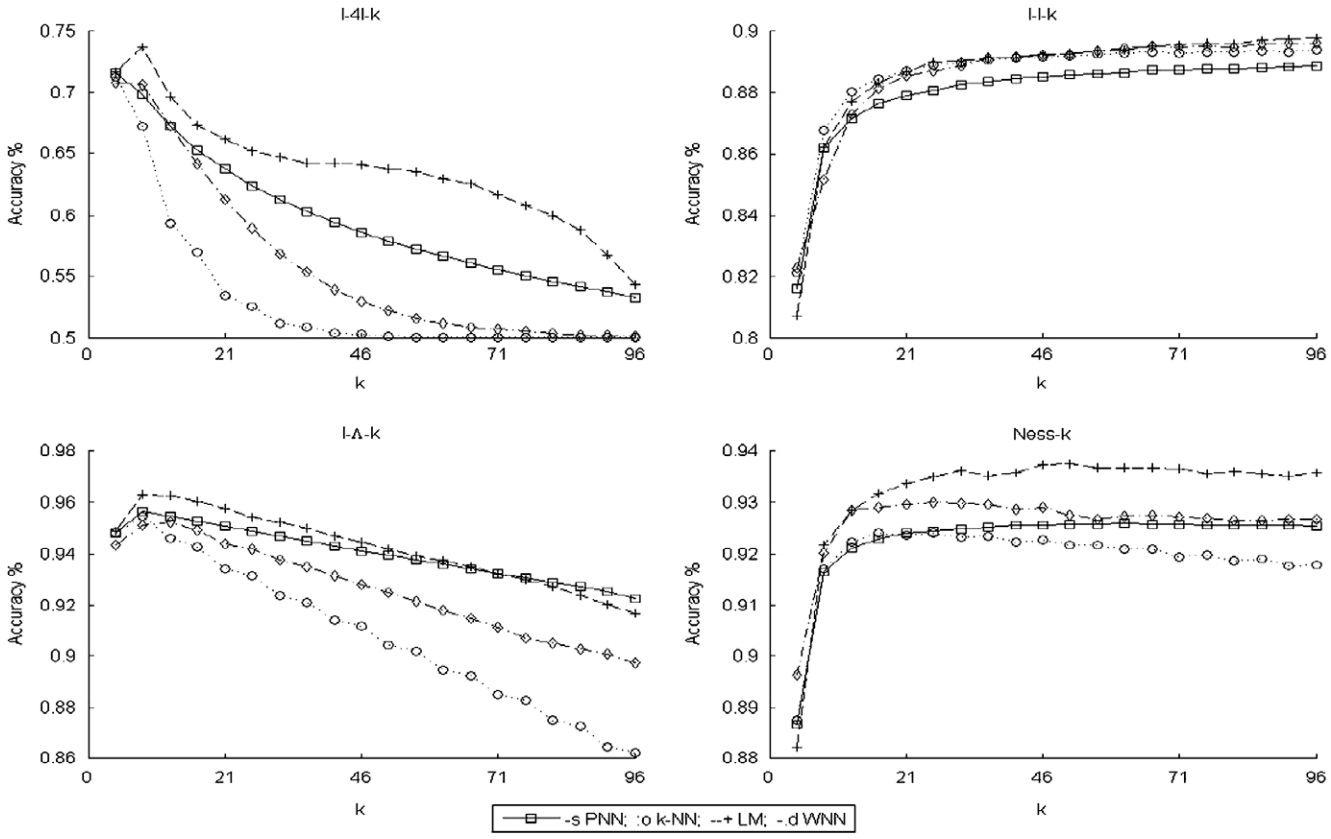

Fig. 3. Accuracy results with varying neighborhood size.

data sets as the singular model data sets, and named the latter as the mixture model data sets. For these six artificial data sets, we can easily independently generate two sets of data, the training and test sets, from the same model by a computer. We can benefit from using artificial data. We can control the number of the available samples. Therefore, we chose the training samples size which directly influences difficulties of the pattern recognition problem, so as to set the training sample size situations in the experiments. On the other hand, we also can set the test samples size. In order to reduce the influence of the test sample size on the error rate, in the experiments, we used 1000 test samples for each class.

5.1.1.1. Singular model data sets. We briefly describe the singular model data sets. We used the I-Λ, I-4I, I-I (Fukunaga, 1990) and **Ness** data sets (Van Ness, 1980). In these artificial data sets, μ_i is the mean vector and $\sum_i$ is the covariance matrix from class ω_i.

I-Λ data set: I-Λ data set consists of 8-dimensional Gaussian data.

$$\mu_1 = 0, \quad \mu_2 = [3.86, 3.10, 0.84, 0.84, 1.64, 1.08, 0.26, 0.01]^{\mathrm{T}},$$
$$\Sigma_1 = I_8,$$
$$\Sigma_2 = \mathrm{diag}[8.41, 12.06, 0.12, 0.22, 1.49, 1.77, 0.35, 2.73],$$

where I_k and diag[.] denote the $k \times k$ identity and diagonal matrices, respectively.

I-4I data set: The I-4I data set consists of 8-dimensional Gaussian data.

$$\mu_1 = \mu_2 = 0,$$
$$\Sigma_1 = I_8, \quad \Sigma_2 = 4I_8.$$

I-I data set: The I-I data set consists of p-dimensional Gaussian data. In this data, the dimensionality p can be controlled.

$$\mu_1 = 0, \quad \mu_2 = [2.56, 0, \ldots, 0]^{\mathrm{T}},$$
$$\Sigma_1 = \Sigma_2 = I_p.$$

Ness data set: The Ness data set consists of p-dimensional Gaussian data.

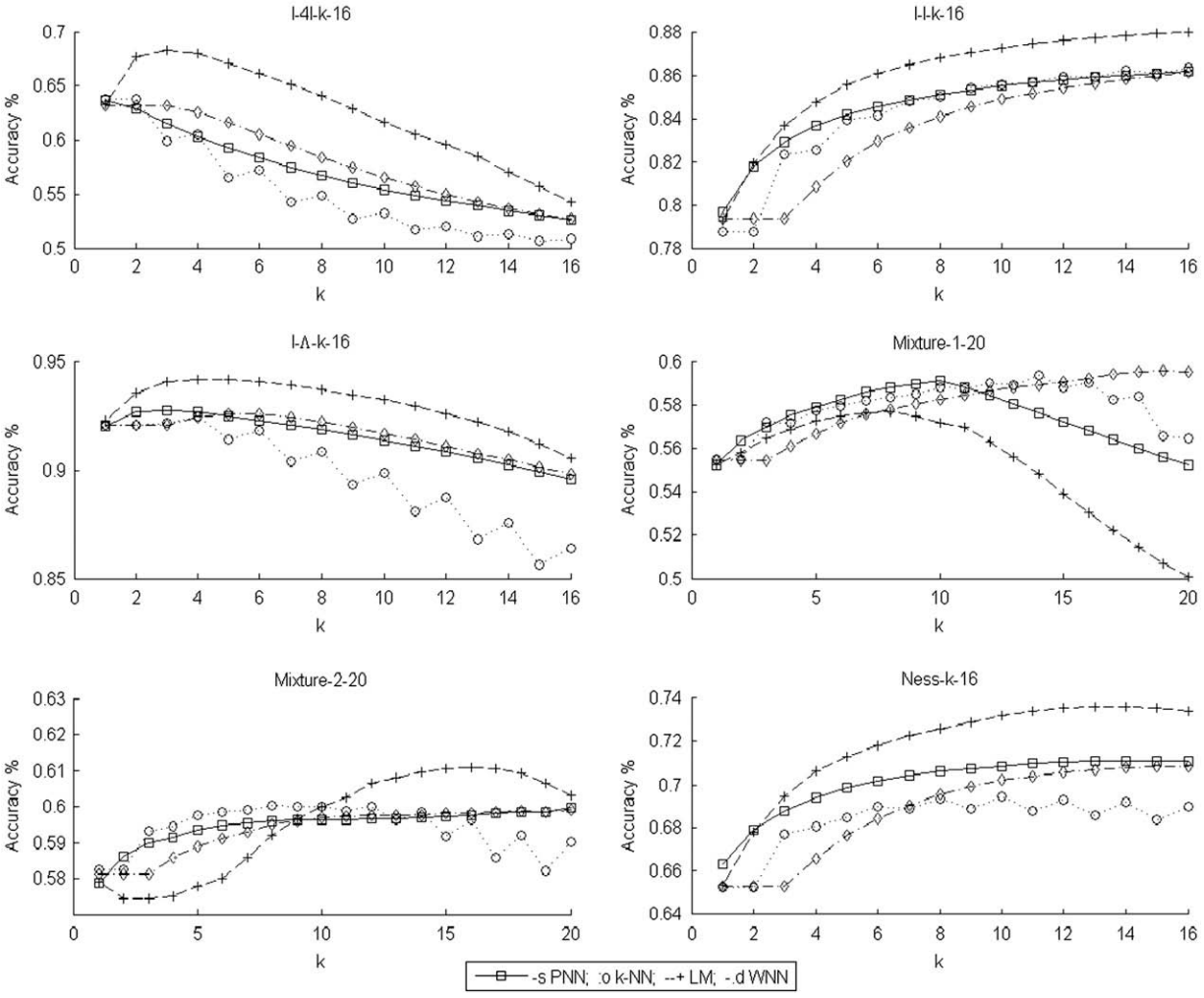

Fig. 4. Accuracy results with varying neighborhood size.

$$\mu_1 = 0, \quad \mu_2 = [\Delta/2, 0, \ldots, 0, \Delta/2]^{\mathrm{T}},$$
$$\Sigma_1 = I_p, \qquad \Sigma_2 = \begin{bmatrix} I_{p/2} & 0 \\ 0 & \frac{1}{2} I_{p/2} \end{bmatrix}.$$

In the experiments, we used $\Delta = 2$, 4, and 6, and also varied $p = 2$–50.

5.1.1.2. Mixture model data sets. The mixture model data sets used are described briefly as follows:

Mixture-1: the class-conditional probability density function in each of the classes is a linear combination of two p-dimensional normal density functions in the Mixture-1 data set.

$$7f^1(x) = 0.5N(1.8, 4\Sigma) + 0.5N(3, 6\Sigma),$$
$$f^2(x) = 0.5N(2, 4\Sigma) + 0.5N(2.8, 6\Sigma),$$

where $f^i(x)$ is the conditional pdf of ith class samples and a priori probabilities are equal.

Mixture-2: the class-conditional probability density function in each of the classes is a mixture of three p-dimensional normal density functions in the Mixture-2 data set.

$$f^1(x) = 0.3N(1, \Sigma) + 0.3N(3, 9\Sigma) + 0.4N(2, 5\Sigma),$$
$$f^2(x) = 0.3N(1.5, 4\Sigma) + 0.3N(2.5, 4\Sigma) + 0.4N(3.5, 5\Sigma),$$

where $f^i(x)$ is the conditional pdf of ith class samples and a priori probabilities are equal.

For the artificial data sets, the classification performance of four classification methods was examined in terms of the classification accuracy. The number of trials was 100. Fresh samples were artificially generated by a computer on each trial. The 100 classification results were averaged.

5.1.2. UCI standard data sets

Some characteristics of the UCI standard data sets used are described in Table 1. For the data sets Letter, Pen and Thyroid, the training sets and test sets are assigned in advance in earlier work on the same data sets.

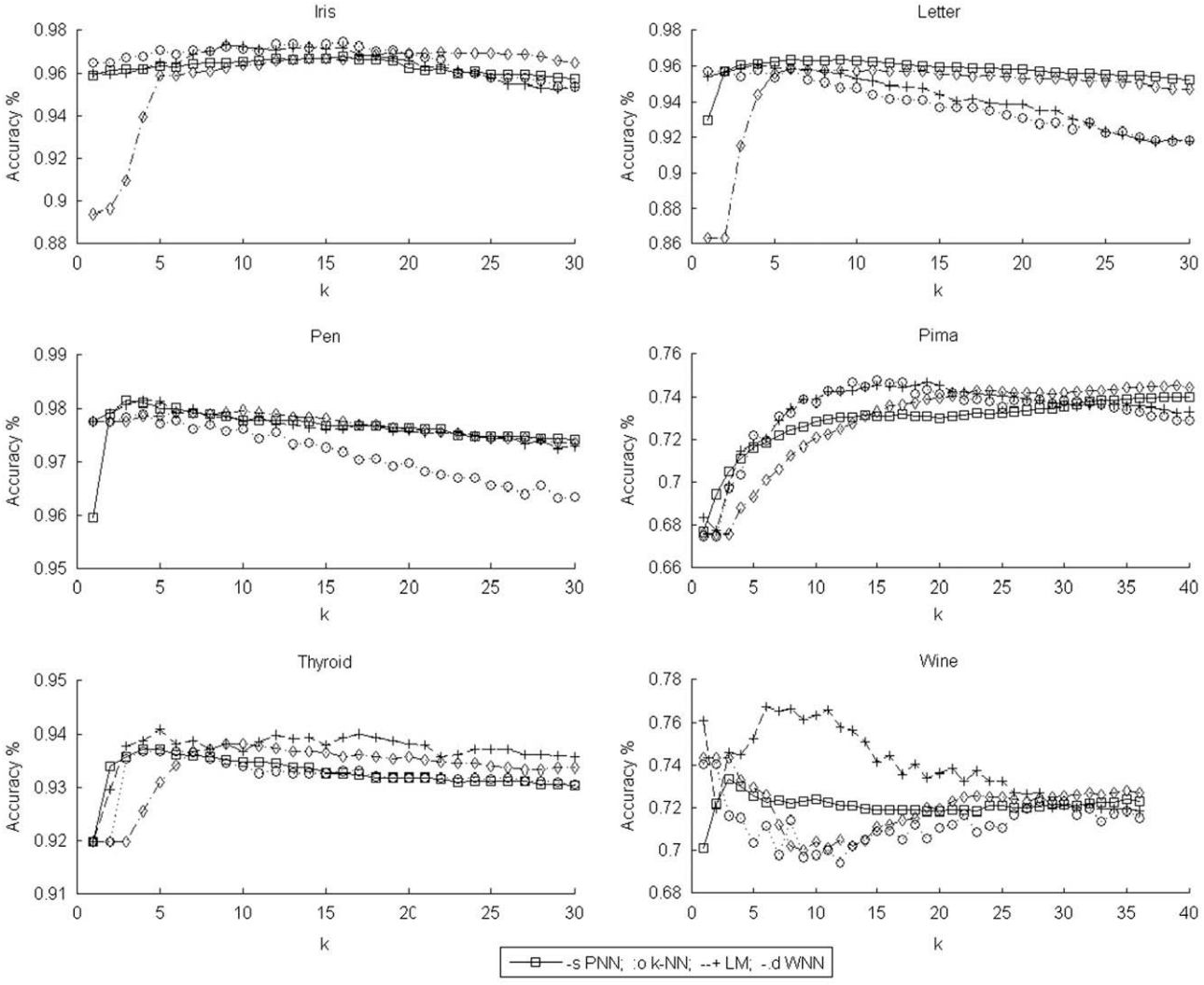

Fig. 5. Accuracy results with varying neighborhood size.

For the other three data sets, we chose a training set that contains 80% of the data and the test set contains the other 20% of the data set. We repeated the experiments 100 times, each time we picked randomly the training and test sets, and we report as result the average over these 100 repetitions.

5.2. Experimental results

5.2.1. Results with varying the neighborhood size k

In all the data sets, we tested the four classification methods mentioned before with different neighborhood size k. As to k-NN and WNN, the neighborhood size is for all training samples of unlabeled sample, while with respect to PNN and LM, is for each class samples of the unlabeled sample. Figs. 3–5 show classification accuracy rates with different neighborhood size. In the experiments about Fig. 3, also assuming the each class has an equal prior probability, for these singular model data sets, we set $p = 8$, $N_1 = N_2 = 100$ and $\Delta = 2$ (for Ness data set). In the experiments about Fig. 4, assuming the each class has an equal prior probability, i.e.,$P(\omega_1) = P(\omega_2)$, for singular model data sets, we set $p = 8$, $N_1 = N_2 = 16$ and $\Delta = 2$ (for Ness data set); for mixture model data sets, we set $p = 6$, $N_1 = N_2 = 20$. For the real UCI data, the experiments were conducted as mentioned above in the foregoing Section 5.1.2.

From Figs. 3–5, we can see that the classification performance of PNN is better than which of k-NN and WNN at most cases, and the classification performance of PNN is also better than the classification performance of LM in the mixture model data sets and large samples situation. Experimental results suggest that the optimal value of neighbor size which maximizes classification accuracy may exist for the given data set.

5.2.2. Results with varying the feature space dimension

Next, we explore the classification performance with varying the feature space dimension. In the experiments related to Fig. 6, we assume that the each class has an equal prior probability, i.e.,$P(\omega_1) = P(\omega_2)$, and we set $N_1 = N_2 = 10$ and $\Delta = 2$, 4, 6 for Ness-1, Ness-2, Ness-3,

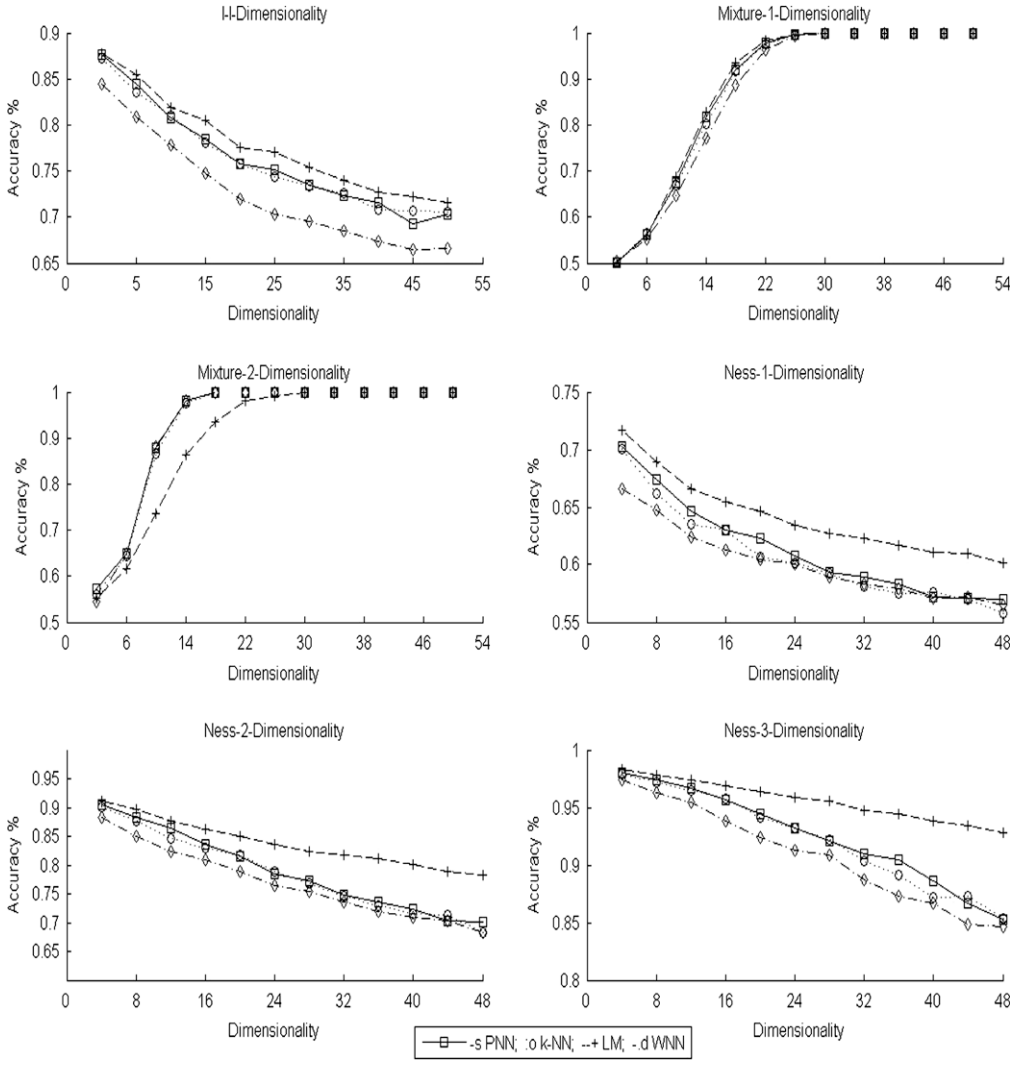

Fig. 6. Accuracy results with varying feature space dimension.

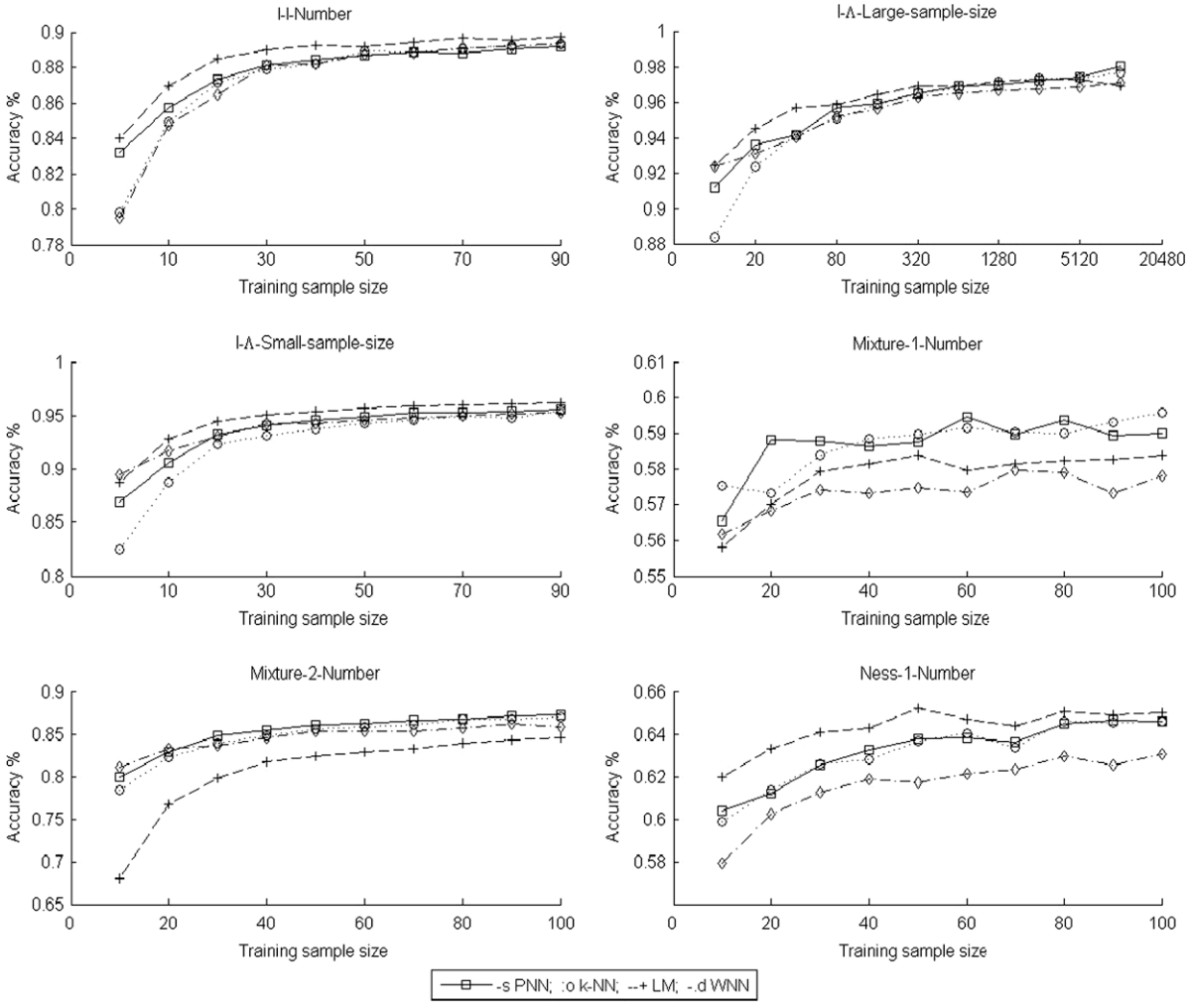

Fig. 7. Accuracy results with varying training sample size.

respectively. For I-I data set, we generated sample with 2, 5, 10, 15, 20, 25, 30, 35, 40, 45, and fifty-dimensional feature spaces. For Ness-1, Ness-2 and Ness-3 data sets, we generated sample with 4, 8, 12, 16, 24, 28, 32, 36, 40, 44, and forty-eight-dimensional feature spaces. For Mixture-1 and Mixture-2 data sets, we generated sample with 2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, and fifty-dimensional feature spaces. Each classification method's parameter i.e. number of the nearest neighbors was trained by leave-one-out cross-validation on the training set. Fig. 6 shows that the classification performance of PNN is better than k-NN and WNN, and is also superior to LM under the mixture model data sets situation.

5.2.3. Results with varying the number of the training set

Finally, we investigated the classification performance with respect to varying the number of the training set. Fig. 7 shows the accuracy as a function of the number of the training set. In the experiments associated with Fig. 7, assuming that the each class has an equal prior probability, i.e.,$P(\omega_1) = P(\omega_2)$, and we set the feature dimensionalities $p = 8$ for Ness-1 ($\Delta = 2$), I-I, I-Λ-Large, I-Λ-Small and Mixture-2 data sets, and assign $p = 6$ for Mixture-1 data set. For I-I and I-Λ-Small data sets, we generated samples with number of 5, 10, 20, 30, 40, 50, 60, 70, 80, and 90, respectively. For Ness-1, Mixture-1 and Mixture-2 data sets, we generated samples with number of 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100, respectively. We also conducted the experiment for I-Λ data set with large samples, and generated samples with number of 10, 20, 40, 80, 160, 320, 640, 1280, 2560, 5120, and 10,240, respectively. In these experiments, we set number of nearest neighbors $k = 5$. Fig. 7 shows that the classification performance of PNN is better than k-NN and WNN, and is also superior to LM under the mixture model data sets situation. Fig. 7 also suggests that the classification accuracy increases with increasing number of training set.

6. Conclusions

In this paper, we propose a new classification approach based on the distance weighted local learning. This new classification approach not only utilizes the information of the nearest neighbor of the unclassified sample point

but also utilizes the information of its $k-1$ neighbors in each class. The experimental results show that this new classification rule usually performs better than the traditional k-nearest neighbor classification rule and the distance weighted nearest neighbor classification rule, and is also superior to the local mean-based nonparametric classification method (Mitani & Hamamoto, 2006) in the large training sample size and mixture model data situations. However, in a small training sample size also singular model data case, the classification accuracy of the proposed classification rule does not outperform the local mean-based nonparametric classification method, this problem should be further explored.

References

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21–27.

Dasarathy, B. V. (Ed.). (1991). *Nearest neighbor norms: NN pattern classification techniques*. IEEE Computer Society Press.

Denoeux, T. (1995). A *k*-nearest neighbor classification rule based on Dempster–Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics, 25*(5), 804–813.

Dudani, S. A. (1976). The distance-weighted *k*-nearest neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics, 6*(4), 325–327.

Ferri, F., Albert, J., & Vidal, E. (1999). Considerations about sample-size sensitivity of a family of edited nearest-neighbor rules. *IEEE Transactions on Systems, Man and Cybernetics, 29*(5), 667–672.

Fix, E., & Hodges, J. L., (1951). Discriminatory analysis nonparametric discrimination: Consistency properties. Technical Report TR4, USAF School of Aviation Medicine, Randolph Field, TX.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). Academic Press.

Hastie, H., & Tibshirani, R. (1996). Discriminant adaptive nearest neighbor classification.. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(6), 607–712.

Kohavi, R., Langley, P., & Yung, Y. (1997). The utility of feature weighting in nearest-neighbor algorithms. In *Proceedings of ninth European conference on machine learning*.

Merz, C., & Murphy, P. (1996). UCI repository of machine learning databases. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

Mitani, Y., & Hamamoto, Y. (2006). A local mean-based nonparametric classifier. *Pattern Recognition Letters, 27*(10), 1151–1159.

Morin, R. L., & Raeside, D. E. (1981). A reappraisal of distance-weighted *k*-nearest-neighbor classification for pattern recognition with missing data. *IEEE Transactions on Systems, Man and Cybernetics, 11*(3), 241–243.

Paredes, R., & Vidal, E. (2000). A class-dependent weighted dissimilarity measure for nearest neighbor classification problems. *Pattern Recognition Letters, 21*(12), 1027–1036.

Paredes, R., & Vidal, E. (2006). Learning prototypes and distances: A prototype reduction technique based on nearest neighbor error minimization. *Pattern Recognition, 39*(2), 180–188.

Penrod, C., & Wagner, T. (1977). Another look at the edited nearest neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics, 7*(2), 92–94.

Ricci, F., & Avesani, P. (1999). Data compression and local metrics for nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 21*(4), 380–384.

Van Ness, J. (1980). On the dominance of nonparametric Bayes rule discriminant algorithms in high dimensions. *Pattern Recognition, 12*(6), 355–368.

Wilson, D. R., & Martinez, T. R. (1996). Value difference metrics for continuously valued attributes. In *Proceedings of international conference on artificial intelligence, expert systems and neural networks* (pp. 11–14).