Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 34 (2008) 866–876
Expert Systems with Applications
www.elsevier.com/locate/eswa

Neighborhood classifiers

Qinghua Hu *, Daren Yu, Zongxia Xie

Harbin Institute of Technology, Harbin 150001, People's Republic of China

Abstract

K nearest neighbor classifier (*K*-NN) is widely discussed and applied in pattern recognition and machine learning, however, as a similar lazy classifier using local information for recognizing a new test, neighborhood classifier, few literatures are reported on. In this paper, we introduce neighborhood rough set model as a uniform framework to understand and implement neighborhood classifiers. This algorithm integrates attribute reduction technique with classification learning. We study the influence of the three norms on attribute reduction and classification, and compare neighborhood classifier with KNN, CART and SVM. The experimental results show that neighborhood-based feature selection algorithm is able to delete most of the redundant and irrelevant features. The classification accuracies based on neighborhood classifier is superior to *K*-NN, CART in original feature spaces and reduced feature subspaces, and a little weaker than SVM.
© 2006 Elsevier Ltd. All rights reserved.

Keywords: Metric space; Neighborhood; Rough set; Reduction; Classifier; Norm

1. Introduction

Giving a set of samples *U*, described with some input variables *C* (also called condition attributes, features) and an output *D* (decision), the task of classification learning is to construct a mapping from the condition to the decision labels based on the set of training samples. One of the most popular learning and classification techniques is the nearest neighbor search, introduced by Fix and Hodges (1951). It has been proven to be a simple and yet powerful recognition algorithm. In 1967, Cover and Hart (1967) showed, under some continuity assumptions on the underlying distributions, that the asymptotic error rate of the 1-NN rule is bounded from above by twice the Bayes error (the error of the best possible rule). What is more, a key feature of this decision rule is that it performs remarkably well considering that no explicit knowledge of the underlying distributions of the data is used. Furthermore, a simple generalization of this method, called *K*-NN-rule, in which a new pattern is classified into the class with the most members present among the *K* nearest neighbors, can be used to obtain good estimates of the Bayes error and its probability of error asymptotically approaches the Bayes error (Duda & Hart, 1973). However, *K*-NN classifiers require computing all the distances between the training set and test samples, it is time-consuming if the available samples are of very great size. Besides, when the number of prototypes in the training set is not large enough, the *K*-NN rule is no longer optimal. This problem becomes more relevant when having few prototypes compared to the intrinsic dimensionality of the feature space. After half century, a wide variety of algorithms have been developed to deal with these problems (Anil, 2006; Fu, Chan, & Cheung, 2000; Fukunaga & Narendra, 1975; Hart, 1968; Kuncheva & Lakhmi, 1999; Kushilevitz, Ostrovsky, & Rabani, 2000; Lindenbaum, Markovitch, & Rusakov, 2004; Short & Fukunaga, 1981; Vidal, 1986; Wilson & Martinez, 2000; Zhou, Yan, & Chen, 2006).

From another viewpoint, some classification algorithms based on neighborhood were proposed, where a new

* Corresponding author. Tel.: +86 451 86413241 252; fax: +86 451 86413241 221.
E-mail address: huqinghua@hcms.hit.edu.cn (Q. Hu).

0957-4174/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2006.10.043

sample is associated with a neighborhood, rather than some nearest neighbors. Owen developed a classifier which uses information from all data points in a neighborhood to classify the point at the center of the neighborhood (Owen, 1984). The neighborhood-based classifier is shown to outperform linear discriminant analysis on some LANDSAT data. Salzberg (1991) proposed a family of learning algorithms based on nested generalized exemplars (NGE), where an exemplar is a single training example, and generalized exemplars is an axis-parallel hyperrectangle that may cover several training examples. Once the generalized exemplars are learned, a test example can be classified by computing the Euclidean distance between the example and each of the generalized exemplars. If an example is contained in a generalized exemplar, the distance to that generalized exemplar is zero. The class of the nearest generalized exemplar is output as the predicted class of the test example. Wettschereck and Dieterich (1995) compared NGE with *K*-NN algorithms, and found that in most cases, *K*-NN outperforms NGE. Then some improved versions of NGE, called NONGE, BNGE and OBNGE, were developed, where NONGE disallows overlapping rectangles while retaining nested rectangles and the same search procedure is uniformly superior to NGE, while OBNGE is a batch algorithm that incorporates an improved search algorithm and disallows nested rectangles (but still permits overlapping rectangles) and is only superior to NGE in one domain and worse in two; BNGE is a batch version of NONGE that is very efficient and requires no user tuning of parameters. They also pointed out that further research is needed to develop an NGE-like algorithm that can be robust in situations where axis-parallel hyperrectangles are inappropriate. Intuitively, the concept of neighborhood should be such that the neighbors are as close to a sample as possible but also, the neighbors should lie as homogeneously around that sample as possible. Sanchez, Pla, and Ferri (1997) showed that the geometrical placement can become much more important than the actual distances to appropriately characterize a sample by its neighborhood. As the nearest neighborhood takes into account the first property only, the nearest neighbors may not be placed symmetrically around the sample if the neighborhood in the training set is not spatially homogeneous. In fact, it has been shown that the use of local distance measures can significantly improve the behavior of the classifier in the case of a finite sample size. They proposed to make use of some alternative neighborhood definitions, obtaining the surrounding neighborhood (SN) samples, the neighbors of a sample will be considered not only in terms of proximity but also in terms of their spatial distribution with respect to that sample. More recently, Wang (2006) showed a nonparametric technique for pattern recognition, named neighborhood counting (NC), where he used neighborhoods of data points measure the similarity between two data points. Considering all neighborhoods that cover both data points, he proposed using the number of such neighborhoods as a generic measure of similarity. However, most of the work is focused on 2-norm neighborhood, few researches compare the influence bringing by different norms, such as 1-norm and infinite-norm. What is more, there is no uniform framework to understand, analyze and compare these algorithms.

In fact, neighborhoods and neighborhood relations are a class of important concepts in topology. Lin (1988, 1997) pointed out that neighborhood spaces are more general topological spaces than equivalence spaces and introduced neighborhood relation into rough set methodology, which has shown to be a powerful tool to attribute reduction, feature selection, rule extraction and reasoning with uncertainty (Hu, Yu, & Xie, 2006; Hu, Yu, Xie, & Liu, 2006; Jensen & Shen, 2004; Swiniarski & Skowron, 2003). Yao (1998) and Wu and Zhang (2002) discussed the properties of neighborhood approximation spaces. However, few applications of the model were reported these years. In this paper, we will review the basic concepts on neighborhood and neighborhood rough sets and show some properties of the model. And then we will use the model to build a uniform theoretic framework for neighborhood-based classifiers. This framework integrates feature selection with classifier construction, and classifies a test sample in the selected subspaces based on the majority class in the neighborhood of the test sample. The proposed technique combines the advantages of feature subset selection and neighborhood-based classification. It is conceptually simple and is straightforward to implement. Some experimental analysis is conducted on UCI data sets. Three kinds of norms, 1-norm, 2-norm and infinite-norm, are tried. The results show that the proposed classification systems outperform the popular CART Learning algorithm and *K*-NN classifier, and a little weaker than SVM for the three norms.

The remainder of the paper is organized as follows. The basic concepts on neighborhood rough set models are shown in Section 2. The neighborhood classifier algorithm is introduced in Section 3. Section 4 presents the experimental analysis. Then the conclusion is given in Section 5.

2. Neighborhood-based rough set model

Formally, the structural data for classification learning can be written as a tuple $IS = \langle U, A, V, f\rangle$, where U is the nonempty set of samples $\{x_1, x_2, \ldots, x_n\}$, called a universe or sample space, A is the nonempty set of variables (also called as features, inputs, attributes) $\{a_1, a_2, \ldots, a_m\}$ to characterize the samples, V_a is the value domain of attribute a; and f is an information function, f: $U \times A \to V$. More specially, $\langle U, A, V, f\rangle$ is also called a decision table if $A = C \cup D$, where C is the set of condition attributes, D is the output, also called decision.

Definition 1. Given arbitrary $x_i \in U$ and $B \subseteq C$, the neighborhood $\delta_B(x_i)$ of x_i in the subspace B is defined as

$$\delta_B(x_i) = \{x_j | x_j \in U, \Delta_B(x_i, x_j) \leqslant \delta\},$$

where Δ is a metric function. $\forall x_1, x_2, x_3 \in U$, it satisfies

(1) $\Delta(x_1, x_2) \geqslant 0$;
(2) $\Delta(x_1, x_2) = 0$, if and only if $x_1 = x_2$;
(3) $\Delta(x_1, x_2) = \Delta(x_2, x_1)$;
(4) $\Delta(x_1, x_3) \leqslant \Delta(x_1, x_2) + \Delta(x_2, x_3)$.

There are three metric functions that are widely used. Consider that x_1 and x_2 are two objects in N-dimensional space $A = \{a_1, a_2, \ldots, a_N\}$, $f(x, a_i)$ denotes the value of sample x in the ith dimension a_i, then a general metric, named Minkowsky distance, is defined as

$$\Delta_P(x_1, x_2) = \left(\sum_{i=1}^{N} |f(x_1, a_i) - f(x_2, a_i)|^P \right)^{1/P}$$

where (1) it is called Manhattan distance Δ_1 if $P = 1$; (2) Euclidean distance Δ_2, if $P = 2$; (3) Chebychev distance if $P = \infty$. The infinite-norm based distance also can be written as

$$\Delta_\infty(x_1, x_2) = \max_{i=1}^{N} (|f(x_1, a_i) - f(x_2, a_i)|)$$

The above metrics equivalently deal with the N attributes. However, the features have different influences on the classification in some cases, they should be distinctively processed. More generally, the weighted distance functions can be defined as

$$\Delta_P(x_1, x_2) = \left(\sum_{i=1}^{N} w_i |f(x_1, a_i) - f(x_2, a_i)|^P \right)^{1/P}$$

where $0 \leqslant w_i \leqslant 1$. A detailed survey on distance function can be seen in Wilson and Martinez (1997).

$\delta_B(x_i)$ is the information granule centered with sample x_i. The size of the neighborhood depends on the threshold δ. The greater δ is, the more samples will fall into the neighborhood, and the shape of the neighborhoods depends on the norm used. In 2-dimension real space, neighborhoods of x_0 in terms of the above three metrics and weighted metrics are shown as Fig. 1. 1-norm based neighborhood is a rhombus region around the center sample x_0; 2-norm based neighborhood is a ball region; while infinite-norm based neighborhood is rectangle or square.

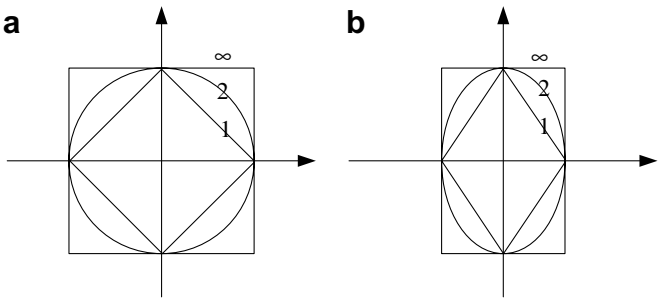

Fig. 1. Neighborhoods of x_0 in terms of three metrics and weighted metrics: (a) three metrics; (b) three weighted metrics.

Given a metric space $\langle U, \Delta \rangle$, the family of neighborhood granules $\{\delta(x_i) | x_i \in U\}$ forms an elemental granule system, which covers the universe, rather than partitioning it. We have

(1) $\forall x \in U$: $\delta(x) \neq \varnothing$;
(2) $\cup_{x \in U} \delta(x) = U$.

A neighborhood relation N over the universe can be written as a relation matrix $M(\mathrm{N}) = (r_{ij})_{n \times n}$ where

$$r_{ij} = \begin{cases} 1, & \Delta(x_i, x_j) \leqslant \delta \\ 0, & \text{otherwise} \end{cases}.$$

It is easy to show that N satisfies the following properties:

(1) reflexivity: $r_{ii} = 1$;
(2) symmetry: $r_{ij} = r_{ji}$.

Obviously, neighborhood relations are one class of similarity relations, which satisfy reflexivity and symmetry. Neighborhood relations draw the objects together for similarity or indistinguishability in terms of distances.

Note 1. $\delta(x)$ is an equivalent class and N is an equivalence relation if $\delta = 0$, this case is applicable to discrete data.

Note 2. δ can take a uniform value for all of the objects or distinct values for different objects.

Note 3. With the same threshold δ, the sizes of neighborhoods with different norms are different, and we have $\delta_1(x) \subseteq \delta_2(x) \subseteq \delta_\infty(x)$. It is easy to find with Fig. 1.

Definition 2. Giving a set of samples U, N is a neighborhood relation on U, $\{\delta(x_i) | x_i \in U\}$ is the family of neighborhood granules. Then we call $\langle U, \mathrm{N} \rangle$ a neighborhood approximation space.

Definition 3. Given $\langle U, \mathrm{N} \rangle$, for arbitrary $X \subseteq U$, two subsets of objects, called lower and upper approximations of X in terms of relation N, are defined as

$$\underline{\mathrm{N}} X = \{x_i | \delta(x_i) \subseteq X, x_i \in U\},$$
$$\overline{\mathrm{N}} X = \{x_i | \delta(x_i) \cap X \neq \emptyset, x_i \in U\}.$$

The boundary region of X in the approximation space is formulated as

$$BNX = \overline{\mathrm{N}} X - \underline{\mathrm{N}} X$$

The size of the boundary region reflects the degree of roughness of the set X in the approximation space. Assuming that X is the sample subset with a decision label, usually we hope that the boundary region of the decision is as little as possible for decreasing uncertainty in decision. The sizes

of the boundary regions depend on X, attributes B to describe U, and the threshold δ.

Theorem 1. *Given $\langle U, N\rangle$ and two nonnegative δ_1 and δ_2, if $\delta_1 \leqslant \delta_2$, we have*

(1) $\forall x_i \in U : N_1 \subseteq N_2,\ \delta_1(x_i) \subseteq \delta_2(x_i)$;
(2) $\forall X \subseteq U : \underline{N_1}X \subseteq \underline{N_2}X; \overline{N_2}X \supseteq \overline{N_1}X$,

where N_1 and N_2 are the neighborhood relations induced with δ_1 and δ_2, respectively.

Proof. $\delta_1 \leqslant \delta_2$, we have $\delta_1(x_i) \subseteq \delta_2(x_i)$. Assuming $\delta_1(x_i) \subseteq X$, we have $\delta_2(x_i) \subseteq X$. Therefore, we must have $x_i \in \underline{\mathrm{N}_2}X$ if $x_i \in \underline{\mathrm{N}_1}X$. However, x_i is not sure in $\underline{\mathrm{N}_1}X$ if we have $x_i \in \underline{\mathrm{N}_2}X$. Hence $\underline{\mathrm{N}_1}X \subseteq \underline{\mathrm{N}_2}X$. Similarly, we can get $\overline{\mathrm{N}_2}X \supseteq \overline{\mathrm{N}_1}X$. □

An information system is called a neighborhood system if the attributes generate neighborhood relation over the universe, denoted by $NIS = \langle U, A, V, f\rangle$, where A is the real-valued attribute set, f is an information function, f: $U \times A \to R$. More specially, a neighborhood information system is also called a neighborhood decision system if there are two kinds of attributes in the system: condition and decision. And then it is denoted as $NDT = \langle U, C \cup D, V, f\rangle$.

Definition 4. Given a neighborhood decision table $NDT = \langle U, C \cup D, V, f\rangle$, $X_1, X_2, \ldots, X_N$ are the object subsets with decisions 1 to N, $\delta_B(x_i)$ is the neighborhood information granules including x_i and generated by attributes $B \subseteq C$, Then the lower and upper approximations of the decision D with respect to attributes B are defined as

$$\underline{\mathrm{N}_B}D = \cup_{i=1}^{N} \underline{\mathrm{N}_B}X_i,$$
$$\overline{\mathrm{N}_B}D = \cup_{i=1}^{N} \overline{\mathrm{N}_B}X_i,$$

where

$$\underline{\mathrm{N}_B}X = \{x_i | \delta_B(x_i) \subseteq X, x_i \in U\},$$
$$\overline{\mathrm{N}_B}X = \{x_i | \delta_B(x_i) \cap X \neq \emptyset, x_i \in U\}.$$

The decision boundary region of D with respect to attributes B is defined as

$$BN(D) = \overline{\mathrm{N}_B}D - \underline{\mathrm{N}_B}D.$$

Decision boundary is the object subset whose neighborhoods come from more than one decision class. On the other hand, the lower approximation of the decision, also called *positive region* of decision, denoted by $POS_B(D)$, is the subset of objects whose neighborhoods consistently belong to one of the decision classes.

It is easy to show $\overline{\mathrm{N}_B}D = U$, $POS_B(D) \cap BN(D) = \emptyset$, $POS_B(D) \cup BN(D) = U$. Therefore, the neighborhood model divides the samples into two groups: positive region and boundary. Positive region is the sample set which can be classified into one of the classes without uncertainty with the existing attributes, while boundary is the set of samples which cannot be determinately classified.

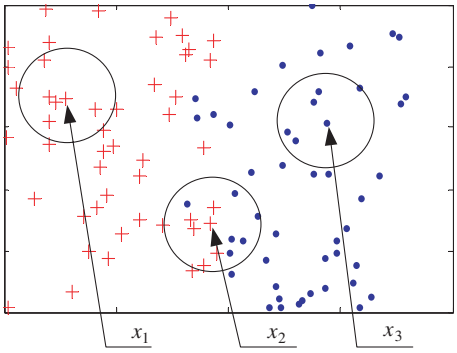

Fig. 2. An example with two classes.

Example 1. Fig. 2 shows an example of binary classification in 2-D space, where d_1 is labeled with "plus" and d_2 is labeled with "point". Consider samples x_1, x_2, and x_3, we assign circle neighborhoods to these samples. We can find $\delta(x_1) \subseteq d_1$ and $\delta(x_3) \subseteq d_2$, while $\delta(x_2) \cap d_1 \neq \emptyset$, $\delta(x_2) \cap d_2 \neq \emptyset$. According to the above definitions: $x_1 \in \underline{\mathrm{N}}d_1$, $x_3 \in \underline{\mathrm{N}}d_2$ and $x_2 \in BN(D)$.

The samples in different feature subspaces will have different boundary regions. The size of the boundary region reflects the discriminability of the classification problem in the corresponding subspaces. It also reflects the recognition power or characterizing power of the condition attributes. The greater the boundary region is, the weaker the characterizing power of the condition attributes will be. It can be formulated as follows.

Definition 5. The dependency degree of D to B is defined as the ratio of consistent objects:

$$\gamma_B(D) = \frac{|POS_B(D)|}{|U|}.$$

Where $\gamma_B(D)$ reflects the ability of B to approximate D. Obviously, $0 \leqslant \gamma_B(D) \leqslant 1$. We say that D completely depends on B if $\gamma_B(D) = 1$, denoted by $B \Rightarrow D$; otherwise we say that $D\gamma$ – *depends* on B, denoted by $B \Rightarrow_{\gamma} D$.

Theorem 2. *$\langle U, C \cup D, V, f\rangle$ is a neighborhood decision system; $B_1, B_2 \subseteq C$, $B_1 \subseteq B_2$, then we have*

(1) $\mathrm{N}_{B_1} \supseteq \mathrm{N}_{B_2}$;
(2) $\forall X \subseteq U$, $\underline{\mathrm{N}_{B_1}}X \subseteq \underline{\mathrm{N}_{B_2}}X$, $\overline{\mathrm{N}_{B_1}}X \supseteq \overline{\mathrm{N}_{B_2}}X$;
(3) $POS_{B_1}(D) \leqslant POS_{B_2}(D)$, $\gamma_{B_1}(D) \leqslant \gamma_{B_2}(D)$.

Proof. $\forall x \in U$, we have $\delta_{B_1}(x) \supseteq \delta_{B_2}(x)$ if $B_1 \subseteq B_2$. Assume that $\delta_{B_1}(x) \subseteq \underline{\mathrm{N}_{B_1}}X$, where X is one of the decision classes, then we have $\delta_{B_2}(x) \subseteq \underline{\mathrm{N}_{B_2}}X$. At the same time, there may be x_i, $\delta_{B_1}(x_i) \not\subset \underline{\mathrm{N}_{B_1}}X$ and $\delta_{B_2}(x_i) \subseteq \underline{\mathrm{N}_{B_2}}X$. Therefore, $POS_{B_1}(D) \subseteq POS_{B_2}(D)$. Accordingly, we have $\gamma_{B_1}(D) \leqslant \gamma_{B_2}(D)$. □

Theorem 2 shows that dependence monotonously increases with attributes, which means that adding a new attribute in the attribute subset at least does not decrease the dependence. This property is very important for constructing feature selection algorithms. Generally speaking, we hope to find a minimal feature subset which has the same characterizing power as the whole samples.

Definition 6. Given a neighborhood decision table $NDT = \langle U, C \cup D, V, f \rangle$, $B \subseteq C$, we say attribute subset B is a relative reduct if

(1) $\gamma_B(D) = \gamma_C(D)$;
(2) $\forall a \in B$, $\gamma_B(D) > \gamma_{B-a}(D)$.

The first condition guarantees that $POS_B(D) = POS_C(D)$. The second condition shows that there is no superfluous attribute in the reduct. Therefore, a reduct is the minimal subset of attributes which has the same approximating power as the whole attribute set. This definition presents a feasible direct to find optimal feature subsets.

3. Classification learning algorithm

Usually we hope to recognize pattern in a relatively lower dimensional space to avoid curse-of dimension, reduce cost in measuring and processing information and enhance the interpretability of learned models. However, as the development of information techniques, more and more samples and features are acquired and stored. Classification algorithms will be confused with a lot of features. Therefore, feature subset selection is implicitly or explicitly conducted for some learning systems (Muni & Pal, 2006; Neumann, Schnorr, & Steidl, 2005; Quinlan, 1993). There are two steps in constructing a neighborhood classifier. First we search an optimal feature subspace, which has a similar discriminating power as the original data, but the number of features is greatly reduced. Then we associate a neighborhood with each test sample in the selected subspace and assign the class with majority samples in the neighborhood to the test.

3.1. Feature selection based on neighborhood model

The motivation of rough set based feature selection is to select a minimal attribute subset, which has the same characterizing power as the whole attribute set, and without any redundant attribute. In other words, the dependency of the selected attributes is the same as that of the original attributes. And the dependency will decrease if any selected attribute is deleted. There are two key problems in constructing a feature selection algorithm. One is how to evaluate the selected features; the other is how to search for a good feature subset. We will discuss them in the following. Here dependence function can be introduced to evaluate the goodness of selected features.

Definition 7. Given a decision system $\langle U, C, D \rangle$, $B \subseteq C$, $a \in B$, we define the significance of an attribute as

$$SIG(a, B, D) = \gamma_B(D) - \gamma_{B-a}(D).$$

The attribute's significance is the function of three variables: a, B and D. An attribute may be of great significance in B_1 but of little significance in B_2. What is more, the attribute's significance will be different for each decision if there are multiple decision attributes in a decision table.

Definition 8. We say attribute a is *superfluous* in B with respect to D if $SIG(a, B, D) = 0$, otherwise a is indispensable. We say B is dependent if $\forall a \in B$, otherwise a is indispensable.

From another standpoint, we can also define the significance of an attribute as follows.

Definition 9. Given a decision system $\langle U, C, D \rangle$, $B \subseteq C$, $a \notin B$, the significance of an attribute is

$$SIG(a, B, D) = \gamma_{B \cup a}(D) - \gamma_B(D).$$

It is a combinational optimization problem to find all of the reducts. There are $2^{|C|}$ combinations of attribute subsets. It is not practical to search all of the reducts in $2^{|C|}$ combinations. Fortunately, in practice, we usually just require one of the reducts to train a classifier, and we do not much care whether the reduct is the minimal one. Then a tradeoff solution can be constructed, such as greedy forward search algorithm.

Algorithm 1. [Forward Attribute reduction based on neighborhood model (FARNeM)]

Input: $\langle U, C, d \rangle$ and δ // δ is the threshold to control the size of the neighborhood
Specify the norm to be used
Output: reduct *red*
Step 1: $\varnothing \to red$; // *red* is the pool to contain the selected attributes
Step 2: For each $a_i \in C - red$
compute $SIG(a_i, B, D) = \gamma_{red \cup a_i}(D) - \gamma_{red}(D)$, //
Here we define $\gamma_{\varnothing}(D) = 0$
end
Step 3: Select the attribute a_k which satisfies $SIG(a_k, B, D) = \max_i(SIG(a_i, red, B))$
Step 4: if $Sig(a_k, B, D) > 0$,
$red \cup a_k \to red$
go to step 2
else
return *red*
Step 5: end

Here the FARNeM algorithm adds an attribute with the great increment of dependence into the reduct in each circle until the dependence does not increase, namely, adding any

new attribute will not increase the dependence in this case. The time complexity of the algorithm is $O(N \times N)$, where N is the number of candidate attributes.

3.2. Classification

Both K-NN classifiers and neighborhood classifiers (NEC) are based on the general idea of estimating the class of a sample from its neighbors, but the NEC considers a kind of neighborhood which allows one to inspect a sufficiently small and near area around the sample, in such a way that all training samples surrounding the test take part in the classification process.

Here, we first find the training samples in the neighborhood of the test, and then assign the majority class of the neighborhood to the test.

Algorithm 2. [Neighborhood classifiers (NEC)]

Input: Training set: $\langle U, C, D\rangle$, Test sample: s; Threshold δ, Specify the norm used
Output: class of s

1. compute the distance between s and $x_i \in U$ with the used norm.
2. find the samples in the neighborhood $\delta(s)$ of s.
3. find the class d_j with the majority training samples in $\delta(s)$.
4. assign d_j to the test s.

The most important problem in neighborhood-based classification is the threshold δ, which determines the size of the neighborhood. No sample will be included in the neighborhood if δ is too small; on the other hand, the neighborhood cannot reflect the local information of the test if a too great neighborhood is taken into consideration. Here we compute δ as follows:

$$\delta = \min(\Delta(x_i, s)) + w \cdot \mathrm{range}(\Delta(x_i, s)), \quad w \leqslant 1,$$

where $x_i | i = 1, \ldots, n$ is the set of training samples, $\min(\Delta(x_i, s))$ means the minimal value of distance between x_i and the test sample s; $\mathrm{range}(\Delta(x_i, s))$ is the value range of $\Delta(x_i, s)$. In this case, the threshold δ is dynamically assigned based on the local and global information around s. We will recommend a value range of w based on experimental analysis in Section 4.

It is notable that neighborhood-based classification is independent of neighborhood-based feature selection. Therefore, the output of neighborhood-based feature selection is also applicable to other classification learning algorithms, such as CART and SVM.

4. Experimental analysis

In order to test the proposed classification model, some data sets are downloaded from the machine learning data repository, University of California at Irvine. The data sets are outlined in Table 1.

There are two objectives to conduct the experiments. The first one is to compare classification performances of K-NN, neighborhood classifier (NEC), CART and SVM in original feature spaces and reduced feature spaces. The second one is to get experiential rule to specify the parameter w used in NEC.

Table 2 shows the comparison of classification performances based on 10-NN and neighborhood classifier, where $\delta = 0.6$ to 0.8. From the average accuracies, we

Table 1
Data description

	Data set	Abbreviation	Samples	Features	Classes
1	Ionosphere	Iono	351	34	2
2	Sonar, mines vs. rocks	Sonar	208	60	2
3	Wisconsin diagnostic breast cancer	WDBC	569	31	2
4	Wisconsin prognostic breast cancer	WPBC	198	33	2
5	Wine recognition	Wine	178	13	3

Table 2
Comparison of classification performances based on K-NN and neighborhood classifier

Data	1-Norm		2-Norm		Infinite-norm	
	10-NN	Neighborhood	10-NN	Neighborhood	10-NN	Neighborhood
Iono	0.8525 ± 0.0471	0.8926 ± 0.0496	0.8240 ± 0.0502	0.8581 ± 0.0592	0.8577 ± 0.0748	0.8673 ± 0.0731
Sonar	0.7502 ± 0.0539	0.8657 ± 0.0533	0.7262 ± 0.0705	0.8367 ± 0.0559	0.7071 ± 0.0789	0.8179 ± 0.0912
WDBC	0.9632 ± 0.0324	0.9685 ± 0.0230	0.9667 ± 0.0209	0.9685 ± 0.0214	0.9475 ± 0.0259	0.9492 ± 0.0240
WPBC	0.7776 ± 0.0706	0.7882 ± 0.0700	0.7626 ± 0.0589	0.7882 ± 0.0700	0.7208 ± 0.0866	0.7113 ± 0.0929
Wine	0.9778 ± 0.0287	0.9660 ± 0.0393	0.9549 ± 0.0354	0.9722 ± 0.0393	0.9319 ± 0.0456	0.9493 ± 0.0412
Average	0.8643	0.8962	0.8469	0.8847	0.8330	0.8590

can find that neighborhood classifier outperforms K-NN for all of the three norms. And as to data Sonar, NEC is greatly superior to 10-NN.

We conduct FARNeM attribute reduction algorithm on these data sets. We try $\delta = 0.125$ and $\delta = 0.25$ used in FARNeM. What is more, all of the three norms are tried. The numbers of selected features based on different thresholds and norms are shown in Table 3. We can see that most of the candidate attributes are deleted. And with the same threshold, 1-norm based attribute reduction gets the least features, while infinite-norm based algorithms get the most ones. However, if we change the threshold from 0.125 to 0.25, the numbers of selected features based on 1-norm algorithm with threshold $\delta = 0.25$ are comparable with or more than that based on 2-norm algorithm with $\delta = 0.125$. The similar case occurs as to 2-norm and infinite-norm based attribute reduction. As we have pointed in Section 2, with the same threshold δ, the sizes of neighborhoods with different norms are different, and we have $\delta_1(x) \subseteq \delta_2(x) \subseteq \delta_\infty(x)$. Therefore, with the same threshold, $\{\delta_\infty(x_i)|i = 1,\ldots,n\}$ is coarser than $\{\delta_2(x_i)|i = 1,\ldots,n\}$ and $\{\delta_1(x_i)|i = 1,\ldots,n\}$. By and large, more features will be required if the samples are partitioned into the same granularity with infinite-norm or 2-norm than 1-norm. How this difference can be avoided is by specifying different thresholds, namely, by adjusting the threshold, we can obtain similar results with different norm based attribute reduction.

Tables 4–6 show the classification accuracies of K-NN and NEC with different norms in 1-norm, 2-norm and infinite-norm based feature subspaces, respectively. Comparing these accuracies with those in Table 2, we can find that the classification performances are kept or improved in the reduced subspaces although most of the features are deleted. It shows that the neighborhood based feature selection is able to find some good features for classification and efficiently delete the redundant and irrelevant features from the original data.

Tables 7 and 8 show the classification accuracies with CART and RBF-SVM learning algorithms in the three

Table 3
Numbers of selected features

Raw data		1-Norm		2-Norm		Infinite-norm	
		$\delta = 0.125$	$\delta = 0.25$	$\delta = 0.125$	$\delta = 0.25$	$\delta = 0.125$	$\delta = 0.25$
Iono	34	6	9	9	16	12	25
Sonar	60	5	6	6	10	7	20
WDBC	31	6	8	8	23	21	23
WPBC	33	5	7	6	12	10	27
Wine	13	4	5	5	7	6	13
Average	34.2	5.2	7	6.8	13.6	11.2	21.6

Table 4
Comparing accuracies in 1-norm feature subspace

Data	1-Norm		2-Norm		Infinite-norm	
	10-NN	Neighborhood	10-NN	Neighborhood	10-NN	Neighborhood
Iono	0.8955 ± 0.0602	0.9376 ± 0.0396	0.8808 ± 0.0706	0.9119 ± 0.0410	0.8385 ± 0.0728	0.9067 ± 0.0470
Sonar	0.7833 ± 0.1064	0.7638 ± 0.0857	0.7450 ± 0.0719	0.7781 ± 0.0762	0.7738 ± 0.0767	0.7788 ± 0.0569
WDBC	0.9615 ± 0.0283	0.9614 ± 0.0199	0.9649 ± 0.0234	0.9632 ± 0.0193	0.9561 ± 0.0237	0.9526 ± 0.0275
WPBC	0.7292 ± 0.1816	0.7350 ± 0.1324	0.7350 ± 0.1471	0.7300 ± 0.1474	0.7463 ± 0.0806	0.7458 ± 0.1057
Wine	0.9722 ± 0.0393	0.9722 ± 0.0293	0.9833 ± 0.0268	0.9660 ± 0.0294	0.9604 ± 0.0379	0.9722 ± 0.0393
Average	0.8683	0.8740	0.8618	0.8698	0.8550	0.8712

$\delta = 0.125$, $w = 0.06$ to 0.08.

Table 5
Comparing accuracies in 2-norm feature subspace

Data	1-Norm		2-Norm		Infinite-norm	
	10-NN	Neighborhood	10-NN	Neighborhood	10-NN	Neighborhood
Iono	0.8902 ± 0.0742	0.9183 ± 0.0649	0.8729 ± 0.0705	0.9155 ± 0.0656	0.8118 ± 0.0738	0.8751 ± 0.0581
Sonar	0.7448 ± 0.0808	0.8179 ± 0.0762	0.7593 ± 0.0657	0.8079 ± 0.0589	0.7736 ± 0.0702	0.7743 ± 0.0670
WDBC	0.9562 ± 0.0219	0.9579 ± 0.0274	0.9562 ± 0.9562	0.9544 ± 0.0308	0.9544 ± 0.0348	0.9474 ± 0.0347
WPBC	0.7053 ± 0.1354	0.7308 ± 0.0985	0.7313 ± 0.0867	0.7316 ± 0.0707	0.7563 ± 0.0881	0.7668 ± 0.0718
Wine	0.9271 ± 0.0372	0.9271 ± 0.0587	0.9382 ± 0.0486	0.9437 ± 0.0524	0.9271 ± 0.0829	0.9493 ± 0.0488
Average	0.8447	0.8704	0.8516	0.8706	0.8446	0.8626

$\delta = 0.125$, $w = 0.06$ to 0.08.

Table 6
Comparing accuracies in infinite-norm feature subspace

Data	1-Norm		2-Norm		Infinite-norm	
	10-NN	Neighborhood	10-NN	Neighborhood	10-NN	Neighborhood
Iono	0.8785 ± 0.0568	0.9045 ± 0.0669	0.8380 ± 0.0557	0.8783 ± 0.0441	0.8151 ± 0.0778	0.8754 ± 0.0648
Sonar	0.7457 ± 0.0767	0.8031 ± 0.0654	0.7552 ± 0.0641	0.8124 ± 0.0829	0.7017 ± 0.0466	0.7976 ± 0.0877
WDBC	0.9632 ± 0.0209	0.9597 ± 0.0234	0.9563 ± 0.0319	0.9580 ± 0.0287	0.9439 ± 0.0334	0.9509 ± 0.0306
WPBC	0.7632 ± 0.0304	0.7582 ± 0.0767	0.7926 ± 0.0794	0.7579 ± 0.0848	0.7471 ± 0.0364	0.7274 ± 0.0480
Wine	0.9778 ± 0.0287	0.9778 ± 0.0388	0.9722 ± 0.0293	0.9833 ± 0.0268	0.9556 ± 0.0438	0.9722 ± 0.0293
Average	0.8657	0.8807	0.8629	0.8780	0.8327	0.8647

$\delta = 0.125$, $w = 0.06$ to 0.08.

Table 7
Comparison of classification accuracies based on CART learning algorithm

	Raw data	1-Norm	2-Norm	Infinite-norm
Iono	0.8755 ± 0.0693	0.8926 ± 0.0557	0.8952 ± 0.0582	0.9063 ± 0.0396
Sonar	0.7207 ± 0.1394	0.6812 ± 0.1196	0.6829 ± 0.0926	0.7550 ± 0.0683
WDBC	0.9050 ± 0.0455	0.9315 ± 0.0253	0.9455 ± 0.0316	0.9228 ± 0.0361
WPBC	0.6963 ± 0.0826	0.6953 ± 0.1117	0.6855 ± 0.1098	0.6453 ± 0.1292
Wine	0.8986 ± 0.0635	0.9208 ± 0.0481	0.9153 ± 0.0483	0.9208 ± 0.0481
Average	0.8192	0.8243	0.8249	0.8300

$\delta = 0.125$.

Table 8
Comparison of classification accuracies based on SVM learning algorithm $\delta = 0.125$

	Raw data	1-Norm	2-Norm	Infinite-norm
Iono	0.9379 ± 0.0507	0.9122 ± 0.0501	0.9264 ± 0.0517	0.9293 ± 0.0627
Sonar	0.8510 ± 0.0948	0.7783 ± 0.1100	0.7543 ± 0.1309	0.8364 ± 0.0837
WDBC	0.9808 ± 0.0225	0.9614 ± 0.0259	0.9667 ± 0.0207	0.9790 ± 0.0161
WPBC	0.7779 ± 0.0420	0.7632 ± 0.0304	0.7632 ± 0.0304	0.7842 ± 0.0769
Wine	0.9889 ± 0.0234	0.9660 ± 0.0294	0.9493 ± 0.0412	0.9833 ± 0.0268
Average	0.9073	0.8762	0.8720	0.9024

norms based feature subspaces, where $\delta = 0.125$, and 1-norm means the 10-fold cross validation accuracies in feature subspaces selected by 1-norm based neighborhood model, the same for 2-norm and infinite-norm. From Table 7, we can find that the neighborhood based feature selection improves recognition power for CART learning algorithm; however, the performance slightly decreases for SVM in Table 8. Especially, with regard to sonar data, the accuracy decreases from 0.8510 to 0.7783 and 0.7543. There are 60 features in the original sonar data set; however, 1-norm and 2-norm based feature selection just select 5 and 6 features. In the meantime, infinite-norm based neighborhood model selects 7 features for classification. Accordingly, the classification accuracies are better and

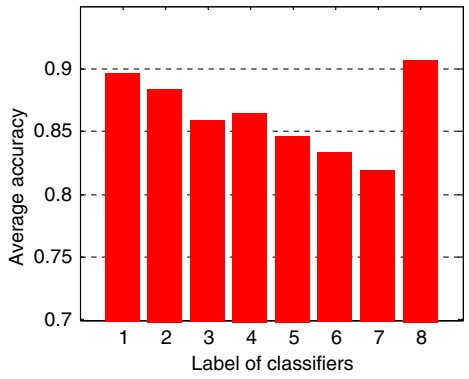

Fig. 3. Accuracies in original feature spaces.

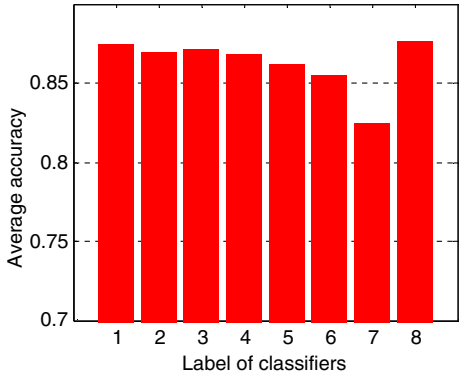

Fig. 4. Accuracies in 1-norm based feature subspaces.

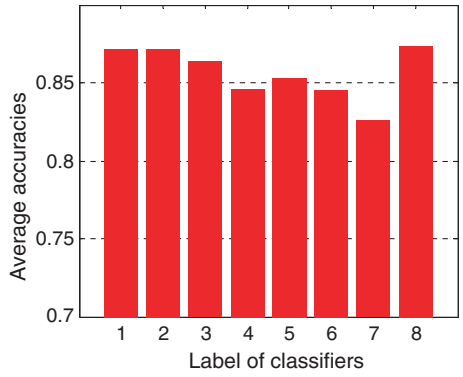

Fig. 5. Accuracies in 2-norm based feature subspaces.

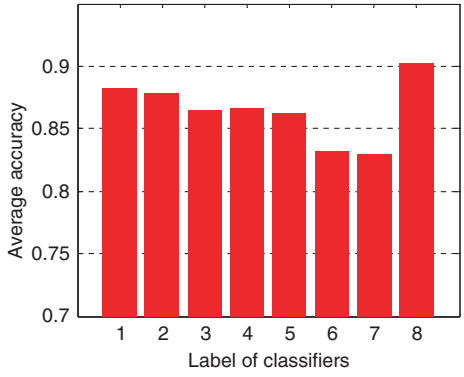

Fig. 6. Accuracies in infinite-norm based feature subspaces.

slightly weaker than the original data for CART and SVM in the infinite-norm based feature subspace. It shows that too many features are deleted in this case; we can specify a proper threshold δ to avoid this problem.

Figs. 3–6 show the comparison of average classification accuracies based on different classifiers and feature subspaces, where labels of classifiers 1, 2 and 3 mean 1-, 2- and infinite-norm neighborhood classifiers; 4, 5 and 6 denote 1-, 2- and infinite-norm based 10-NN classifiers; 7 is CART and 8 is SVM. In terms of average accuracies of the five data sets, we can conclude that NEC is superior to *K*-NN and CART, and a little weaker than SVM.

With the above experimental analysis, we can obtain the following conclusions: neighborhood classifier is a kind of simple, easy to implement, yet powerful classification system; neighborhood model based feature selection is able to find the useful features and delete redundant and irrelevant attributes.

Now we conduct a series of experiments to find the optimal parameter w used to control the size of the neighborhood. We try w from 0 to 0.6 with step 0.02, and compute classification accuracies based on the 10-fold cross validation. All of the three norms are tried. Fig. 7 presents the classification accuracy curves varying with w as for data sets: iono, sonar, WDBC and wine. Here we can find that there are similar trends in these curves. Accuracies increase at first, and then decrease after a threshold. The

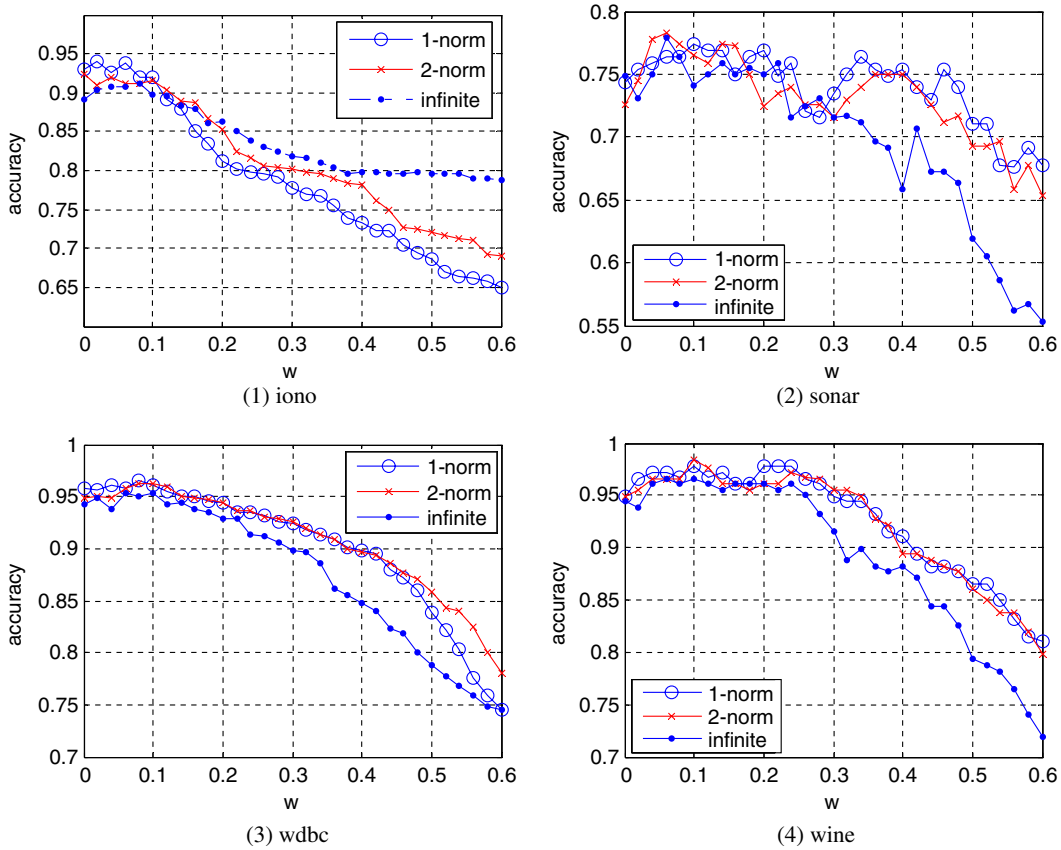

Fig. 7. Classification accuracy curves varying with w.

accuracies near the point $w = 0.1$ are optimal or near optimal. Here we recommend that w should take values in the range $[0, 0.1]$. Neighborhood classifier is equivalent to 1-NN if $w = 0$ because only the sample with minimal distance is included in the neighborhood in this case.

5. Conclusion and future work

K-NN classifiers are widely discussed and applied; however, as another classification technique based on local information, neighborhood classifiers have not been carefully studied. In this paper, we introduce neighborhood rough set model as a basic theoretic framework, which presents a conceptually simple and easy to implement method to understand and construct neighborhood-based attribute reduction technique and classifiers.

Experiments with UCI data sets show both in the original feature spaces and in neighborhood-based feature subspaces, Neighborhood classifiers outperform *K*-NN and CART algorithm, and are a little weaker than SVM. However, considering the simpleness and interpretability, neighborhood classifiers will get their identity in some applications. What is more, we also find that the neighborhood model has great power in attribute reduction. The classification accuracies are kept or improved although most of the features are deleted from the original data with neighborhood rough set based attribute reduction, which shows that neighborhood-based attribute reduction algorithm can select the useful features and eliminate the redundant and irrelevant information.

Neighborhood classifier can be understood as a classification system which uses the samples in the neighborhood to estimate the local class probability density of the test samples. In fact, Parzen window based probability density estimation has been widely analyzed and applied (Duda & Hart, 1973; Girolami & He, 2003). In this paper, the samples in the neighborhood have the same influence on the estimated probability density; however, if we consider the neighborhood as a window function, we can use other window functions to predict the class probability, where different weights will be assigned to the samples in the neighborhood. On the other hand, we also can use the concept of fuzzy neighborhood to generalize the proposed algorithm. Similar to *K*-NN, neighborhood classifier is a lazy learning algorithm, the test process is time-consuming. Therefore, some techniques used to speed up and improve *K*-NN (Hart, 1968; Tan, 2005; Zhang & Srihari, 2004) also can be introduced.

References

Anil, K. G. (2006). On optimum choice of k in nearest neighbor classification. *Computational Statistics and Data Analysis, 50*, 3113–3123.

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

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Fix, E., & Hodges, J. (1951). Discriminatory analysis. *Nonparametric discrimination: Consistency properties*. Tech. Report 4, USAF School of Aviation Medicine, Randolph Field, Texas.

Fu, A. W., Chan, P. M., Cheung, Y. L., et al. (2000). Dynamic vp-tree indexing for n-nearest neighbor search given pair-wise distances. *VLDB Journal, 9*, 154–173.

Fukunaga, K., & Narendra, M. (1975). A branch and bound algorithm for computing k-nearest neighbors. *IEEE Transactions on Computers, 24*, 750–753.

Girolami, M., & He, C. (2003). Probability density estimation from optimally condensed data samples. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*, 1253–1264.

Hart, P. E. (1968). The condensed nearest neighbor. *IEEE Transactions on Information Theory, 14*, 515–516.

Hu, Q. H., Yu, D. R., & Xie, Z. X. (2006). Information-preserving hybrid data reduction based on fuzzy-rough techniques. *Pattern Recognition Letters, 27*, 414–423.

Hu, Q. H., Yu, D. R., Xie, Z. X., & Liu, J. F. (2006). Fuzzy probabilistic approximation spaces and their information measures. *IEEE Transactions on Fuzzy Systems, 14*, 191–201.

Jensen, R., & Shen, Q. (2004). Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. *IEEE Transactions of Knowledge and Data Engineering, 16*, 1457–1471.

Kuncheva, L. I., & Lakhmi, C. J. (1999). Nearest neighbor classifier: simultaneous editing and feature selection. *Pattern Recognition Letters, 20*, 1149–1156.

Kushilevitz, E., Ostrovsky, R., & Rabani, Y. (2000). Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM Journal on Computing, 30*, 457–474.

Lin, T. Y. (1988). Neighborhood systems and relational database. In *Proceedings of 1988 ACM sixteenth annual computer science conference, February 23–25*.

Lin, T. Y. (1997). Neighborhood systems – application to qualitative fuzzy and rough sets. In P. P. Wang, *Advances in machine intelligence and soft-computing*, Department of Electrical Engineering, Duke University Durham, North Carolina, USA (pp. 132–155).

Lindenbaum, M., Markovitch, S., & Rusakov, D. (2004). Selective sampling for nearest neighbor classifiers. *Machine Learning, 54*, 125–152.

Muni, D. P., & Pal, N. R. D. (2006). Genetic programming for simultaneous feature selection and classifier design. *IEEE Transactions on Systems Man and Cybernetics Part B – Cybernetics, 36*, 106–117.

Neumann, J., Schnorr, C., & Steidl, G. (2005). Combined SVM-based feature selection and classification. *Machine Learning, 61*, 129–150.

Owen, A. (1984). A neighbourhood-based classifier for LANDSAT data. *The Canadian Journal of Statistics, 12*, 191–200.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo: Morgan Kaufman.

Salzberg, S. (1991). A nearest hyperrectangle learning method. *Machine Learning, 6*, 277–309.

Sanchez, J. S., Pla, F., & Ferri, F. J. (1997). On the use of neighbourhood-based non-parametric classifiers. *Pattern Recognition Letters, 18*, 1179–1186.

Short, R. D., & Fukunaga, K. (1981). Optimal distance measure for nearest neighbor classification. *IEEE Transactions on Information Theory, 27*, 622–627.

Swiniarski, R. W., & Skowron, A. (2003). Rough set methods in feature selection and recognition. *Pattern Recognition Letters, 24*, 833–849.

Tan, S. B. (2005). Neighbor-weighted K-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications, 28*, 667–671.

Vidal, E. (1986). An algorithm for finding nearest neighbours in (approximately) constant average time complexity. *Pattern Recognition Letters, 4*, 145–157.

Wang, H. (2006). Nearest neighbors by neighborhood counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*, 942–953.

Wettschereck, D., & Dieterich, T. G. (1995). An experimental comparison of the nearest neighbor and nearest-hyperrectangle algorithms. *Machine Learning, 19*, 5–27.

Wilson, D. R., & Martinez, T. R. (1997). Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research, 6*, 1–34.

Wilson, D. R., & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning, 38*, 257–286.

Wu, W. Z., & Zhang, W. X. (2002). Neighborhood operator systems and approximations. *Information Sciences, 144*, 201–217.

Yao, Y. Y. (1998). Relational interpretations of neighborhood operators and rough set approximation operators. *Information Sciences, 111*, 239–259.

Zhang, B., & Srihari, S. N. (2004). Fast k-nearest neighbor classification using cluster-based trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*, 525–528.

Zhou, C., Yan, Y., & Chen, Q. (2006). Improving nearest neighbor classification with cam weighted distance. *Pattern Recognition, 39*, 635–645.