Application of Proximity Graphs to Editing Nearest Neighbor Decision Rules*

Binay K. Bhattacharya

School of Computing Science
Simon Fraser University

Ronald S. Poulsen

Departments of Biomedical Engineering and Pathology
McGill University

Godfried T. Toussaint

School of Computer Science
McGill University

Abstract

Non-parametric decision rules, such as the nearest neighbor (*NN*) rule, are attractive because no a priori knowledge is required concerning the underlying distributions of the data. Two traditional criticisms directed at the *NN*-rule concern the large amounts of storage and computation involved due to the apparent necessity to store all the sample (training) data. Thus there has been considerable interest in "editing" or "thinning" the training data in an attempt to store only a fraction of it. Previous editing algorithms suffered from the drawback that they delivered edited sets that were not *decision-boundary consistent*, i.e., the decision boundary determined by the edited set differed from that specified by the entire original training data. In this paper several geometric methods based on proximity graphs are proposed for editing the training data for use in the *NN*-rule. Most notably, one of the methods yields a decision-boundary consistent edited set and therefore a decision rule that preserves all the desirable convergence properties of the *NN*-rule that is based on the original entire training data. The methods are all derived from the Voronoi diagram of the sample data and make use of subgraphs of the Delaunay triangulation. The methods are compared empirically through experiments on synthetic data as well as real world data in the automatic detection of cervical cancer. Finally, algorithms for the efficient implementation of these techniques are discussed.

1. Introduction

In computer vision and pattern recognition problems it is often required to make a decision of class membership for a given unknown object on the basis of some numerical information obtained by making measurements (observing features) on the object at hand. Let each of the objects to be classified belong to one of M classes denoted by C_i, i=1, 2,..., M. Let $P(C_i)$ denote the *a priori*

* This research was supported by grants NSERC-OGP0009293, FCAR-93ER0291 and NSERC-OGP0004516, as well as the Macdonald-Stewart Foundation in Montreal.

probability of occurrence of objects belonging to class C_i. Let $X = (x_1, x_2,..., x_d)$, $X \in \mathbf{R}^d$, denote the set of d measurements made on an object and let $p(X|C_i)$ denote the probability density function of X given that the pattern on which X was observed belongs to class C_i. Then it is well known that the decision rule that minimizes the expected probability of error (miss-classification) in making a decision on X is to choose class C_i if: $p(X|C_i)P(C_i) > p(X|C_j)P(C_j)$ for all $j \neq i$. It is also well known that the resulting Bayes (optimal) probability of error, denoted by P_e is given by the expression:

$$P_e = 1 - \int_{-\infty}^{\infty} max_i[p(X/C_i)P(C_i)]dX$$

To be able to use the above Bayes (optimal) decision rule it is required to know the *a priori* probabilities $P(C_i)$ and the class conditional probability density functions $p(X|C_i)$ for all i. When these are not known one may resort to the use of non-parametric decision rules such as the nearest neighbor decision rule (*NN*-rule).

In the non-parametric classification problem we have available a set of n feature vectors taken from a collected data set of n objects (patterns) denoted by $\{\mathbf{X},\Theta\} = \{(X_1,\theta_1), (X_2,\theta_2),..., (X_n,\theta_n)\}$, where X_i and θ_i denote, respectively, the feature vector on the ith object and the class label of the ith object. The labels θ_i are assumed to be correct and are taken from the integers $\{1,2,...,M\}$, i.e., the patterns may belong to one of M classes. One of the most attractive non-parametric decision rules is the so-called nearest-neighbor rule (*NN*-rule) [CH67], [De81]. Let X be a new object (feature vector) to be classified and let $X_k^* \in \{X_1, X_2,..., X_n\}$ be the feature vector closest to X, where closeness is measured by, say, the Euclidean distance between X and X_k^* in $\mathbf{R}^d$. The nearest neighbor decision rule classifies the unknown object X as belonging to class θ_k^*. Let $P_e^n(NN) = Pr\{\theta \neq \theta_k^*\}$ denote the resulting probability of misclassification (error), where θ is the true class of X, and let $P_e(NN)$ denote the limit of $P_e^n(NN)$ as n approaches infinity. It has been shown by Cover and Hart [CH67] that as n goes to infinity the asymptotic nearest neighbor error is bounded in terms of the Bayes error by:

$$P_e \le P_e(NN) \le P_e\left[2 - M\left(\frac{P_e}{M-1}\right)\right]$$

Therefore the asymptotic probability of error of the nearest neighbor rule is close to optimal. Furthermore, with a suitable modification of the *NN*-rule we can obtain a probability of error as close to optimal as desired. Such a modification (the *k-NN* rule) will be discussed in the conclusion.

In proving the above result Cover and Hart [CH67] had some restrictions on the underlying distributions but more recently Devroye [De81] and Stone [St77] proved the above results for all distributions. These bounds, together with the transparent simplicity of the rule, make the rule very attractive. However, the apparent necessity to store all the data $\{\mathbf{X},\Theta\}$ and the resulting excessive computational requirements, have discouraged many researchers from using the rule in practice.

In order to combat the storage problem, and resulting computation, many researchers, starting with Hart [Ha68], proposed schemes for “editing” the original data $\{\mathbf{X},\Theta\}$ (also referred to as “reducing,” “thinning,” “condensing,” “pre-processing” and “prototype selection”) so that fewer feature vectors need be stored. Denote the edited subset of $\{\mathbf{X},\Theta\}$ by $\{\mathbf{X},\Theta\}_E$. At least a dozen other examples of editing schemes have been proposed [Ri75], [To76a], [To76b], [Sw72], [GK79], [FP70], [Ul74], [Ga72], [Ch74], [FM84], [Ol79]. Most recently neural networks have been used to select the prototypes to be used in the nearest neighbor rules [Ya93], [YM91]. All these techniques have several properties in common. For one, most are *sequential* in nature and the resulting $\{\mathbf{X},\Theta\}_E$ is a function of the *order* in which $\{\mathbf{X},\Theta\}$ is processed. Secondly they all attempt to obtain an edited set that will determine only *approximately* the original decision boundary in $\mathbf{R}^d$ that is determined by $\{\mathbf{X},\Theta\}$. To this end they use heuristics which complicate the algorithms, in some cases requiring a great deal of computation if a minimal-size edited set is required, and generally result in rather involved procedures that are very difficult to analyze theoretically. Furthermore, it has been shown that obtaining minimal size edited sets with some of these editing algorithms is NP-complete [Wi92]. While some of the schemes [Ha68] result in an edited set that is *training-set consistent* (i.e., $\{\mathbf{X},\Theta\}_E$ classifies all objects in $\{\mathbf{X},\Theta\}$ correctly), none of them yield an edited set which is *decision-boundary consistent* (i.e., $\{\mathbf{X},\Theta\}_E$ defines precisely the same decision boundary in $\mathbf{R}^d$ as $\{\mathbf{X},\Theta\}$). Thus with these editing schemes we have not only the disconcerting fact that $\{\mathbf{X},\Theta\}_E$ does not implement the originally intended decision boundary, but we do not even know the relationship that exists, if any, between the resulting $\{\mathbf{X},\Theta\}_E$ and one that is decision-boundary consistent.

In this paper we propose several new methods for editing the data for the *NN*-rule and compare them theoretically and experimentally, with respect to (1) storage requirements, (2) computation time and (3) resulting probability of misclassification, to the exhaustive (full training set) rule. The proposed approaches are based on well-known graph structures that are first computed on $\{\mathbf{X},\Theta\}$. The graph structures are proximity graphs obtained from the Voronoi diagram of $\{\mathbf{X},\Theta\}$. The methods have the merit that they are *exact* and yield edited sets *independent* of the order in which the data are processed. Furthermore, one method yields edited sets which are not only both training-set and decision-boundary consistent, but also of *minimal* size when the input data $\{\mathbf{X},\Theta\}$ is given in general position. The methods are compared empirically through experiments on synthetic data as well as real world data for the problem of the automatic detection of cervical cancer. Finally algorithms are given for obtaining the edited sets efficiently.

2. The Voronoi Diagram Approach

2.1 A Geometric Look at the Nearest Neighbor Rule

We shall explain and illustrate most concepts in the plane for simplicity of notation and in the interest of clarity. However, the arguments extend to higher dimensions. Indeed, the crucial theorems will be proved in $\mathbf{R}^d$. Let $\{\mathbf{X},\Theta\}$ consist of the 20 planar points which are correctly classified as either class 1 or class 2 points (see Fig. 1). The points denoted by solid dots belong to class 1 and those denoted by empty squares belong to class 2. The nearest-neighbor decision boundary (*NN*-boundary) is defined as the boundary of a subdivision of space which separates the data points of $\{\mathbf{X},\Theta\}$ into two sets associated with each of the two classes in such a way that any point on the *NN*-boundary is nearest (and equally close) to at least two data points of $\{\mathbf{X},\Theta\}$ that belong to different classes. Fig. 1 shows the *NN*-boundary determined by the 20 data points shown. The plane is thus partitioned, for the data points shown in Fig. 1, into two disjoint regions such that all the data points in one region belong to one and the same class. Note that in general the region associ-

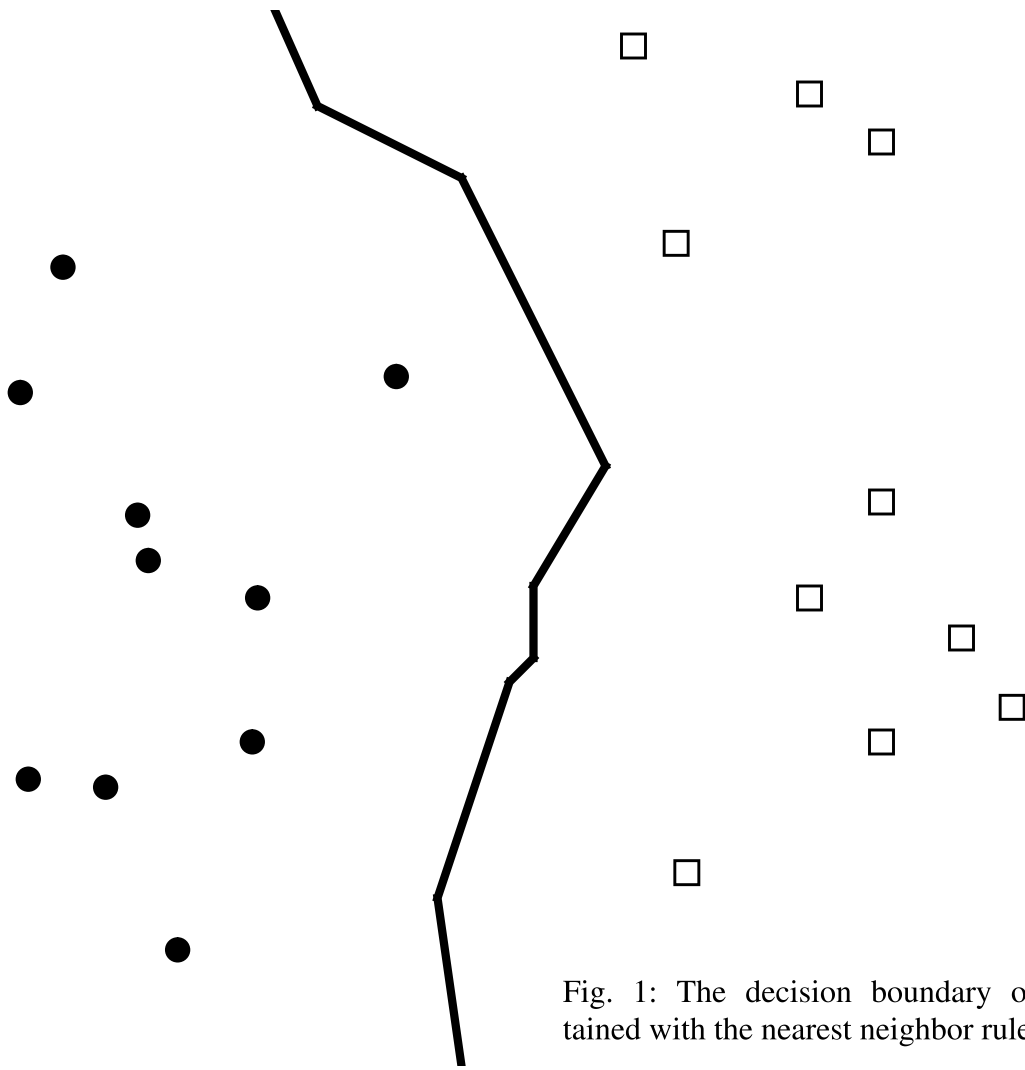

Fig. 1: The decision boundary obtained with the nearest neighbor rule.

ated with a single class may consist of several (disjoint) connected components. A new unknown point that is to be classified and is contained in the class 1 region will have a data point of $\{\mathbf{X},\Theta\}$ belonging to class 1 as its nearest neighbor. Therefore, using the *NN*-rule, the unknown point would be classified into class 1. Similarly, any unknown point, lying in the class 2 region, is classified into class 2. Thus, the problem of classifying an unknown point, using the *NN*-rule, reduces to the problem of determining the connected open region of the planar subdivision generated by the *NN*-boundary, in which the given unknown point lies [DW78].

It should be pointed out that when the *NN*-rule is used in practice the *NN*-boundary is usually not computed explicitly. Instead distances are calculated to all the data points in $\{\mathbf{X},\Theta\}$ and the nearest neighbor is found by selecting the smallest distance encountered. We mention in passing that finding the nearest neighbor of X among $\{\mathbf{X},\Theta\}$ does not require that distances must be computed from X to all elements of $\{\mathbf{X},\Theta\}$, as is often believed. Indeed there are several efficient methods for computing very few distances. These methods can be used once the data is edited with the methods proposed in this paper. For the best such methods the reader is referred to [AM93]. The algorithm of Dasarathy and White [DW78] is the only known algorithm which generates the *NN*-boundary explicitly and directly from the reference set $\{\mathbf{X},\Theta\}$ using the fact that any arbitrary point on the *NN*-boundary is nearest to at least two data points of the reference set belonging to different classes. They consider generating the *NN*-boundary as an application of a maximin optimization problem. The worst-case complexity of their algorithm, to find the *NN*-boundary determined by a reference set $\{\mathbf{X},\Theta\}$ containing n data points in d-space, is $O(dn^{d+2})$. Even for a moderate size problem the computation time is phenomenal. The authors also claim that the average complexity of their algorithm for d=3 is $O(n^{3.85})$. However this version of their algorithm does not appear to extend to higher dimensions.

The Voronoi editing algorithm described below is the only algorithm which reduces the reference set $\{\mathbf{X},\Theta\}$ in such a way that the *NN*-boundaries, defined by the *reduced* set and the reference set $\{\mathbf{X},\Theta\}$, are exactly the same, i.e., the reduced set is *decision boundary consistent* and therefore, also *reference set consistent*. Furthermore it is the minimal-size such edited set when the input data consist of points in *general position*.

2.2 The Voronoi Editing Algorithm

The Voronoi editing algorithm finds a reduced reference set by using the Voronoi diagram of the reference set $\{\mathbf{X},\Theta\}$.

Let us consider the same reference set as shown in Fig. 1. The Voronoi diagram of the reference set is shown in Fig. 2. The early work on the Voronoi diagram, its construction, and other associated properties are discussed in detail in [Bo81], [BR79], [Br79a], [Br79b], [BDF78], [GS78], [Sh78], [Kl80], and [AB83]. More recent work on Voronoi diagrams can be found in the book by Edelsbrunner [Ed87]. An entire book on the subject was written by Klein [Kl89]. An exhaustive and unified exposition of the mathematical and algorithmic properties of Voronoi diagrams was recently published by Aurenhammer [Au91].

From Fig. 2 it is noticed that the *NN*-boundary (shown in thick lines) is contained in the Voronoi diagram. This is due to the fact that the Voronoi diagram, by definition, is a partition of space into regions which are the loci of points of space closer to each data point than to any other data point and therefore it contains all the proximity information determined by a given set of data points necessary by the *NN*-rule [Sh78]. We now present the Voronoi editing algorithm.

<u>Algorithm-Voronoi Editing</u>

Begin

Step 1: Construct the Voronoi diagram of $\{\mathbf{X},\Theta\}$.

Step 2: Visit each data point of $\{\mathbf{X},\Theta\}$, find all its Voronoi neighbors and mark the data point if

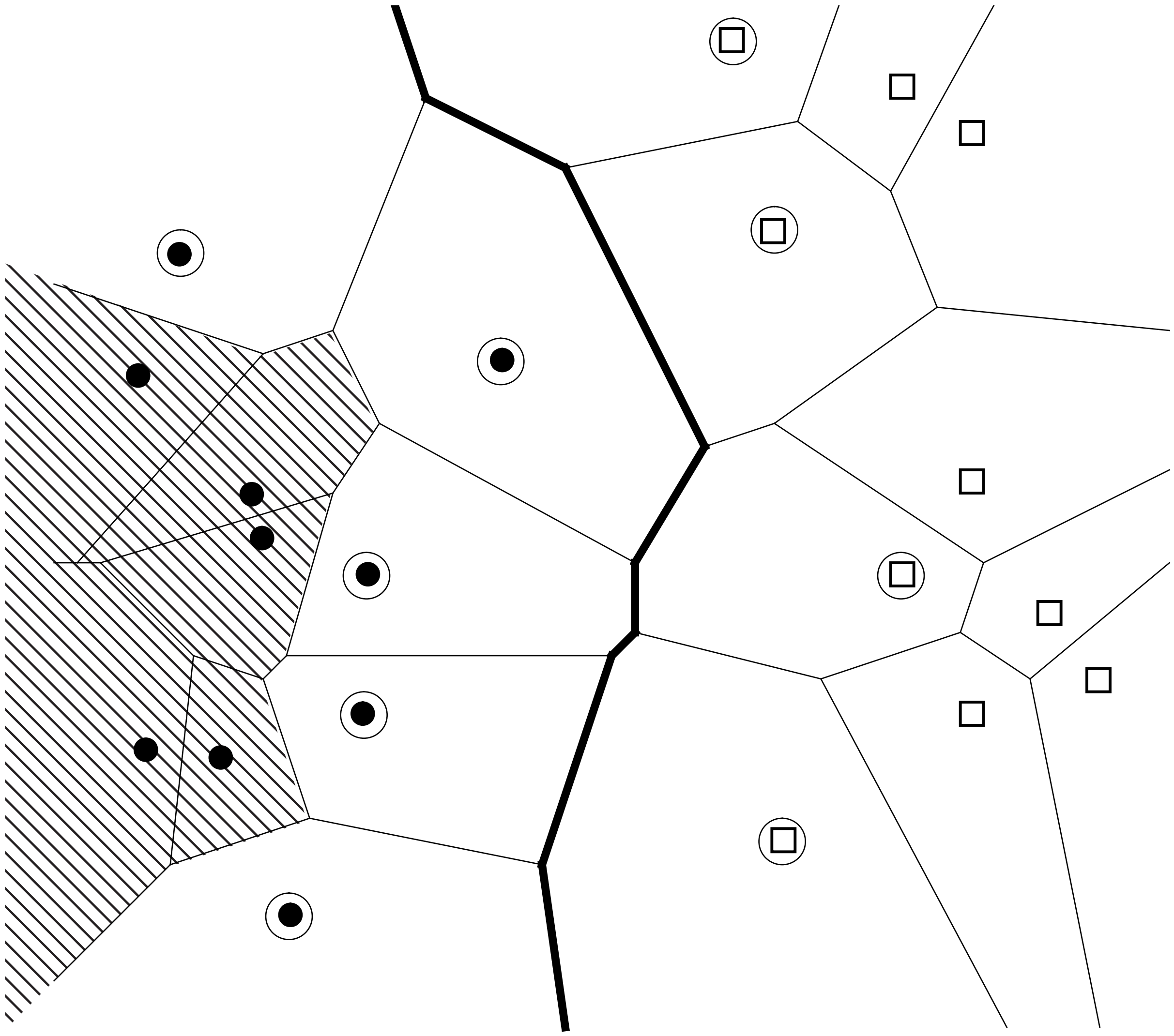

Fig. 2: The Voronoi diagram of the data points of Fig. 1 contains the *NN*-boundary. Each circled point indicates that at least one of its Voronoi neighbors belongs to a different class. The thick lines form the exact *NN*-boundary. A connected component of the union of Voronoi regions of data points from one class that are to be discarded is shown shaded.

all its Voronoi neighbors are not from the same class as that of the visited point.

Step 3: Discard all points that are not marked.

Step 4: Exit with the marked points as the Voronoi edited set.

End

The marked (circled) points in Fig. 2 are shown with their Voronoi diagram in Fig. 3 and constitute the Voronoi edited set of the data points of Fig. 1. For this example we see in Fig. 3 that the Voronoi edited set, which is a subset of the reference set $\{\mathbf{X},\Theta\}$, maintains the original *NN*-boundary exactly. When the *NN*-rule based on an edited set implements the same decision boundary as the *NN*-rule based on the full training set $\{\mathbf{X},\Theta\}$, we say that the edited set is *decision-*

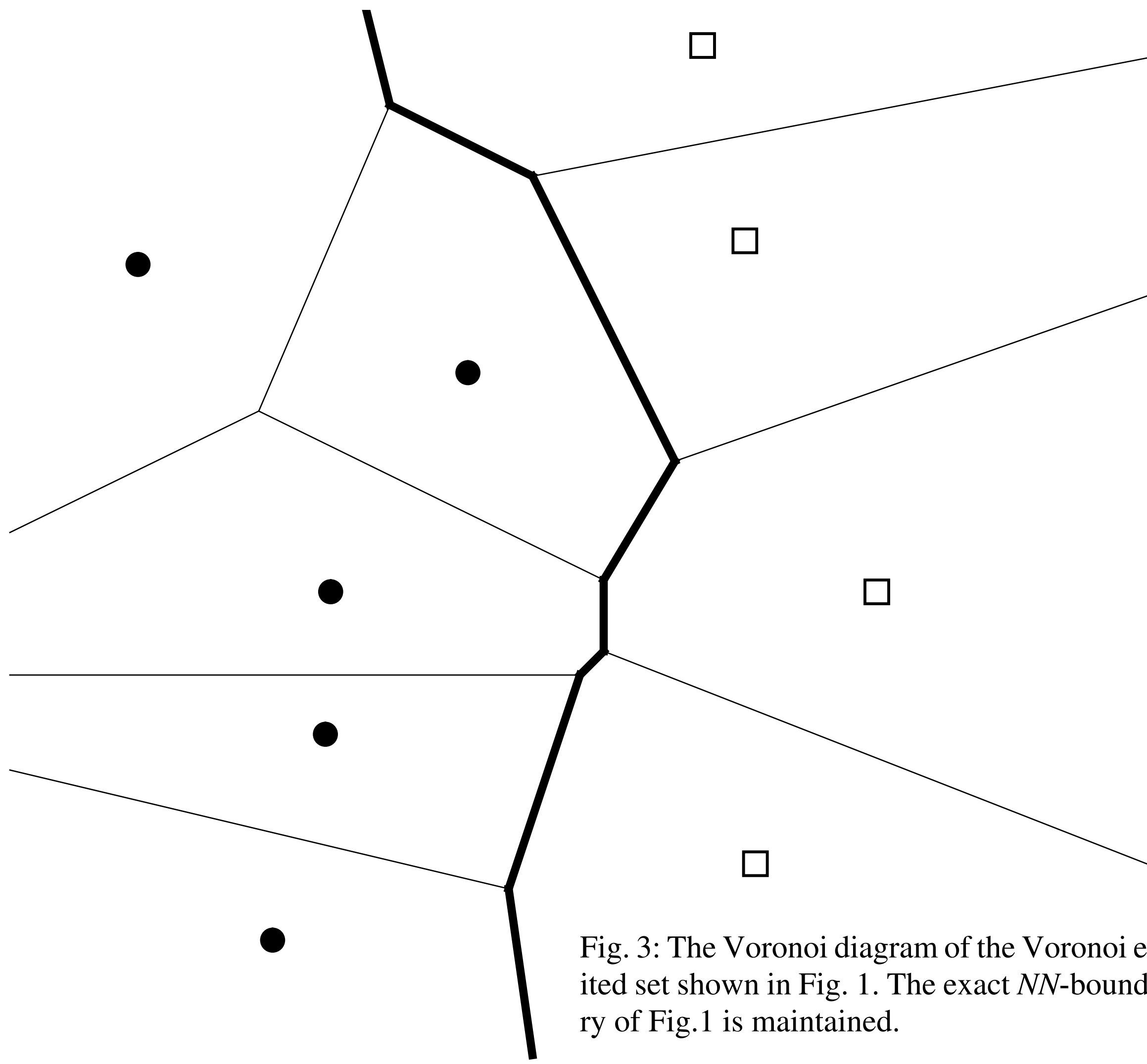

Fig. 3: The Voronoi diagram of the Voronoi edited set shown in Fig. 1. The exact *NN*-boundary of Fig.1 is maintained.

boundary consistent. We now prove that for Voronoi editing this is always the case.

Let $\{D\}$ denote the set of points of $\{\mathbf{X},\Theta\}$ which are discarded and let $\{\mathbf{X},\Theta\}_V$ denote the resulting Voronoi edited set.

Theorem 1: The Voronoi edited set of $\{X,\Theta\}$ is decision-boundary consistent.

Proof: (*by contradiction*) Assume that $\{\mathbf{X},\Theta\}_V$ is *not* decision-boundary consistent. This means that there exists a point $y \in \mathbf{R}^d$ that is classified into different classes with $\{\mathbf{X},\Theta\}$ and $\{\mathbf{X},\Theta\}_V$. Let $q \in \{X,\Theta\}$ be the nearest neighbor of y among $\{\mathbf{X},\Theta\}$ and $p \in \{X,\Theta\}$ be the nearest neighbor of y among $\{\mathbf{X},\Theta\}_V$. Let C denote the hypersphere with center y and radius yp. Note that only elements of $\{D\}$ may lie in C. We now determine another hypersphere C' which has the following properties:

(1) C' is contained in C,

(2) the center of C lies on the line segment yp,

(3) C' contains no points of $\{\mathbf{X},\Theta\}$ in its interior,

(4) C' passes through p and r, where r is an element of $\{D\}$.

Note that, by assumption, r must exist because $\{D\}$ contains at least the element q. Because of (3) and (4), p and r are Voronoi neighbors. It follows that if the labels of p and r were different r could not have been discarded, thus contradicting the fact that r is an element of $\{D\}$. Therefore we conclude that p and r have the same class labels. Note that if $r=q$ we have a contradiction because q and p have different class labels by the assumption that $\{\mathbf{X},\Theta\}_V$ is not decision boundary consistent. Therefore we must have that r is a data point different from q.

Let us now consider the hypersphere C^* with y as its center and yr as its radius. Since q is the nearest neighbor of y among $\{D\}$, it must lie in C^*. Since $C^* \subseteq C$, only data points in $\{D\}$ may lie in C^*. We have established above that the labels of p and r are the same but the labels of q and p are different. Therefore the labels of r and q are different.

We now repeat the above process, i.e., we determine another empty hypersphere which now passes through r and another data point, say s, and that is contained in C^*. By the same arguments as above s will have the same class label as r and therefore p. If $s=q$, we thus arrive at a contradiction for the class labels. We are therefore forced to conclude that s is different from q. If we continue this process we will eventually run out of data points of $\{D\}$ and will eventually choose a point equal to q resulting in the final contradiction. Q.E.D.

The following two corollaries are immediately implied by the above theorem.

Corollary 1: The Voronoi edited set is reference-set consistent.

Corollary 2: The Voronoi edited set is independent of the order in which the data is processed.

The worst-case complexity of the algorithm to obtain the Voronoi edited set from the reference set containing n data points in d-space is $O(n^{[d/2]+1}) + O(d^3 n^{[d/2]} \log n)$ where $[d/2] = k$ when $d=2k$ or $2k-1$ using the algorithm of Avis and Bhattacharya [AB83]. A more efficient but complicated algorithm exists [Se86]. This compares favorably with the algorithm of Dasarathy and White [DW78] whose worst-case complexity to generate the NN-boundary is $O(dn^{d+2})$. However, the Voronoi thinned set is not minimal as is illustrated in Fig. 4. Let the points denoted by '•' belong to class 1 and the points denoted by '❐' belong to class 2. The Voronoi diagram of this set is also shown in Fig. 4. It is easy to see that the Voronoi edited set keeps all the data points in $\{\mathbf{X},\Theta\}$ but no more than two points are sufficient to implement the same decision boundary. It should be pointed out however that the data in Fig. 4 is pathological. Indeed, when the points are in *general position*, Voronoi editing yields a *minimal* decision boundary consistent set. The points of S in $\mathbf{R}^d$ are said to be in general position provided that the following conditions are satisfied: (1) at most d data points may lie in a d-dimensional hyperplane, (2) at most d+1 data points may lie in a d-dimensional hypersphere, and (3) the perpendicular bisecting hyperplanes between each two distinct

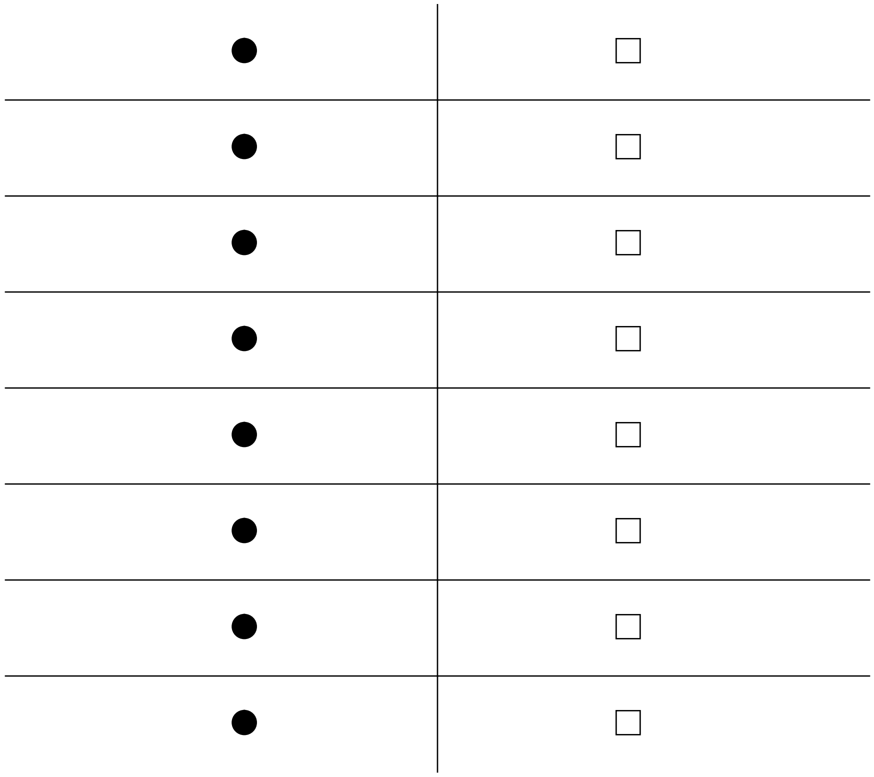

Fig. 4: A pathological example illustrating that the Voronoi edited set need not be minimal.

pairs of data points, are also distinct.

Theorem 2: If the data $\{\mathbf{X},\Theta\}$ are in general position then the Voronoi edited set $\{\mathbf{X},\Theta\}_V$ is a minimal-size decision-boundary consistent set.

Proof: Consider a point (X_k,θ_k) belonging to class C_i in the Voronoi edited set $\{\mathbf{X},\Theta\}_V$. It must therefore have at least one Voronoi neighbor, say (X_l,θ_l) belonging to class C_j, for j not equal to i, and a portion of the bisecting hyperplane between X_k and X_l must be in the *NN*-decision boundary implied by $\{\mathbf{X},\Theta\}_V$. Assume $\{\mathbf{X},\Theta\}_V$ is not a minimal size set. This implies that a face (a portion of the bisecting hyperplane between X_k and X_l) of the *NN*-decision boundary can be linearly extended to cover an adjacent face. But this implies that $\{\mathbf{X},\Theta\}$ is not in general position Q.E.D.

It is worth emphasizing that the above results imply that the performance of the nearest neighbor classifier using the reduced set $\{\mathbf{X},\Theta\}_V$ is identical to that which uses the full training set $\{\mathbf{X},\Theta\}$ and contrary to claims often made in the literature. For example Yan [Ya93] states that "unfortunately, a reduction in the number of training samples used as prototypes always causes a degradation of the performance of the classifier."

2.3 Drawbacks of the Voronoi Editing Algorithm

If we look at Fig. 2 it is observed that the two top-most marked data points, one from

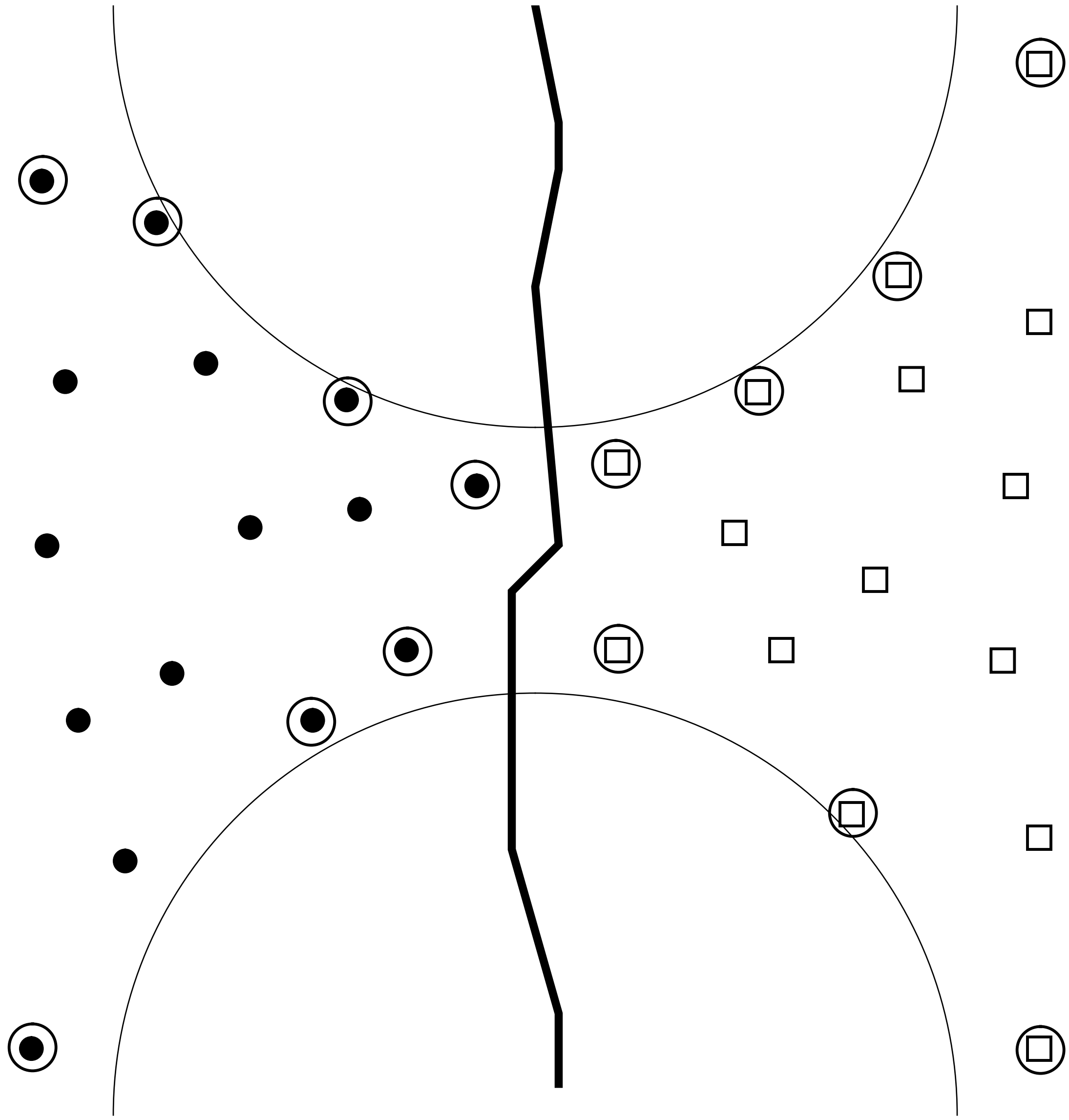

Fig. 5: Illustrating the situation when Voronoi editing keeps more data points than are necessary for good performance in practice.

each class, are kept in the Voronoi edited set, even though they are well separated. This situation is more clearly illustrated in Fig. 5, where 30 data points were uniformly distributed in the unit square between the circles

$$(x - 0.5)^2 + y^2 = (0.4)^2$$

and

$$(x - 0.5)^2 + (y - 1)^2 = (0.4)^2$$

After the application of the Voronoi editing algorithm, the marked data points were kept in the Voronoi edited set. Note that the Voronoi edited set also maintains the *NN*-boundary outside the region of interest which, for all practical purposes, is not necessary. Thus, the Voronoi editing algorithm treats all regions of the feature space with equal importance. If the unknown data point to be classified comes from the same distribution as the sample points of the reference set, the *NN*-boundary outside the region of interest is of very little importance. Therefore, all those data points in the Voronoi edited set, which only maintain the *NN*-boundary outside the “region of interest”, could be neglected.

Furthermore, the construction of the Voronoi diagram in high dimensions is still a time consuming process [AB83]. Any algorithm in the worst case will take at least $O(n^{[d/2]})$ time [Kl80]. These drawbacks of Voronoi editing from the practical point of view lead us to consider alternative approximate methods to which we now turn

3. Gabriel Editing

3.1 The Gabriel Editing Algorithm

The Gabriel editing algorithm is similar in spirit to the Voronoi editing algorithm except for the fact that the Gabriel editing algorithm, as the name suggests, uses the Gabriel graph of the reference set $\{\mathbf{X},\Theta\}$ instead of the Voronoi diagram. The Gabriel graph is defined as follows. For each pair of points (p_i, p_j) in the reference set $\{\mathbf{X},\Theta\}$, construct the *diametral* sphere, denoted by, $S(p_i, p_j)$, i.e., the sphere such that (p_i, p_j) forms the diameter of $S(p_i, p_j)$. Two points (p_i, p_j) are said to be *Gabriel neighbors* if $S(p_i, p_j)$ is empty, i.e., if no other points of $\{\mathbf{X},\Theta\}$ other than p_i and p_j lie in $S(p_i, p_j)$. The *Gabriel graph* is obtained by joining a pair of points with an edge if they are Gabriel neighbors. For further properties and algorithms for computing the Gabriel graph the reader is referred to [MS80] and [Ur83]. We now describe the Gabriel editing algorithm.

<u>Algorithm Gabriel Editing</u>

Begin

Step 1: Construct the Gabriel graph of the reference set $GG\{\mathbf{X},\Theta\}$.

Step 2: Visit each node of $GG\{\mathbf{X},\Theta\}$ and mark the visited node if all its Gabriel neighbors are not from the same class as the node visited.

Step 3: Discard all data points in $\{\mathbf{X},\Theta\}$ corresponding to nodes in $GG\{\mathbf{X},\Theta\}$ that are not marked.

Step 4: Exit with the marked data points as the Gabriel edited set.

End

Figures 6 and 7 illustrate the results obtained with this algorithm. Fig. 7 also shows the *NN*-boundary determined by the Gabriel edited set. This boundary, when compared with the *NN*-boundary determined by the original reference set $\{\mathbf{X},\Theta\}$, differs mainly in the region outside of the convex hull of $\{\mathbf{X},\Theta\}$, which is usually of not much interest for the classification problem. For comparison the Gabriel editing algorithm is also applied to the reference set given in Fig. 5. The Gabriel edited set and the corresponding *NN*-boundary are shown in Fig. 8. It is observed that the

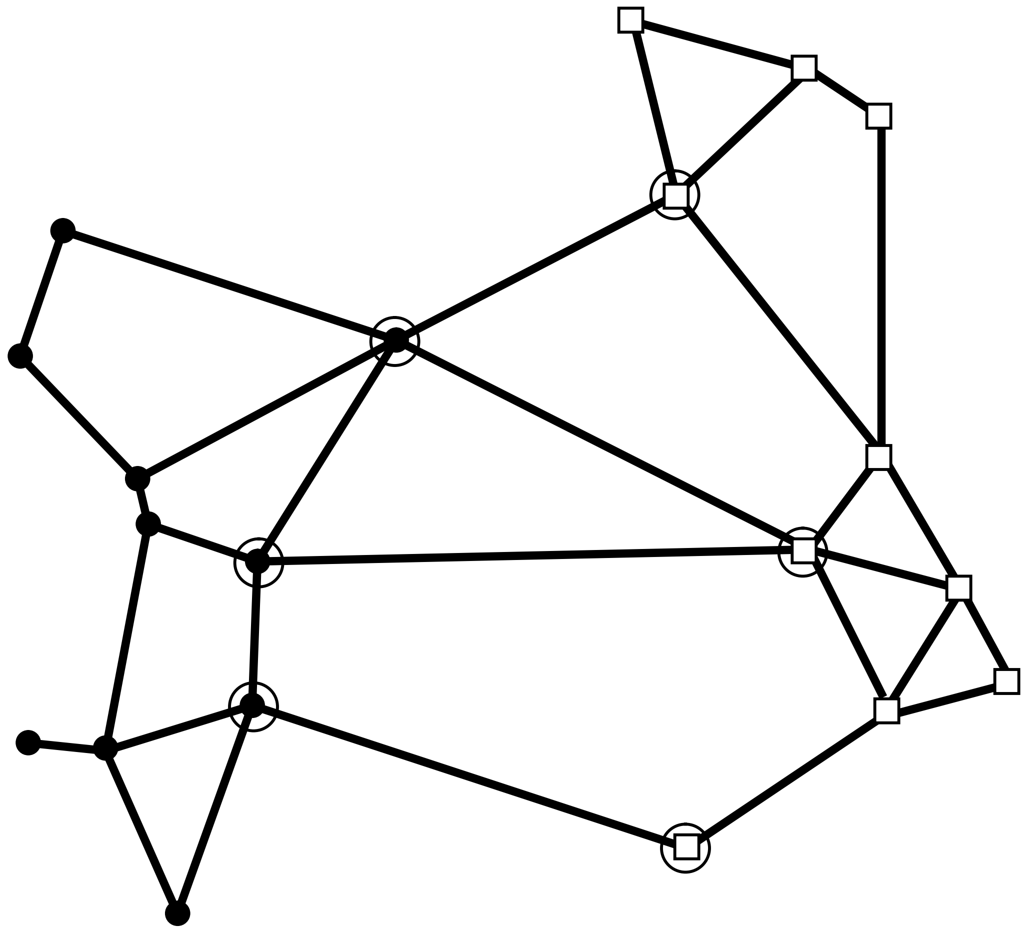

Fig. 6: The Gabriel graph of the data set shown in Fig. 1. Each marked data point indicates that at least one of its Gabriel neighbors is from a class different than its own.

Gabriel editing algorithm has completely ignored those points of the reference set which maintain the *NN*-boundary outside the region of interest.

The Gabriel edited set is always a subset of the Voronoi edited set because of the fact that a pair of data points which are Gabriel neighbors are also Voronoi neighbors [MS80]. Therefore, it can be said that the Gabriel editing algorithm reduces the Voronoi edited set further. However it is clear that the Gabriel edited set is not decision boundary consistent. From Fig. 9, it is seen that the Gabriel edited set is also not reference-set consistent. On the other hand, as we shall see later, this phenomenon does not appear to degrade the performance of the Gabriel edited set in practice.

3.2 Construction of the Gabriel Graph in *d*-Space

3.2.1 The brute force method

The construction of the Gabriel graph of a planar set of points has been described in [MS80]. This method uses the Voronoi diagram construct as the preprocessing step. Since our

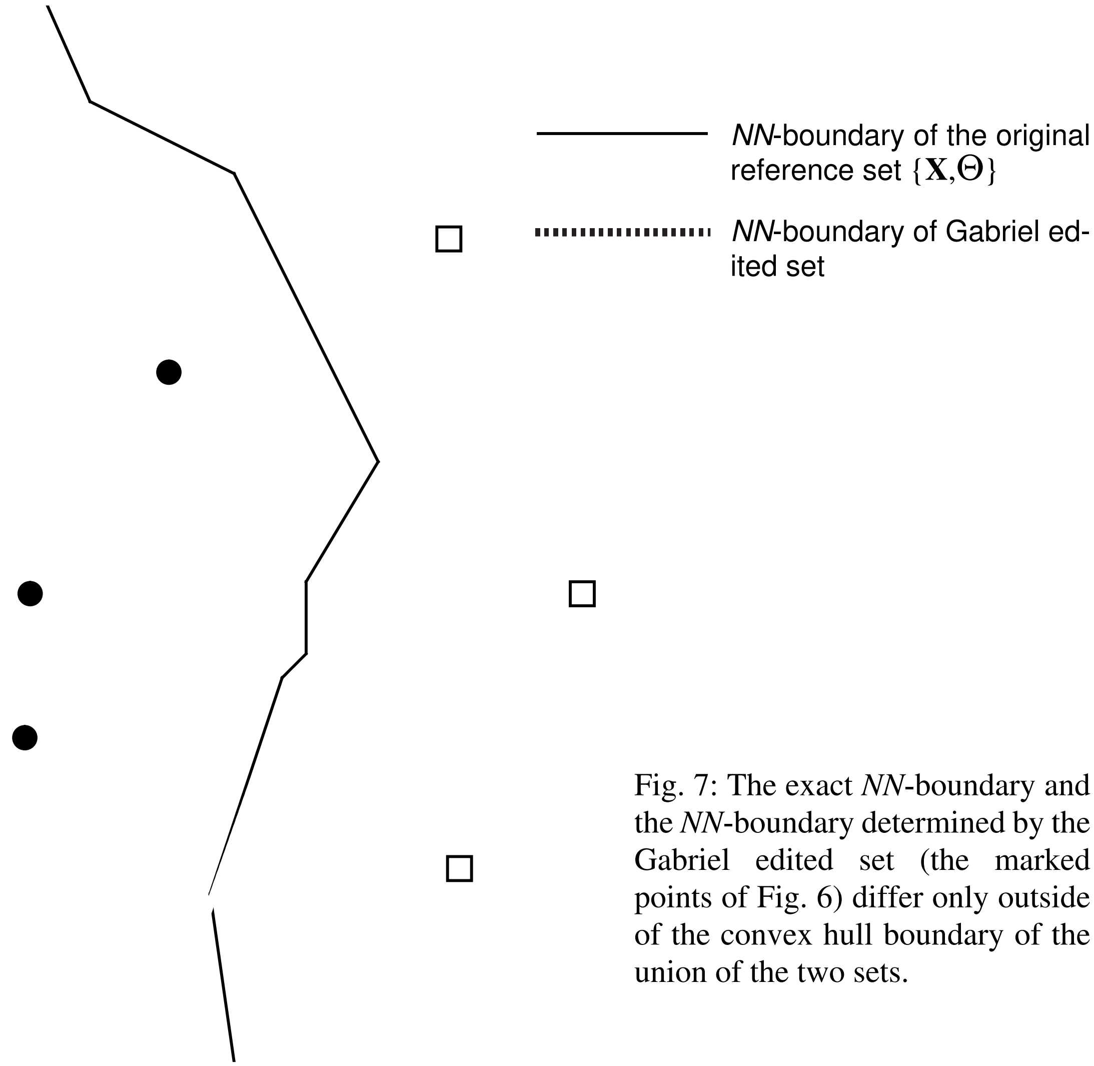

Fig. 7: The exact *NN*-boundary and the *NN*-boundary determined by the Gabriel edited set (the marked points of Fig. 6) differ only outside of the convex hull boundary of the union of the two sets.

main intention is to construct the Gabriel graph efficiently in higher dimensions, the use of the Voronoi diagram construct is not desirable. Therefore, we present an algorithm to construct the Gabriel graph in *d*-space which does not require computing the Voronoi diagram.

By definition, two data points of a set are Gabriel neighbors if and only if their sphere of influence (the diametral sphere) is empty. We can always construct the Gabriel graph once all the Gabriel neighbors of the set are known. The Gabriel neighbors of a set can be determined exhaustively by using the brute-force method. Let our given set of data points be $\{\mathbf{X}\} = \{X_1, X_2, ..., X_n\}$. Then the key steps of the brute-force method are:

Step 1: Consider all the pairs of points (X_i, X_j), for i,j=1,2,...,n; i < j.

Step 2: For each such pair (X_i, X_j) test whether there exists a point X_k, $k \neq i,j$, belonging to $\{\mathbf{X}\}$ such that:

If such a point does not exist, X_i and X_j are Gabriel neighbors.

Step 1 of the algorithm requires $O(n^2)$ operations to yield $O(n^2)$ pairs. For each such pair

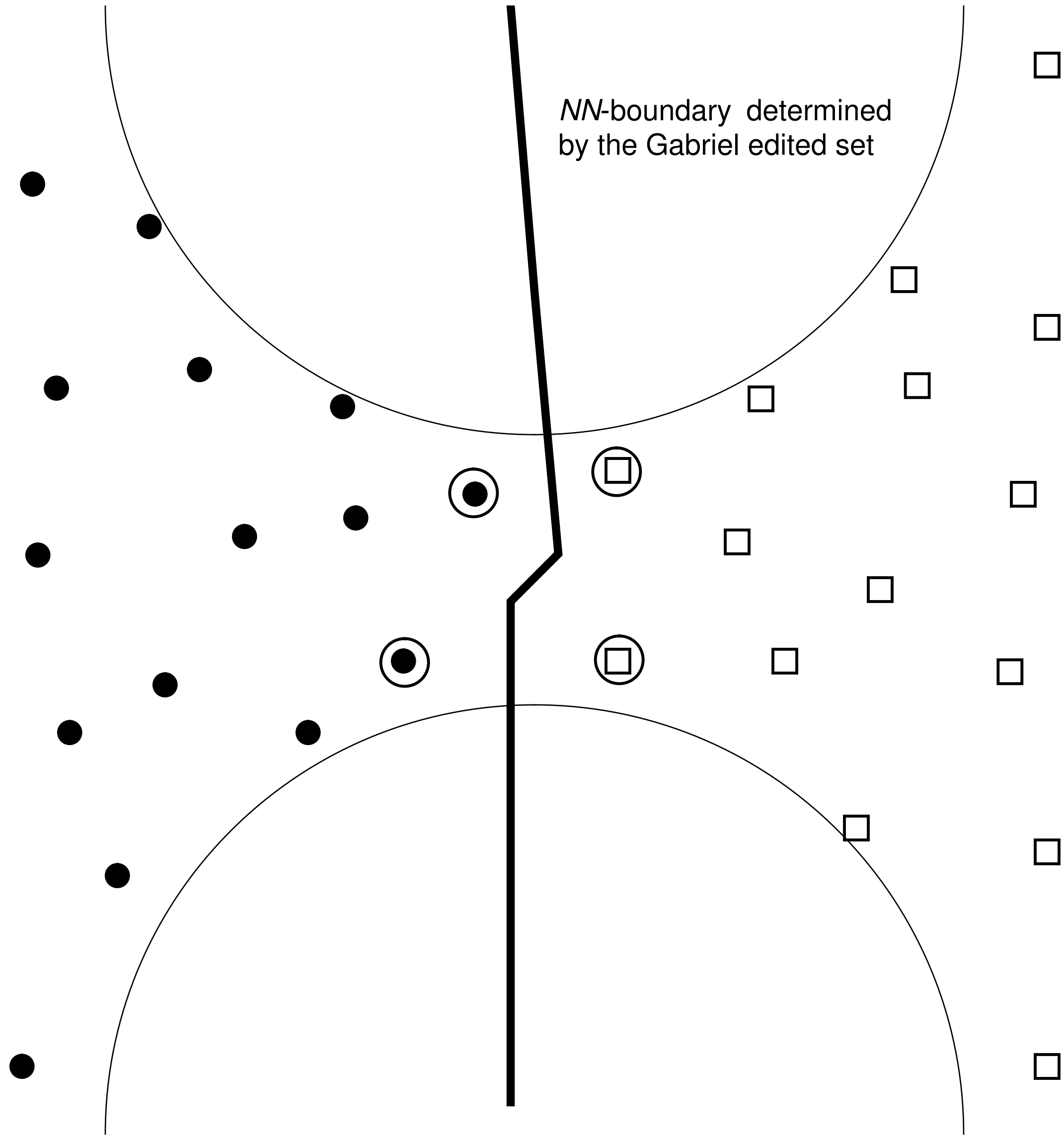

Fig. 8: The Gabriel edited set of the data points used in Fig. 5 consists of only four points which are marked. The *NN*-boundary determined by this set maintains the exact *NN*-boundary in the region of interest only.

$$d^2(X_i, X_j) > d^2(X_i, X_k) + d^2(X_j, X_k)$$

of points (X_i, X_j), step 2 requires O(nd) operations. Hence the overall complexity of the algorithm is O(dn^3)

Thus the complexity of the brute-force method is primarily dependent on the number of data points of the set. This is not so when the Voronoi diagram is used to compute the Gabriel graph because in that situation the worst-case complexity is at least O($n^{[d/2]}$). Thus, for large d the brute force method, with a complexity of O(dn^3), is much faster than the method that uses the Voronoi diagram. However, if n is very large O(dn^3) may still be prohibitive. Therefore we now present a heuristic method which is practical. The expected complexity of the heuristic method is believed

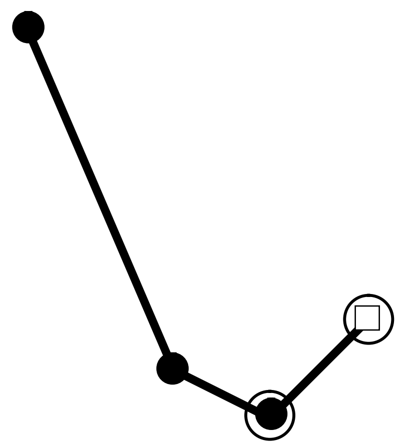

Fig. 9: Illustrating an instance when the Gabriel edited set is not reference-set consistent. The reference set consists of four data points where those denoted by '●' are from class 1 and the one denoted by '❐' belongs to class 2. The circled data points constitute the Gabriel edited set.

to be closer to O(dn^2) although a theoretical analysis has yet to be carried out.

3.2.2 A heuristic method

The number of pairs of Gabriel neighbors in a set of n points is, in general, very much less than the total number of pairs, $n(n-1)/2$, considered in the brute-force method. For example, in Fig. 5 there are 31 pairs of Gabriel neighbors out of 190 possible pairs. Thus if by some means we can reduce the number of pairs to be tested for Gabriel neighbors then the brute-force method will be more efficient. A heuristic approach to achieve this goal is described below.

For simplicity we describe the method for the two dimensional case. Generalization to higher dimensions is straight forward. Let us consider the set of points shown in Fig. 10. Let p be a data point of the set whose Gabriel neighbors we are interested in computing. Consider a point q belonging to {**X**}. Draw a line $B(p,q)$ through q, which is perpendicular to the line joining p to q. Let LH(B,p) be the half-space, determined by $B(p,q)$, which contains the point p. Let RH(B,p) be the half-space which does not contain the point p. We then have the following lemma.

Lemma: No data point contained in the set in RH(B,p) can be a Gabriel neighbor of p.

Proof: Consider the point p and some point r in RH(B,p). Let $D(p,r)$ denote the diametral disc determined by p and r. Since point r is contained in RH(B,p) it follows that angle p,q,r is greater than 90 degrees. Therefore point q is contained in $D(p,r)$ and r cannot be a Gabriel neighbor of p. Q.E.D.

Using the above heuristic, the brute-force method can then be improved as follows.

Algorithm Gabriel Editing

Begin

Consider each point p_i of the given set {**X**} separately and **do** the following:

Step 1: Start with $\mathbf{N}_i = \{p_1, p_2, ..., p_{i-1}, p_{i+1}, ..., p_n\}$ as the set of potential Gabriel neighbors of the

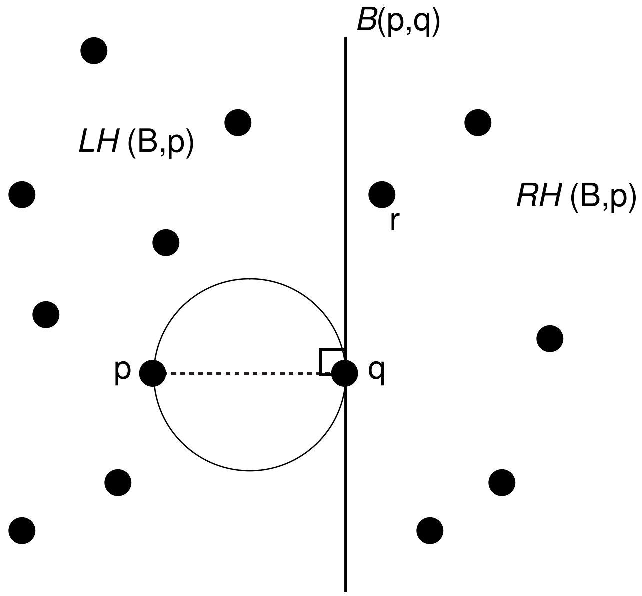

Fig. 10: When testing q to determine if it is a Gabriel neighbor of p a point r is first tested for containment in RH(B,p). All such points are deleted from the list of candidates for possible Gabriel neighbors of p because q lies in their discs.

point p_i.

Step 2: For each potential Gabriel neighbor p_r belonging to $\mathbf{N}_i$ do the following:

For every point p_k of $\{\mathbf{X}\}$, $p_k \neq p_i \neq p_r$:

i) Test whether p_k lies inside the sphere of influence determined by p_i and p_r. If so, remove p_r from the set $\mathbf{N}_i$ and go to Step 1 with a new potential Gabriel neighbor.

ii) If p_k is contained in $\mathbf{N}_i$, then test whether p_r lies inside the circle of influence determined by p_i and p_k. If so, remove p_k from the set $\mathbf{N}_i$.

Step 3: Accept the remaining points of the set $\mathbf{N}_i$ as the Gabriel neighbors of p_i.

End

3.2.3 Monte Carlo Simulation

A Monte Carlo simulation was carried out to determine the extent to which the heuristic method rejects pairs of data points. The experiment was performed by generating sets of data points of sizes 100, 300, 500, 700 and 1000 uniformly distributed in the unit d-cube, d=2, 3 or 4. Each case was repeated 20 times and the average value is shown in Table 1. Let T denote the total number of pairs rejected. The percent of pairs rejected, in a set of n points, denoted by $P_r(n)$, was calculated as follows:

$$P_r(n) = \left(\frac{T}{0.5n(n-1)}\right) \times 100$$

From Table 1, it is observed that most of the pairs are rejected before they are tested. For example, when n=500 and d=3, on the average, 119,120 pairs (out of 124,750) need not be considered at all for the Gabriel neighborhood test. For a particular dimension, the percent of pairs rejected increases as n increases, while for a particular number of points, the percent of pairs rejected

Number of Points	Percent of Pairs Rejected		
	$d = 2$	$d = 3$	$d = 4$
100	91.37	86.01	80.51
300	96.23	93.45	90.41
500	97.46	95.49	93.30
700	98.04	96.49	94.67
1000	98.43	97.31	95.92

Table 1: Monte Carlo simulation result to determine the percent of pairs of data points not considered for the Gabriel neighborhood test.

decreases as the dimension of the points increases.

Therefore it can be concluded that for high dimensions the brute force method of computing the Gabriel neighbors when modified heuristically, saves a lot of computation over the naive brute force method and the method via the Voronoi diagram.

4. Relative Neighborhood Graph Editing

4.1 Introduction

We have seen that the Gabriel editing algorithm reduces the Voronoi edited set. It is thus logical to extend this concept further, i.e., to further reduce the Gabriel edited set. One way of accomplishing this is to use the same idea on a sub-graph of the Gabriel graph. The editing algorithm discussed in this section is based on the geometrical construct known as the relative neighborhood graph (RNG), a graph first investigated in [To80] for the purpose of extracting the shape of a set of points in the plane. Since then much work has been done on relative neighborhood graphs and their relatives in two and higher dimensions. For a survey of the results known about RNG's the reader is referred to [JT92].

4.2 The Relative Neighborhood Graph

Let $\{\mathbf{X}\}$ be a set of n points in d-space: $\{\mathbf{X}\} = \{X_1, X_2, ..., X_n\}$. Two points X_i and X_j are defined as being "relatively close" if for k=1,2,...,n; k not equal to i:

The relative neighborhood graph is obtained by constructing an edge between points X_i and X_j for all i,j = 1,2,...,n; i not equal to j, if X_i and X_j are relatively close. Fig. 11 shows the RNG of

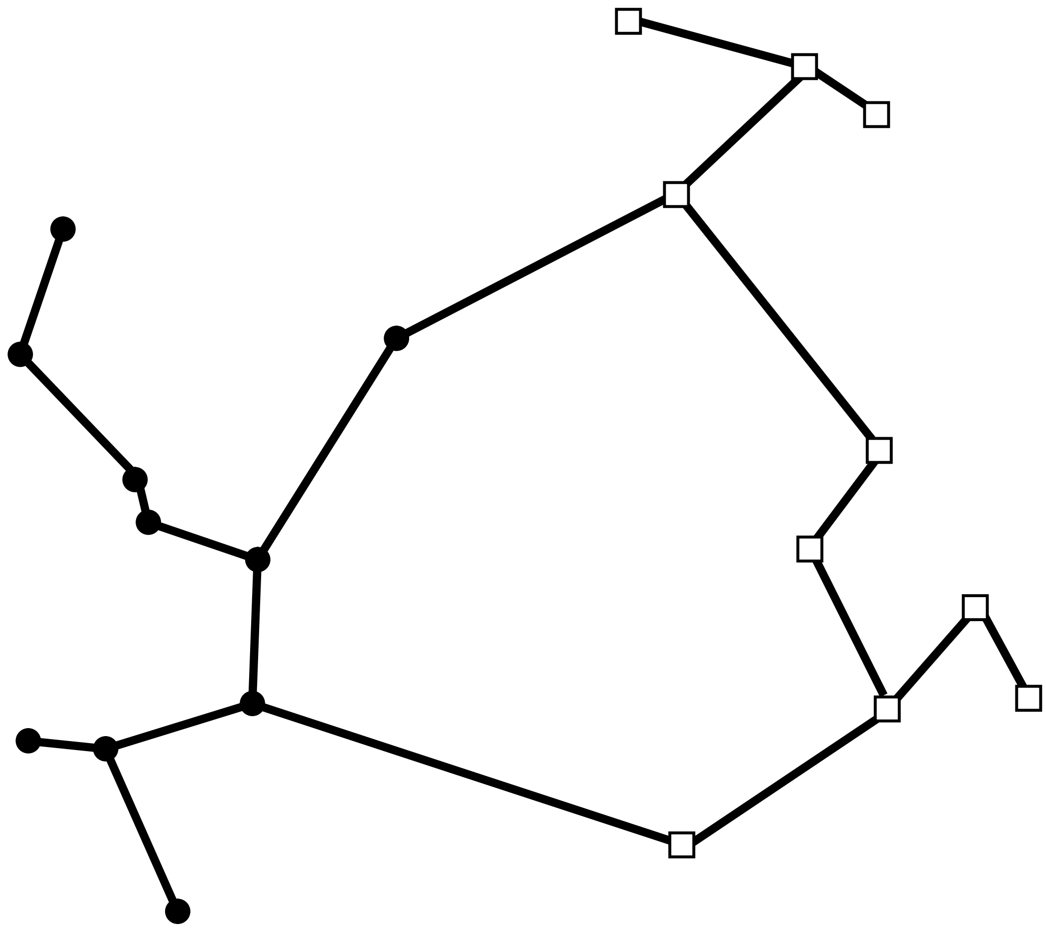

Fig. 11: The RNG of the data points of Fig. 1.

$$d(X_i, X_j) \leq max[d(X_i, X_k), d(X_j, X_k)]$$

the set of points given in Fig. 1.

4.3 The RNG Editing Algorithm

The RNG editing algorithm is similar to the Gabriel algorithm and thus we leave out the details. When it is applied to the reference set, given in Fig. 1, the edited reference set obtained is shown in Fig. 12. This reduced set, called the RNG edited set, is a subset of the Gabriel set [Figs. 7 and 12]. Therefore, we notice that

RNG edited set $\supseteq$ Gabriel edited set $\supseteq$ Voronoi edited set $\supseteq \{\mathbf{X},\Theta\}$

The *NN*-boundary generated by the RNG edited set, when compared with the *NN*-boundary, determined by the Voronoi edited set (and hence by the original reference set) [Fig. 12], differs considerably. Since the RNG edited set is contained in the Gabriel edited set, the RNG edited set

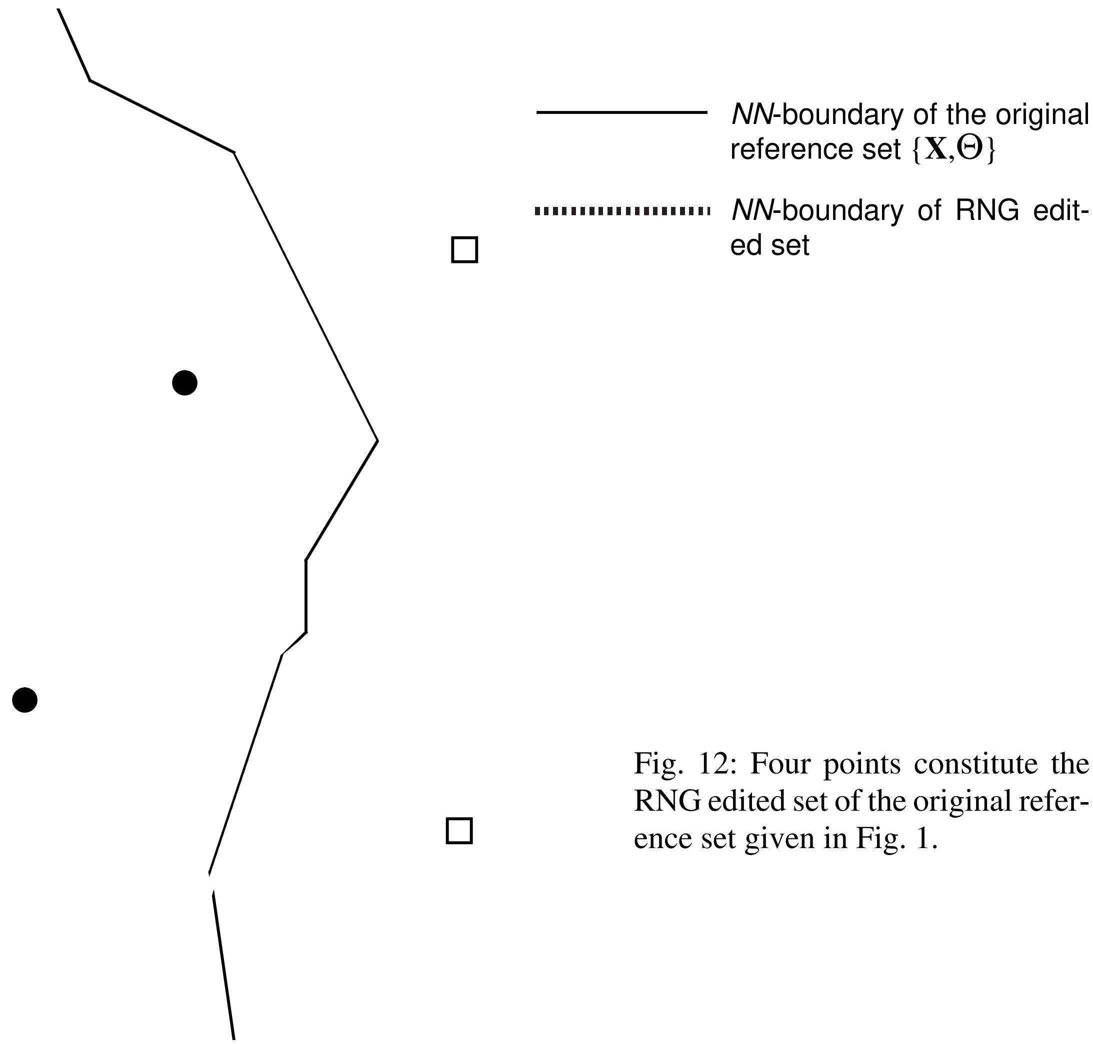

Fig. 12: Four points constitute the RNG edited set of the original reference set given in Fig. 1.

is neither decision boundary consistent nor reference set consistent.

The RNG of a set can be constructed by first computing the set of all relative neighbors exhaustively. The brute-force method similar to the one for the Gabriel graph can be used. The brute-force method can also be improved using a similar heuristic described earlier.

5. Experimental Results

5.1 Introduction

The three editing algorithms were compared experimentally to determine the number of points deleted and the resulting error rate in each case. Several Monte Carlo simulations were performed with different distributions and varying dimensions for both synthetic and real world data sets. We report here only two experiments with real world data in the interest of brevity. The con-

Table 2	Original Set	Voronoi Edited Set	Gabriel Edited Set	RNG Edited Set
Size	150	109	39	21
NN-error	1.3%	1.3%	1.3%	2.1%
Variance	0.31E-4	0.31E-4	0.31E-4	0.81E-4

Table 2: The Iris Data: 150 data points in 4-space

Table 3	Original Set	Voronoi Edited Set	Gabriel Edited Set	RNG Edited Set
Size	1999	1313	820	452
NN-error	5.9%	5.9%	6.0%	9.7%
Variance	0.71E-4	0.71E-4	0.67E-4	0.45E-4

Table 3: The Cervical Cell Data: 1999 points in 4-space

clusions for synthetic data are strong and the same as those reported here.

5.2 Iris Data

The so called Iris data consist of four measurements made on each of 150 flowers. There are three pattern classes, Virginica, Setosa, and Versicolor corresponding to three different types of Iris. Therefore, in this case the reference set consists of 150 feature vectors in 4-space each of which is assigned to one of the three above mentioned classes. This data was first collected by R.A. Fisher [Fi36] in 1936 and since then have become somewhat of a classic “text book” example on which to try out ideas and algorithms. Obtaining an estimate of the performance or error rate of a decision rule is a field unto itself [To74], [RJ91]. In these experiments, the *NN*-error rate was estimated using Effron’s [Ef79] bootstrap method with a uniform window. We first select a feature vector $p \in \{X,\Theta\}$ at random. We place a rectangular window with p at its center. The size of the window is determined by the nearest neighbor of p. We then generate a random point uniformly distributed inside the window and the true class of the generated point is assumed to be that of p. In this way a new testing data set is generated. Finally 200 such generated data sets are created and the results for each are averaged. The variance serving as a confidence interval is also computed.The bootstrap methods are considered the best estimators of the performance of a classifier in the sense that not only do they provide estimates that are *unbiased* and have a low *variance* but they give an estimate (or confidence interval) for the variance as well. This allows for easy comparison of different experiments in a statistically significant manner [JDC87].

Table 2 shows the results when the editing algorithms are applied to the Iris data. As expected the *NN*-error rate using the Voronoi edited set is the same as the exact *NN*-error rate. 109 sample data points were selected by the algorithm to maintain the original *NN*-boundary exactly. The size of the Gabriel edited set is only 39, almost one third of the size of the Voronoi edited set. But the number of miss-classifications of the *NN*-rule on the unknown boot-strapped data using the Voronoi edited set (and hence the original reference set) and the Gabriel edited set are the same. On the other hand the RNG edited set increases the *NN*-error rate by almost 100%. Thus by removing 70 additional sample points from the Voronoi edited set, the performance of the *NN*-classifier remains the same, but if we remove only 18 additional sample data points from the Gabriel edited set the *NN*-error rate doubles. Therefore, the Gabriel edited set appears to contain sufficient information for discrimination of the Iris data with the *NN*-rule.

5.3 Cervical Cell Data

The Biomedical Image Processing Laboratory at McGill University has a data base of about 2000 cervical cell images which are assigned to one of 13 cell types (sub-classes). Eight of the types are considered to be subclasses of the normal cell class and the other five types are subclasses of the abnormal cell class [CPT77], [POCLT77], [OPTL79] and [OPT77].

The images were subjected to the preprocessing and feature extraction methods described in [POCLT77], [OPTL79] and [OPT77]. Each cell is represented by a four-dimensional feature vector using the features

(l) log (cytoplasm diameter/nucleus diameter),
(2) log (nucleus area),
(3) average cytoplasm density, and
(4) average nucleus density.

We have only considered the two-class problem - normal and abnormal classes. Thus our reference set contains 1999 samples in 4-space labelled either as normal or abnormal. The results of the editing algorithms are shown in Table 3.

As in the previous experiments, the Gabriel edited set contains fewer sample points than the Voronoi edited set even though the corresponding *NN*-error rates are not significantly different. The RNG edited set increases the *NN*-error rate considerably.

6. Concluding Remarks

We have exhibited in this paper several new methods for editing the data with the nearest neighbor decision rule (*NN*-rule) and compared them experimentally, with respect to (1) storage requirements, (2) computation time and (3) resulting probability of misclassification, to the exhaustive (full training set) rule. The new methods have several advantages over previous methods. The proposed approaches are based on well-known graph structures that are first computed on

$\{\mathbf{X},\Theta\}$. The graph structures are proximity graphs obtained from the Voronoi diagram of $\{\mathbf{X},\Theta\}$. The methods have the merit that they are exact and yield edited sets independent of the order in which the data are processed. Furthermore, one method yields edited sets which are not only both training-set and decision-boundary consistent but are minimal in size when $\{\mathbf{X},\Theta\}$ is in general position. The methods were compared empirically through experiments on synthetic data as well as real world data in the automatic detection of cervical cancer. Algorithms were given for obtaining the edited sets efficiently in practice.

Experiments have shown that by sacrificing the properties of reference set and decision boundary consistency, editing schemes can be obtained, such as Gabriel-graph-editing, that in practice keep far fewer data points with negligible deterioration in performance. It is tempting to be greedy here by further exploiting this idea and the RNG editing scheme is an unsatisfactory outcome of this greediness. However there may exist other graphs sparser than the Gabriel graph that will discard additional points without deterioration in performance. Such graphs have recently been explored in other contexts [Ur82], [KR85] and may well be fruitful for the editing problem in nonparametric decision rules. These graphs are presently being explored.

A final word concerning the performance of the *NN*-rule (*1-NN* rule) is in order. Some researchers may require a performance closer to the optimal Bayes error. It it known that the *k-NN* rule (when *k* is suitably chosen) will approximate the Bayes error. Some researchers may consider finding the *k* nearest neighbors computationally undesirable. We mention here that there is a simple method of approximating the *k-NN* rule with the *1-NN* rule that gives excellent performance in practice. It suffices to *re-label* the original data by classifying it with the *k-NN* rule. Then the *1-NN* rule is used on the re-labeled data and the editing methods discussed in this paper may be readily applied.

7. Acknowledgment

A preliminary version of this paper was presented at the 16th Symposium on the Interface of Computer Science and Statistics, Atlanta, Georgia, March 14-16, 1984.

8. References

[AB83] Avis, D. and Bhattacharya, B.K., "Algorithms for computing d-dimensional Voronoi diagrams and their duals," in *Computational Geometry*, Ed., F.P. Preparata, JAI Press, 1983, pp. 159-180.

[AM93] Arya, S. and Mount, D., "Approximate nearest neighbor queries in fixed dimensions," *Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, January 25-27, 1993, Austin, Texas.

[Au91] Aurenhammer, F., "Voronoi diagrams - A survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, No. 3, September 1991, pp. 345-405.

[BDF78] Brostow, W., Dussault, J.P. and Fox, B.L., "Construction of Voronoi polyhedra," *Jour-*

nal of Computational Physics, vol. 29, 1978, pp. 81-92.

[Bo81] Bowyer, A., “Computing Dirichlet tessellations,” *The Computer Journal*, vol. 24, No. 2, 1981, pp. 162-166.

[BR79] Brassel, K.E. and Rief, D., “A procedure to generate Thiessen polygons,” *Geographical Analysis*, vol.11, No. 3, 1979, pp. 289-303.

[Br79a] Brown, K.Q., *Geometric transforms for fast geometric algorithms*, Ph.D. Thesis, Carnegie-Mellon University, Department of Computer Science, 1979.

[Br79b] Brown, K.Q., “Voronoi diagrams from convex hulls,” *Information Processing Letters*, vol. 9, No. 5,1979, pp. 223-228.

[Ch74] Chang, C.-L., “Finding prototypes f or nearest neighbor classifiers, *IEEE Trans. Computers*, vol. C-23, November 1974, pp.1179-1184.

[CH67] Cover, T. M. and Hart, P.E., “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. IT-13, No.1, 1967, pp. 21-27.

[CPT77] Cahn, R.L., Poulsen, R.S. ant Toussaint, G.T., “Segmentation of cervical cell images”, *Journal of Histochemistry and Cytochemistry*, vol.25, 1977, pp. 681-688.

[De81] Devroye, L.P., “On the inequality of Cover and Hart in nearest neighbor discrimination,” *IEEE Transactions on Pattern analysis and Machine Intelligence*, vol. PAMI-3, January 1981, pp. 75-78.

[DW78] Dasarathy, B. and White, L.J., “A characterization of nearest neighbor rule decision surfaces and a new approach to generate them,” *Pattern Recognition*, vol. 10, 1978, pp. 41-46.

[Ed87] Edelsbrunner, H., *Algorithms in Combinatorial Geometry*, Springer-Verlag, Heidelberg, 1987.

[Ef79] Effron, B., “Bootstrap methods: another look at the jackknife,” The *Annals of Statistics*, vol.7, No.l, 1979, pp.1-26.

[Fi36] Fisher, R.A., “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, Part 2, 1936, pp. 179-188.

[FM84] Fukunaga, K. and Mantock, J.M., “Nonparametric data reduction,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-6, January 1984, pp. 115-118.

[FP70] Fisher, F.P. and Patrick, E.A., “A preprocessing algorithm for nearest neighbor decision rules,” *Proc. National Electronics Conf.*, Dec. 1970, pp. 481-485.

[Ga72] Gates, G.W., “The reduced nearest neighbor rule,” *IEEE Trans. Information Theory*, vol.IT-18, May 1972, pp. 431-433.

[GK79] Gowda, K.C. and Krishna, G., “The condensed nearest-neighbor rule using the concept of mutual nearest neighborhood,” *IEEE Transactions on Information Theory*, vol. IT-

25, No.4, 1979, pp. 488-490.

[GS78] Green, P.J. and Sibson, R., “Computing Dirichlet tessellations in the plane,” *The Computer Journal*, vol.21, No.2, 1978, pp. 168-173.

[Ha68] Hart, P.E., “The condensed nearest-neighbor rule,” *IEEE Transactions on Information Theory*, vol. IT-4, May 1968, pp. 515-516.

[JDC87] Jain, A. K., Dubes, R. C. and Chen, C.-C., “Bootstrap techniques for error estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-9, No. 5, September 1987, pp. 628-633.

[JT92] Jaromczyk, J. W. and Toussaint, G. T., “Relative neighborhood graphs and their relatives,” *Proceedings IEEE,* vol. 80, No. 9, September 1992, pp. 1502-1517.

[Kl80] Klee, V., “On the complexity of d-dimensional Voronoi diagrams,” *Arch. Math*., vol. 34, 1980, pp. 75-80.

[Kl89] Klein, R., *Concrete and Abstract Voronoi Diagrams*, Springer-Verlag, Heidelberg, 1989.

[MS80] Matula, D.W. and Sokal, R.R., “Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane,” *Geographical Analysis,* vol. 12, 1980, pp. 205-222.

[OPT77] Oliver, L.H., Poulsen, R.S. and Toussaint, G.T., “Estimating false positive and false negative error rates in cervical cell classification”, *Journal of Histochemistry ant Cytochemistry*, vol.25, 1977, pp. 696-701.

[OPTL79] Oliver, L.H., Poulsen, R.S., Toussaint, G.T. ant Louis, C., “Classification of atypical cells in the automatic cyto-screening for cervical cancer,” *Pattern Recognition*, vol.ll, 1979, pp. 205-212.

[POCLT77] Poulsen, R.S., Oliver, L.H., Cahn, R.L., Louis, C. and Toussaint, G.T., “High resolution analysis of cervical cells - a progress report”, *Journal of Histochemistry and Cytochemistry*, vol. 25, 1977, pp. 689-695.

[Ri75] Ritter, G.L., et al., “An algorithm for a selective nearest neighbor decision rule,” *IEEE Trans. Information Theory*, vol. IT-21, Nov. 1975, pp. 665-669.

[RJ91] Raudys, S. J. and Jain, A. K., “Small sample size effects in statistical pattern recognition: recommendations for practitioners,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, No. 3, March 1991, pp. 252-264.

[Se86] Seidel, R., “Constructing higher-dimensional convex hulls at logarithmic cost per face,” *Proc. 18th Annual ACM Symposium on the Theory of Computing*, 1986, pp. 404-413.

[Sh78] Shamos, M.I., *Computational geometry*, Ph.D. Thesis, Department of Computer Science, Yale University, May 1978.

[St77] Stone, C.J., “Consistent nonparametric regression,” *Annals of Statistics*, vol. 5, 1977 pp. 595-645.

[Sw72] Swonger, C.W., “Sample set condensation for a condensed nearest neighbor decision rule for pattern recognition,” in *Frontiers in Pattern Recognition*, Ed., S. Watanabe, Academic Press, 1972, pp. 511-526.

[To76a] Tomek, I., “Two modifications of CNN,” *IEEE Trans. Systems, Man and Cybernetics*, vol. SMC-6, Nov. 1976, pp. 769-772.

[To76b] Tomek, I., “An experiment with the edited nearest neighbor rule,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-6, June 1976, pp. 448-452.

[To74] Toussaint, G.T., “Bibliography on estimation of misclassification,” *IEEE Transactions on Information Theory*, vol.IT-20, No.4, 1974, pp. 472-479.

[To80] Toussaint, G.T., “The relative neighborhood graph of a finite planar set,” *Pattern Recognition*, vol.12, No.4, 1980, pp. 261-268.

[Ul74] Ullmann, V.R., “Automatic selection of reference data for use in a nearest-neighbor method of pattern classification,” *IEEE Trans. Information Theory*, vol. IT-20, July 1974, pp. 541-544.

[Wi92] Wilfong, G., “Nearest neighbor problems,” Technical Report, AT&T Bell Laboratories, Murray Hill, New Jersey, 1992.

[Ya93] Yan, H., “Prototype optimization for nearest neighbor classifiers using a two-layer perceptron,” *Pattern Recognition*, vol. 26, No. 2, 1993, pp. 317-324.

[YM91] Yau, H. and Manry, M. T., “Iterative improvement of a nearest neighbor classifier,” *Neural Networks,* vol. 4, 1991, pp. 517-524.