Proximity-Graph Instance-Based Learning, Support Vector Machines, and High Dimensionality: An Empirical Comparison

Godfried T. Toussaint[1], Constantin Berzan[2]

[1] Faculty of Science, New York University Abu Dhabi, Abu Dhabi, United Arab Emirates
gt42@nyu.edu
[2] Department of Computer Science, Tufts University, Medford, MA 02155, USA
constantin.berzan@tufts.edu

Abstract.
Previous experiments with low dimensional data sets have shown that Gabriel graph methods for instance-based learning are among the best machine learning algorithms for pattern classification applications. However, as the dimensionality of the data grows large, all data points in the training set tend to become Gabriel neighbors of each other, bringing the efficacy of this method into question. Indeed, it has been conjectured that for high-dimensional data, proximity graph methods that use sparser graphs, such as relative neighbor graphs (RNG) and minimum spanning trees (MST) would have to be employed in order to maintain their privileged status. Here the performance of proximity graph methods, in instance-based learning, that employ Gabriel graphs, relative neighborhood graphs, and minimum spanning trees, are compared experimentally on high-dimensional data sets. These methods are also compared empirically against the traditional *k*-NN rule and support vector machines (SVMs), the leading competitors of proximity graph methods.

Keywords: Instance-based learning, Gabriel graph, relative neighborhood graph (RNG), minimum spanning tree (MST), proximity graphs, support vector machines (SVM), sequential minimal optimization (SMO), machine learning

1 Introduction

Instance-based learning algorithms are among the most attractive methods used today in many applications of machine learning to a wide variety of pattern recognition problems. The quintessential instance-based learning algorithm is the well-known *k*-Nearest Neighbor (*k*-NN) rule, whereby a new unknown pattern is classified into the class most heavily represented among its *k* nearest neighbors present in the training set. Two of the most attractive features of this method are immediately evident: (1) its simplicity, and (2) the fact that no knowledge is required about the underlying distribution of the training data. Nevertheless, unanswered questions about the rule's performance left doubts among its potential users. In 1967 Cover and Hart [4]

showed that, under some continuity assumptions the number n of patterns in the training data becomes infinite, the asymptotic probability of misclassification of the 1-NN rule is at most twice the Bayes error. Furthermore Devroye showed that for all distributions, the asymptotic probability of misclassification of the k-NN rule approaches the Bayes error provided that k and n approach infinity, and the ratio k/n approaches zero [7-8]. These constraints are satisfied for example when $k = n^{1/2}$. Thus the ultimate classificatory power of the k-NN rule was finally firmly established. Despite this great news, resistance to using k-NN rules in practice persisted, fueled by several misconceptions, one being that all n points must be stored, thus requiring too much storage. It was pointed out in 1979 that the decision boundary of the 1-NN rule remains unchanged when the data points surrounded by Voronoi neighbors of the same class are discarded (in parallel) [23-24]. A second false claim frequently encountered is that in order to determine the nearest neighbor of an unknown query point X, the distances between X and *all* the n points in the training data must be computed. Today there exist a plethora of methods for avoiding such exhaustive search.

The above-mentioned false claims notwithstanding, in practice it is desired to reduce the size of the training set, or the concomitant memory of the data structure into which the training set is embedded, as much as possible, while maintaining a low probability of misclassification. Therefore much research has been devoted to this topic, yielding a cornucopia of different approaches [20]. One elegant and promising approach generalizes Voronoi (or Delaunay triangulation) editing [23] to incorporate more general proximity graphs [24]. In the latter approach, for any definition of proximity, data points with the property that all their proximity-graph neighbors belong to the same class are discarded (in parallel). Previous experiments with low-dimensional data sets have shown that proximity graph methods that used the Gabriel graph were among the best machine learning algorithms for pattern classification applications [2-3, 24, 28]. However, as the dimensionality of the data grows large, all data points in the training set tend to become Gabriel neighbors of each other, effectively forsaking the notion of proximity, and bringing the efficacy of this method into question [14]. We conjectured that for high-dimensional data, methods that use sparser graphs, such as relative neighbor graphs (RNG) or minimum spanning trees (MST), would yield better results by avoiding proximity-graphs with too many edges. In this paper we conduct experiments that confirm this hypothesis.

Here the performance of various proximity graph methods for instance-based learning is compared experimentally using Gabriel graphs, relative neighborhood graphs, and minimum spanning trees, on a group of high-dimensional data sets. These three graphs vary considerably in the number of neighbors they admit, thus allowing us to test the hypothesis. For completeness, these methods are also compared empirically against the traditional k-NN rule that does not condense the data, and the optimization-based support vector machine (SVM), a leading competitor of proximity graph methods that has also enjoyed widespread success in practice.

2 Proximity Graphs

Given a set *S* of $n \geq 3$ points in the plane, two points *a*, *b* are Gabriel neighbors if all other points in *S* lie in the exterior of the smallest circle that contains *a* and *b*, i.e., the circle with diameter determined by points *a* and *b*. The Gabriel graph of *S* is obtained by joining all pairs of points with an edge provided they are Gabriel neighbors of each other. Figure 1 (left) shows the Gabriel graph of a set of 42 points in the plane. Note that the graph is adaptive in the sense that the number of edges it contains (or alternately the number of bounded regions it encloses) may be large or small relative to the number of points in the set. This property allows proximity-graph decision rules to automatically vary the number and spatial location of the neighbors they utilize, as a function of the local density and structure of the data. The Gabriel graph in Figure 1 (left) contains 27 bounded regions.

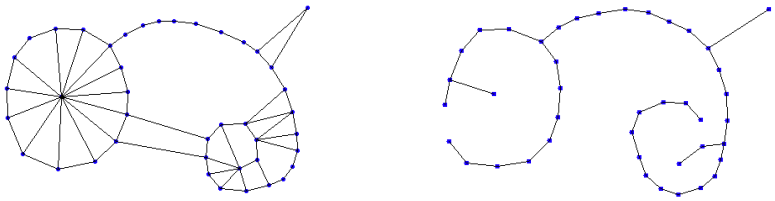


Fig. 1. The Gabriel graph (left) and the MST (right).

The properties of the Gabriel graph most relevant to the present research are that every minimum spanning tree (MST) of a set of points S is a sub-graph of the Gabriel graph of *S*, and that the Gabriel graph of *S* is a sub-graph of the Delaunay triangulation of *S* [21]. Devroye [6] has shown that for almost all probability density functions, as the number of points grows to infinity, the expected number of edges in the Gabriel graph approaches $2^{d-1}n$, where *d* is the dimensionality of the space.

The minimum spanning tree of a set of points is the sparsest connected proximity graph in terms of the number of edges it contains, whereas the Delaunay triangulation is much denser. Figure 1 (right) shows the MST for a similar set of points. Naturally, since this proximity graph is a tree it contains no bounded regions whatsoever. Figure 2 (left) shows the Delaunay triangulation of a similar set of points along with its dual, the Voronoi diagram.

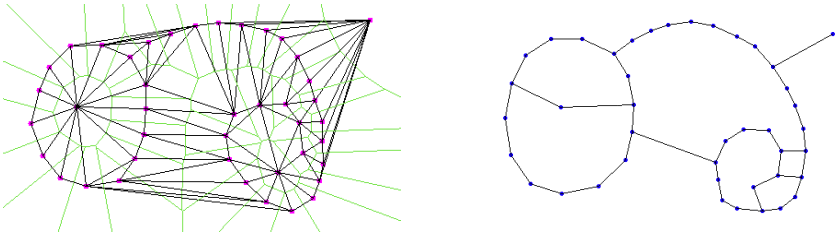


Fig. 2. The Delaunay triangulation with Voronoi diagram (left) and the RNG (right).

Toussaint [21] defined the relative neighborhood graph (RNG) of a set S of n points as the graph obtained by adding an edge between every pair of points a and b provided that $d(a, b) \leq \max [d(a, x), d(b, x)]$ for all x in S. The RNG is generally (for random configurations of points) much sparser than the Gabriel graph. Figure 2 (right) shows the RNG of 42 points arranged much like the points in Figure 1, but here the number of bounded regions is only 6. All four graphs constitute snapshots of a continuum of proximity graphs called β-skeletons that vary from the dense Delaunay triangulation to the sparse minimum spanning tree, [15]. These proximity graphs have been applied to a wide variety of problems in different fields [14].

3 Reducing the Size of the Training Data

The first published algorithm for reducing the size of the training data for use in nearest neighbor decision rules was the *condensed* nearest neighbor rule [12]. The purpose of this algorithm was to discard as much data as possible by removing data points that were "far" from the decision boundary, in a heuristic fashion, without knowing precisely where the decision boundary lay. However, although the condensed training set discarded many points (one of its best features), and obtained zero misclassification errors when classifying the original full training set, a property referred to as *training-set consistency*, the performance on separate testing data was modest.

In 1972, Wilson [26] proposed an algorithm that did the exact opposite of Hart's algorithm, by discarding the data that were "near" the decision boundary, also in a heuristic manner. This had the effect of "smoothing" the decision boundary. In the pre-processing stage of Wilson's edited nearest neighbor rule, all the data are first classified using the k-NN rule in a leave-one-out setting, and all the data misclassified are then discarded. In the decision stage new data are classified using the 1-NN rule with the resulting (smaller) edited set. As would be expected, although this algorithm does not remove much of the training data, its redeeming feature is that for a large class of problems the asymptotic performance of the edited nearest neighbor rule is difficult to differentiate from that of the optimal Bayes decision rule. In effect, Wilson's editing scheme makes the 1-NN rule perform like the k-NN rule with the optimal value of k. This edited nearest neighbor decision rule has come to be known in the literature as *Wilson editing*, and has left a lasting impact on subsequent instance-based learning algorithms. Indeed, almost all algorithms proposed since 1972 use Wilson editing as one of their initial steps, with the primary goal of lowering the final overall probability of misclassification.

One of the best algorithms in the literature for considerably reducing the size of the training set without degrading its classification performance is the iterative case-filtering algorithm (ICF) of Brighton and Mellish [1]. This algorithm consists of a two-step procedure for discarding data. The first step is Wilson editing, in which the k-NN rule is used to classify the data, in this case with $k = 3$. The second step is an original condensing step that makes use of novel notions of *reachable* sets and *coverage* sets. New query points are then classified using the 1-NN rule with the resulting condensed set.

An omnipresent concern with the *k*-NN rule has been the selection of the value of *k* that will yield the lowest probability of misclassification for the problem at hand. Furthermore, in the traditional *k*-NN rule, once the value of *k* has been chosen it remains fixed for all subsequent classification decisions. Therefore, in this setting the *k*-NN rule does not adapt its value of *k* to either the local density or the spatial distribution of the neighbors of a query point. Proximity graphs provide an approach that not only takes care of these issues, but does so in a fully automatic way without having to tune any parameters. Instead of taking a majority vote from among the *k* nearest neighbors of a query point *X*, the proximity graph decision rule takes a majority vote from among *all* the *graph* neighbors of *X* determined by a suitable proximity graph of the training data. As a concrete example, consider the Delaunay triangulation of the set of points in Figure 2 (left). It is evident that the number of Delaunay neighbors of a point varies greatly depending on the point's location. The leftmost point for example has only three neighbors, whereas the point in the center of the circularly arranged group of points has thirteen neighbors. Similarly, the point in the upper right has eleven Delaunay neighbors, and furthermore these eleven do not even correspond to the eleven nearest neighbors. Proximity graph methods may then replace the *k*-NN rule in Wilson editing, thus dispensing with the problem of determining a suitable value of *k*.

Proximity graph methods may also be used for discarding points that are far from the decision boundary, in condensing algorithms. The first condensing algorithm that used proximity graphs employed the Delaunay triangulation [23]. In this algorithm a data point *X* is first marked if all its Delaunay neighbors belong to the same class as that of *X*. Then all marked points are deleted. The strength of this condensing method lies in the fact that the 1-NN decision boundary of the original full training set is not changed after the data are deleted, a property referred to as *decision-boundary consistency*. All the discarded data are completely redundant and hence do not play a role in determining the decision boundary of the original larger training set. Initial experiments with Delaunay graph condensing, performed with cervical cancer data collected at McGill University that consisted of two thousand cells in four dimensions [18], revealed that the amount of data discarded was not particularly impressive [24]. An effect of the curse-of-dimensionality is that even in a space of only four dimensions, two thousand points are sufficiently scattered to allow points to have many Delaunay neighbors. Clearly, a point with a greater number of neighbors has a higher probability that one of its neighbors belongs to a different class, especially if it lies near the decision boundary, and thus a smaller probability that it will be discarded. This observation prompted the application of proximity graphs that have fewer edges than the Delaunay triangulation. Furthermore, in order to minimize the distortion of the decision boundary of the entire set it was advocated that the proximity graphs employed should be sub-graphs of the Delaunay triangulation, such as Gabriel graphs and relative neighborhood graphs [24]. It was found experimentally that Gabriel graphs discarded a significantly greater number of data points without degrading the performance, whereas relative neighbor graphs (RNGs) significantly increased the probability of misclassification. For this reason minimum spanning trees (which are sparser than RNGs) were not even tried in those experiments.

The application of Gabriel and Relative Neighborhood graphs to editing as well as condensing methods was experimentally investigated using two data sets (Image data and Vowel data) by Sánchez, Pla, and Ferri [22]. For the Image data the RNG gave the best recognition accuracy if only editing was used, but the Gabriel graph did best when editing was followed by condensing. On the other hand, with the Vowel data the simple 1-NN rule gave the highest accuracy. The data sets used in these experiments were composed of low-dimensional feature vectors: 5 for the Vowel data, and 2 for the Image data. It is thus difficult, from this study, to make conclusions about other data sets in general, and high-dimensional data in particular.

The promising results obtained independently by two very different approaches, namely, the iterative case-filtering (ICF) algorithm of Brighton and Mellish [1] and the Gabriel graph methods [22, 24], provided sufficient motivation to concatenate these approaches into a hybrid algorithm called GSASH that honed the best of both worlds [2-3]. GSASH employed Wilson-type editing but using a Gabriel decision rule, followed by Gabriel neighbor condensing and iterative-case filtering. The final decisions in queries were also made using the Gabriel decision rule.

A word is in order concerning the acronym GSASH appended to the title of this hybrid algorithm. No practically efficient algorithm exists for computing the Gabriel graph of very large training data sets. The fastest algorithm available is essentially the brute-force method that runs in $O(dn^3)$ time, where d is the dimension, although at least one algorithm has been proposed to speed up the average computation time by a constant factor [24]. In order to obtain a truly efficient algorithm one must resort to computing approximate Gabriel graphs with a suitable data structure, in the spirit of SASH, an approximate nearest neighbor method originally proposed by Houle [13]. GSASH is a modification of SASH to handle Gabriel rather than nearest neighbors. It allows the data structure containing the training data to be computed in $O(dn \log n)$ time using $O(dn)$ memory, so that the k approximate Gabriel neighbors of a query point may be computed in $O(dk \log n)$ time.

Experiments with the 25 data sets available at the time in the UCI Repository of Machine Learning Database [16] demonstrated that the Gabriel condensed set, using the approximate Gabriel graph, preserved quite faithfully the original decision boundary based on the exact Gabriel graph [2]. It was also experimentally observed that the ICF algorithm was overly zealous in discarding data, resulting in a slight decrease in recognition accuracy. The Hybrid GSASH algorithm on the other hand tends to incorporate the best of both worlds: preserve the decision boundary, thus maintaining the recognition accuracy, and reduce significantly the storage space required. Nevertheless, the sizes and dimensionalities of the 25 data sets used in the experiments were not particularly large compared to those of the data sets that have been added more recently to the UCI repository. Only two of the data sets used in [2] had more than 4000 patterns; the maximum was 5000, the minimum 150, and the average 944. In addition, only 6 data sets had dimensions greater than 19, the maximum dimension was 69, the minimum 3, and the average 16. Only two data sets had a number of classes greater than 9. Therefore one goal of the present study was to investigate how well proximity graph methods perform with larger data sets in higher dimensions, and how their performance scales with respect to the sparseness of the proximity graphs uti-

lized. A second goal was to compare proximity-graph methods with the traditional *k*-NN rules and support vector machine (SVM) algorithms, a class of completely different methods for designing classifiers, based on optimization techniques, that also has a history of superlative performance [9, 28]. Support vector machines have received widespread attention in the machine learning literature [5, 25]. Zhang and King [28] compared the performance of support vector machines with methods that employ the Gabriel graph. Indeed, it is conjectured that the Gabriel thinned set contains the support vectors [28].

4 Experiments and Results

The primary goal of this research project was to test the efficacy of the proximity graphs and their scalability with respect to their edge-density, rather than the running time of the algorithms used. For this reason, the *k*-NN, Gabriel, RNG, and MST decision rules were implemented using exact rather than approximate methods such as GSASH. Another drawback of using GSASH is that it puts a constraint on the maximum number of approximate neighbors it can return (to achieve computational efficiency). In contrast, our goal was to investigate the true error rates that the proximity graphs can deliver, as well as the number of neighbors required. For this, an exact approach was needed. In the following, the terms "point" and "instance" are used interchangeably.

For the *k*-NN, Gabriel, RNG, and MST decision rules, the voting neighbors of a query point were first calculated. Then the query point was classified by taking an unweighted majority vote of the voting neighbors' class memberships. Ties were broken arbitrarily by selecting the class with the smallest index value. The distances between pairs of instances were computed using the Hybrid Value Difference Metric (HVDM) described by Wilson and Martinez [27]. This metric allows the handling of both real-valued and categorical attributes, and provides approximate normalization between the different attributes.

The algorithm for the *k*-NN rule simply found the *k* nearest neighbors of the query point, by computing the distance to every point in the data set, and keeping the *k* smallest distances seen so far. No fancy data structure for computing neighbors more efficiently was used. In our implementation the time complexity of finding the *k* nearest neighbors of each point was $O((k+d)n)$, where n is the number of points, and d is the dimensionality. The algorithm for the Gabriel neighbor rule found the Gabriel neighbors of a query point q by trying each point p in the data set as a potential neighbor, and checking that no other point o lies in the sphere with diameter determined by q and p. A speedup for avoiding unnecessary checks [24] was used, but the worst-case time complexity for each query point was still $O(dn^2)$, where n is the number of points in the data set, and d is the dimensionality. The algorithm for the RNG neighbor rule operated exactly like the Gabriel algorithm, except that the distance check was different (lunes were used instead of diametral spheres). The algorithm for the MST neighbor rule computed the minimum spanning tree of the set of training points, into which the query point was first inserted. It then returned the neighbors of

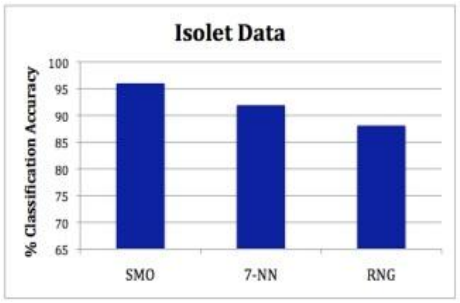
the query point in the resulting MST. Source code used for computing geometric minimum spanning trees in arbitrary dimensions was provided by Giri Narasimhan [17]. In this code, the (un-normalized) Euclidean metric was used as the distance between instances, instead of HVDM.

The algorithms for the *k*-NN, Gabriel, and RNG neighbor rules were implemented in C++. The imported geometric MST code was also written in C++. For the support-vector machine classifier, the Sequential Minimal Optimization (SMO) [19] algorithm from the off-the-shelf Weka data mining package [29] was used. The algorithms were tested on a 64-bit Linux machine with a 2.26GHz CPU and 3GB of RAM. No editing or condensing was done in the experiments presented below, unless stated otherwise. The experiments used four data sets, including three from the UCI Machine Learning Repository [10] (Isolet, Dermatology, and Segmentation).

Isolet Data.

The Isolet speech data set has 26 classes (the spoken name of each letter of the alphabet), 6238 training instances, and 1559 testing instances, each with 617 real-valued features consisting of spectral coefficients, contour features, sonorant features, pre-sonorant features, and post-sonorant features. The suggested split between training and testing subsets as indicated by the data set was used, rather than performing cross-validation. This data set is truly massive in terms of both size and dimensionality.

The accuracy (rate of correct classification) of the *k*-NN rule as a function of *k* for the Isolet data is shown in Figure 3. The maximum accuracy was 91.9% for *k* = 7. The RNG accuracy was slightly lower at 88.1%. The SMO classifier (with parameters: complexity = 1.5, polynomial kernel exponent = 1.5) reached an accuracy of 96.0%, surpassing the other instance-based methods. These results are summarized in the chart on the right.



No results were obtained with the Gabriel and MST classifiers because the algorithms were too slow to run to completion. In the MST classifier, every test instance required re-computing the MST. In the Gabriel classifier, there were simply too many neighbors. This underscores the fact that despite their similar theoretical worst-case time complexity, the actual running time required by the RNG and Gabriel algorithms can be vastly different. In this data set, each point has very many Gabriel neighbors, but only a few RNG neighbors. Verifying that two points are Gabriel (or RNG) neighbors requires checking that no other point lies in their diametral sphere (or lune). This test uses O(*dn*) time, an expensive operation in this context. For the RNG, most pairs of points are not neighbors. Thus most neighbors are discarded by promptly finding a counterexample. On the other hand, for the Gabriel graph most pairs of nodes are neighbors. Thus, the linear-time check will be required for almost every pair of nodes.

From the theoretical point of view Devroye [6] has shown that for most probability density functions governing the data, the asymptotic expected number of edges in the Gabriel graph is essentially $2^{d-1}n$, where n is the number of instances in the training set, and thus the number of Gabriel neighbors of a point grows exponentially in terms of the dimension. To obtain some practical intuition for this curse of dimensionality, the Gabriel graph was computed for only 35 test instances, and the average number of neighbors already reached 4588. This result provided empirical evidence to support the hypothesis that the Gabriel graph, like its denser parent Delaunay triangulation, quickly becomes extremely dense in high-dimensional spaces. Furthermore, once the number of neighbors reaches a significant fraction of the entire data set, the resulting decisions are primarily based on the prior probability of the classes, effectively ignoring much of the feature information, and resulting in a natural degradation of performance.

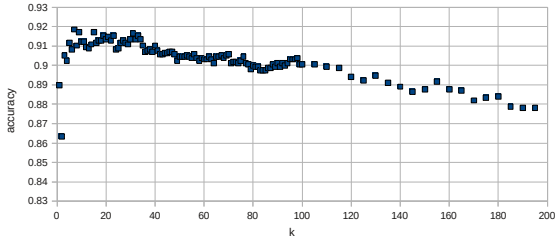


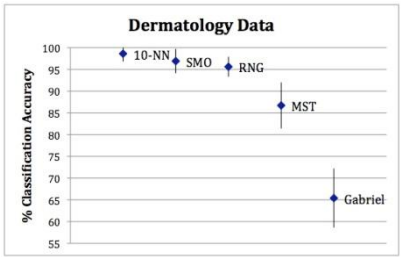
Fig. 3. The classification accuracy of the *k*-NN rule with the Isolet data.

Dermatology Data.

The Dermatology data set has six classes (skin diseases) and 366 instances, each with 33 categorical attributes and one real-valued attribute. The total dimensionality of the feature space is therefore 34. Eight instances contained missing data, and were removed from the data set. In these experiments, randomly selected subsets consisting of 20% of the data were used for the testing sets, leaving the remaining 80% for training. All the accuracy results were averaged over 10 trials.

The classification accuracy of the *k*-NN rule as a function of *k* for the Dermatology data is shown in Figure 4. The *k*-NN rule does very well, achieving a maximum value of 98.6% for both $k = 10$ and $k = 15$, although it does pretty well for all the values of *k* tried. The following table and graph summarize the results, where the error bars indicate ± one standard deviation.

Classifier	Mean Accuracy (%)	Standard Deviation
10-NN	98.6	1.8
SMO	96.9	2.8
RNG	95.6	2.3
MST	86.7	5.3
Gabriel	65.4	6.8



The SMO algorithm performs slightly worse than 10-NN, obtaining a top mean accuracy of 96.9% (for parameters: complexity = 2.0, polynomial kernel exponent = 3.0). However, as the error bars show, the results with 10-NN, SMO, and RNG are not significantly different from each other. On the other hand, these three classifiers yield results significantly better than those with the MST, which in turn are significantly better than those obtained with the Gabriel graph.

Interestingly, this is the only data set for which an instance-based method appears to do better than the SMO (although not significantly). The *k*-NN rule is significantly better than all three proximity graph methods. The MST appears to be too sparse to capture the requisite discrimination information.

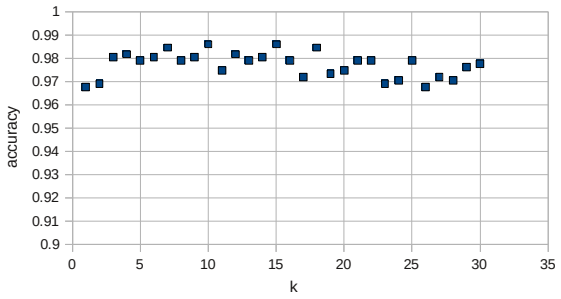


Fig. 4. The classification accuracy for the *k*-NN rule with the Dermatology data.

Since the number of Gabriel graph neighbors grows so fast with *d*, it seemed conceivable that capping might still capture local information better than the *k*-NN rule. Accordingly, an experiment was performed that looked at only the *k* closest Gabriel neighbors. The mean classification accuracy for this decision rule with the Dermatology data is shown in Figure 5. For all values of *k* up to 30, the mean accuracy is better than all three proximity graph methods, and for $k = 22$ the value of 98.6% matches the accuracy of *k*-NN for $k = 10$ and $k = 15$. These results however do not appear to provide any additional insight into the workings of the Gabriel graph, since the values of *k* that yield results comparable to the *k*-NN rule are greater, and therefore the *k* closest Gabriel neighbors may in fact include the *k* nearest neighbors.

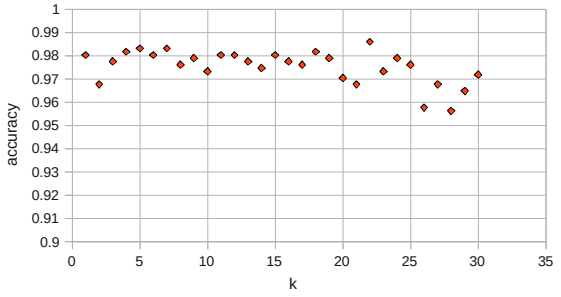


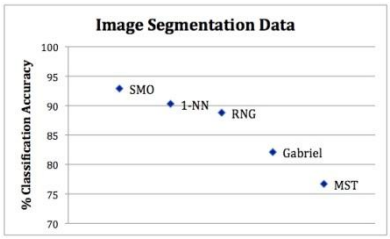
Fig. 5. The mean classification accuracy for the *closest k Gabriel neighbors* rule with the Dermatology data.

Image Segmentation Data.

The Image Segmentation data set has 7 classes, 210 train instances, and 2100 test instances, each with 18 real-valued attributes. We used the suggested train/test split given by the data set, rather than performing cross-validation (see Figure 6).

The 1-NN rule does best, and the SMO algorithm (with complexity = 3.0, polynomial kernel exponent = 5.0) surpasses all methods (see table and chart below). Again, the RNG is superior to the Gabriel graph, and the MST does not perform as well.

Classifier	Classification Accuracy (%)
SMO	92.9
1-NN	90.3
RNG	88.8
Gabriel	82.1
MST	76.7



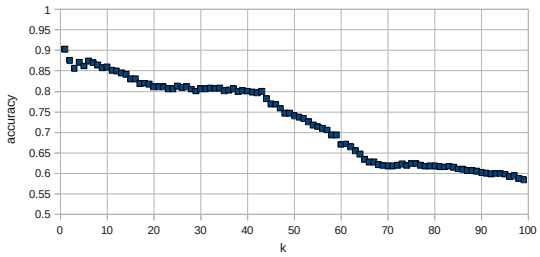


Fig. 6. The classification accuracy for the *k*-NN rule with the Image Segmentation data.

Figure 7 shows the results, and the amount of storage used (as a fraction of the initial training set) for different k used in Wilson editing. Storage was reduced by 15%, at the expense of accuracy. A top accuracy of 85.4% was achieved with $k = 6$.

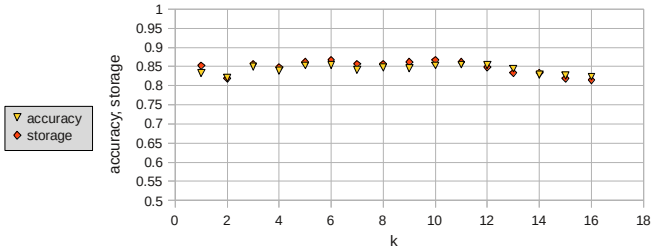


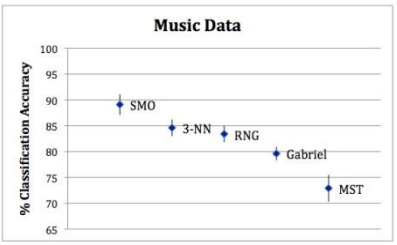
Fig. 7. The classification accuracy and storage used when applying Wilson editing to the Image Segmentation data.

Music Data.

The Music data set has 2 classes (Western and Non-Western music) and 1641 instances, each with 41 real-valued attributes, of which 1 was constant [11]. The total dimensionality of the feature space is therefore 40. In the experiments, 20% of the data was randomly set aside as the testing set, and the remaining 80% was used for training. All accuracy results were averaged over 10 trials.

The classification accuracy of the *k*-NN rule as a function of *k* for the Music data is shown in Figure 8. The *k*-NN rule achieves a top mean accuracy of 84.6% for $k = 3$. The SMO support-vector machine classifier achieves an accuracy of 86.5% according to Gomez and Herrera [11], but they reported no standard deviations. Replicating their experiment using the same parameters (complexity = 1.5, polynomial kernel exponent = 1.5), a fairly significantly higher mean accuracy of 89.1% was obtained. The results are summarized in the following table and graph:

Classifier	Mean Accuracy (%)	Standard Deviation
SMO	89.1	2.0
3-NN	84.6	1.6
RNG	83.4	1.6
Gabriel	79.6	1.3
MST	72.9	2.6



The RNG results are significantly better than those of the Gabriel graph, which are in turn significantly better than the MST results. It is worth noting that for the Gabriel rule, the test instances had an average number of 303 voting neighbors, further illustrating that the Gabriel graph becomes dense in high dimensions. With the RNG classifier, the average number of neighbors was only 4.2. The 3-NN rule is superior to all three proximity graph methods, although not statistically significantly better than the RNG. The SMO classifier yields results statistically significantly better than all the other classifiers, with a mean accuracy of 89.1%.

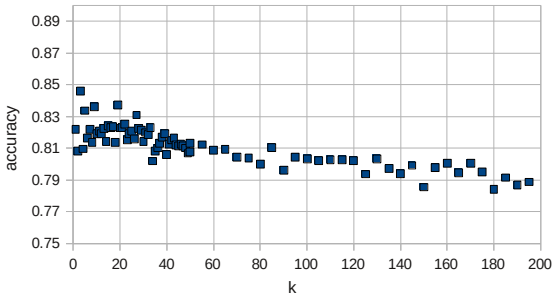


Fig. 8. The classification accuracy for the *k*-NN rule with the Music data.

5 Concluding Remarks

In theory it is known that for almost all probability density functions governing the training data, the expected number of edges in the Gabriel graph grows exponentially with the dimensionality of the space in which the data are embedded [6]. The results of the experiments obtained here provide empirical confirmation of the theory.

In previous research with data sets having small-dimensionality *d*, it has been frequently found that the Gabriel graph performed best among the proximity graph methods. It was hypothesized that with higher *d* a sparser graph would give better results. The empirical results obtained here confirm this hypothesis. In all the experiments the relative neighbor graph (RNG) performed significantly better than the Gabriel graph, probably as a result of the explosive growth of the number of Gabriel graph neighbors as *d* increases. To determine the degree to which the sparseness of proximity graphs can help, experiments were performed with the (connected) proximity graph that has the fewest possible number of edges, the minimum spanning tree (MST). The experimental results showed conclusively that the MST is too sparse to capture sufficient discrimination information to achieve good performance, and thus it appears to be useless for this application in instance-based learning, thus settling a long-standing speculation.

The traditional *k*-NN decision rule has a long recorded history of yielding high classification accuracy in practice. Its drawbacks of high memory and computation requirements motivated the introduction of proximity graph methods for reducing the size of the training data. For low values of *d* it appeared that little would be lost in terms of performance, by resorting to proximity graphs. However, the results obtained here with large *d* clearly indicate that *k*-NN methods are superior, thus challenging proximity graph methods, if performance is the only issue. In all the experiments the *k*-NN rule performed statistically significantly better than the Gabriel graph, and although it also typically achieved better results than the RNG these were not statistically significant.

It is known that in theory the *k*-NN rules are asymptotically Bayes optimal, and that therefore there should not exist any other classifier that gives strictly better average classification accuracy [7-8]. Nevertheless, as the experiments reported here demonstrate, even for large real-world data sets, the SMO support vector machine yields statistically significantly better results than *k*-NN. This suggests that in practice SVMs should be the classifiers of choice, other things being equal. The drawback of traditional implementations of SVMs is high computation in the design stage of the classifier, although the sequential minimal optimization (SMO) version offers improvements in this regard. Their advantage is fast classification of new query data. Proximity graph decision rules on the other hand are very slow for this task unless approximate methods such as GSASH are used. Therefore the results of this study suggest that the most fruitful approach for classification is to use SMO in order to obtain the best classification performance, if the computation time spent on classifier design can be significantly reduced. Furthermore, since this computation time depends to a large extent on the size of the training data, it is worthwhile reducing the number of instances in a computationally inexpensive manner before subjecting them

to an SMO support vector machine training algorithm. This appears to be a more appropriate role for proximity graphs to play than to act as neighbor filters in decision rules.

References

1. Brighton, H., Mellish, C.S.: Advances in Instance Selection for Instance Based Learning Algorithms. Data Mining and Knowledge Discovery. 6, 153-172 (2002)
2. Bhattacharya, B., Mukherjee, K., Toussaint, G.T.: Geometric Decision Rules for Instance-Based Learning Algorithms. Proc. Pattern Recognition and Machine Intelligence: First International Conference, S. K. Pal et al., (Eds.): LNCS 3776, Kolkata, India, December 20-22, pp. 60-69 (2005)
3. Bhattacharya, B., Mukherjee, K., Toussaint, G.T.: Geometric Decision Rules for High Dimensions. Proc. 55th Session of the International Statistics Institute, Sydney, Australia, April 5-12, (2005)
4. Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory, 13, 21–27 (1967)
5. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning, 20, September, 1-25 (1995)
6. Devroye, L.: The Exptected Size of Some Graphs in Computational Geometry. Computers and Mathematics with Applications. 15, 53-64 (1988)
7. Devroye, L.: On the Inequality of Cover and Hart in Nearest Neighbor Discrimination. IEEE Transactions on Pattern Analysis and Machine Intelligence, 3, 75-78 (1981)
8. Devroye, L., Györfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition, Springer Verlag, (1996)
9. Duan, K.-B., Keerthi, S.S.: Which Is the Best Multiclass SVM Method? An Empirical Study. N.C. Oza et al., (Eds.), Sixth International Workshop on Multiple Classifier Systems (MCS-2005), LNCS 3541, pp. 278-285, Springer-Verlag, Berlin-Heidelberg (2005)
10. Frank, A., Asuncion, A.: UCI Machine Learning Repository. [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science (2010)
11. Gomez, E., Herrera, P.: Comparative Analysis of Music Recordings from Western and Non-Western traditions by Automatic Tonal Feature Extraction. Empirical Musicology Review, 3, (2008)
12. Hart, P.E.: The Condensed Nearest Neighbor Rule. IEEE Transactions on Information Theory, 14, 515-516 (1968)
13. Houle, M.: SASH: A Spatial Approximation Sample Hierarchy for Similarity Search. Tech. Report RT-0517, IBM Tokyo Research Lab. (2003)
14. Jaromczyk, J.W., Toussaint, G.T.: Relative Neighborhood Graphs and their Relatives. Proceedings of the IEEE, 80, 1502–1517 (1992)
15. Kirkpatrick, D.G., Radke, J.D.: A Framework for Computational Morphology.

In: G. T. Toussaint, Ed., Computational Geometry, pp. 217-248, North Holland, Amsterdam, Netherlands (1985)
16. Merz, C.J., Murphy, P.M.: UCI Repository of Machine Learning Database, Internet http://www.ics.uci.edu/mlearn/MLRepository.html, Department of Information and Computer Science, University of California.
17. Narasimhan, G., Zhu, J., Zachariasen, M.: Experiments with Computing Geometric Minimum Spanning Trees. Proceedings of Algorithm Engineering and Experiments, (ALENEX'00), pp. 183-196, Springer Lecture Notes in Computer Science, (2000)
18. Oliver, L.H., Poulsen, R.S., Toussaint, G.T.: Estimating False Positive and False Negative Error Rates in Cervical Cell Classification. J. Histochemistry and Cytochemistry, 25, 696-701 (1977)
19. Platt, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: Advances in Kernel Methods - Support Vector Learning, B. Schoelkopf, et al. (Eds.), MIT Press (1988)
20. Toussaint, G.T.: Geometric Proximity Graphs for Improving Nearest Neighbor Methods in Instance-Based Learning and Data Mining. International J. Computational Geometry and Applications, 15, 101-150 (2005)
21. Toussaint, G.T.: The Relative Neighborhood Graph of a Finite Planar Set. Pattern Recognition, 12, 261-268 (1980)
22. Sánchez, J.S., Pla, F., and Ferri, F.J.: Prototype Selection for the Nearest Neighbor Rule through Proximity Graphs. Pattern Recognition Letters, 18, 507–513 (1997)
23. Toussaint, G.T., Poulsen, R.S.: Some New Algorithms and Software Implementation Methods for Pattern Recognition Research. Proc. Third International Computer Software and Applications Conference, pp. 55-63, IEEE Computer Society (1979)
24. Toussaint, G.T., Bhattacharya, B.K., Poulsen, R.S.: The Application of Voronoi Diagrams to Nonparametric Decision Rules. Proc. Computer Science and Statistics: 16th Symposium on the Interface, pp. 97-108, North-Holland, Amsterdam (1985)
25. Vapnik, V.N.: The Nature of Statistical Learning Theory, Springer-Verlag, Heidelberg (1995)
26. Wilson, D.L.: Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE Transactions on Systems, Man, and Cybernetics, 2, 408-421 (1973)
27. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine Learning, 38, 257–286 (2000)
28. Zhang, W., King, I.: A Study of the Relationship Between Support Vector Machine and Gabriel Graph. Proc. IEEE International Joint Conference on Neural Networks, IJCNN'02, Honolulu, 1, 239-244 (2002)
29. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA Data Mining Software: An Update. SIGKDD Explorations, 11, (2009)