Geometric Decision Rules for Instance-Based Learning Problems

Binay Bhattacharya[1,⋆], Kaustav Mukherjee[1], and Godfried Toussaint[2,⋆⋆]

[1] School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada
[2] School of Computer Science, McGill University, Montréal, Québec, Canada

Abstract. In the typical nonparametric approach to classification in instance-based learning and data mining, random data (the training set of patterns) are collected and used to design a decision rule (classifier). One of the most well known such rules is the k-nearest neighbor decision rule (also known as *lazy learning*) in which an unknown pattern is classified into the majority class among the k-nearest neighbors in the training set. This rule gives low error rates when the training set is large. However, in practice it is desired to store as little of the training data as possible, without sacrificing the performance. It is well known that thinning (condensing) the training set with the Gabriel proximity graph is a viable partial solution to the problem. However, this brings up the problem of efficiently computing the Gabriel graph of large training data sets in high dimensional spaces. In this paper we report on a new approach to the instance-based learning problem. The new approach combines five tools: first, editing the data using Wilson-Gabriel-editing to smooth the decision boundary, second, applying Gabriel-thinning to the edited set, third, filtering this output with the ICF algorithm of Brighton and Mellish, fourth, using the Gabriel-neighbor decision rule to classify new incoming queries, and fifth, using a new data structure that allows the efficient computation of *approximate* Gabriel graphs in high dimensional spaces. Extensive experiments suggest that our approach is the best on the market.

1 Introduction

In the typical non-parametric classification problem (see Devroye, Gyorfy and Lugosi [4]) we have available a set of d measurements or observations (also called a feature vector) taken from each member of a data set of n objects (patterns) denoted by $\{X, Y\} = \{(X_1, Y_1), ..., (X_n, Y_n)\}$, where X_i and Y_i denote, respectively, the feature vector on the ith object and the class label of that object. One of the most attractive decision procedures, conceived by Fix and Hodges in 1951, is the nearest-neighbor rule (1-*NN*-rule). Let Z be a new pattern (feature vector) to be

⋆ Research supported by NSERC and MITACS.
⋆⋆ Research supported by NSERC and FCAR.

S.K. Pal et al. (Eds.): PReMI 2005, LNCS 3776, pp. 60–69, 2005.

classified and let X_j be the feature vector in $\{X, Y\} = \{(X_1, Y_1), ..., (X_n, Y_n)\}$ closest to Z. The nearest neighbor decision rule classifies the unknown pattern Z into class Y_j. The resulting feature space is partitioned into convex polyhedra. This partitioning is called the *Voronoi* diagram. Each pattern (X_i, Y_i) in (X, Y) is surrounded by its Voronoi polyhedron consisting of those points in the feature space closer to (X_i, Y_i) than to (X_j, Y_j) for all $j \neq i$. The *1-NN*-rule classifies a new pattern Z that falls into the Voronoi polyhedron of pattern X_j into class Y_j. Therefore, the decision boundary of the *1-NN*-rule is determined by those portions of the Voronoi diagram that separate patterns belonging to different classes.

A key feature of this decision rule (also called *lazy learning, instance-based learning, and memory-based learning*) is that it performs remarkably well considering that no explicit knowledge of the underlying distributions of the data is used. Furthermore, a simple generalization of this rule called the *k-NN*-rule, in which a new pattern Z is classified into the class with the most members present among the k nearest neighbors of Z in $\{X, Y\}$, can be used to obtain good estimates of the Bayes error and its probability of error asymptotically approaches the Bayes error (Devroye et al. [4]).

In practice the size of the training set $\{X, Y\}$ is not infinite. This raises two fundamental questions of both practical and theoretical interest. How fast does the k-nearest neighbor error rate approach the Bayes error rate as n approaches infinity, and what is the finite-sample performance of the *k-NN*-rule ([9,6]) These questions have in turn generated a variety of additional problems concerning several aspects of *k-NN*-rules in practice. How can the storage of the training set be reduced without degrading the performance of the decision rule? How large should k be? How can the rule be made robust to overlapping classes or noise present in the training data? How can new decisions be made in a practical and computationally efficient manner? Geometric proximity graphs such as Voronoi diagrams and their many relatives provide elegant approaches to these problems.

2 Editing the Training Data to Improve Performance

Methods that have as their goal the improvement of recognition accuracy and generalization, rather than the reduction of the size of the stored training set, are called *editing* rules in the pattern recognition literature. In 1972 Wilson [12] first conceived the idea of editing with this goal in mind, and proposed an elegant and simple algorithm. Delete all points (in parallel) misclassified by the *k-NN*-rule. Classify a new unknown pattern Z using the 1-*NN* rule with the *edited* subset of $\{X, Y\}$.

This simple editing scheme is so powerful that the error rate of the 1-*NN* rule that uses the edited subset converges to the Bayes error as n approaches infinity. One problem with with Wilson-editing is that, although the final decision is made using the *1-NN*-rule, the editing is done with the *k-NN*-rule, and thus one is faced again with the problem of how to choose the value of k in practice. In our approach we will modify Wilson-editing as described in the following.

3 Thinning the Training Data to Reduce Storage

3.1 Decision-Boundary-Consistent Subsets Via Proximity Graphs

In 1979 Toussaint and Poulsen [11] used d-dimensional Voronoi diagrams to delete "redundant" members of $\{X, Y\}$ in order to obtain a subset of $\{X, Y\}$ that implements *exactly* the same decision boundary as would be obtained using all of $\{X, Y\}$. For this reason they called their method *Voronoi condensing*. The algorithm in [11] is very simple. Two points in $\{X, Y\}$ are called *Voronoi neighbors* if their corresponding Voronoi polyhedra share a face. First mark each point X_i if all its Voronoi neighbors belong to the same class as X_i. Then discard all marked points. The remaining points form the Voronoi condensed subset. Voronoi condensing does not change the error rate of the resulting decision rule because the nearest neighbor decision boundary with the reduced set is identical to that obtained by using the entire set. For this reason the Voronoi condensed subset is called *decision-boundary consistent*. While this approach to editing sometimes does not discard a large fraction of the training data, that information in itself is extremely important to the pattern classifier designer because the fraction of the data discarded is a measure of the resulting reliability of the decision rule. If few points are discarded it means that the feature space is relatively empty because few points are completely "surrounded" by points of the same class. This means that either there are too many features (the dimensionality of the space is too high) or more training data are urgently needed to be able to obtain reliable and robust estimates of the future performance of the rule. The main problem with Voronoi-condensing is that the complexity of computing all the Voronoi neighbors of a point is prohibitive in high dimensions.

3.2 Condensing Prototypes Via Proximity Graphs

Bhattacharya [2] and Toussaint, Bhattacharya and Poulsen [10] generalized Voronoi condensing so that it would discard more points in a judicious and organized manner so as not to degrade performance unnecessarily. To better understand the rationale behind their proximity-graph-based methods it is useful to cast the Voronoi condensing algorithm in its dual form. The dual of the Voronoi diagram is the Delaunay triangulation. Therefore an equivalent description of Voronoi-condensing is to discard all points (in parallel) if all their Delaunay neighbors belong to the same class. The idea is then to use subgraphs of the Delaunay triangulation in exactly the same manner. Experimental results obtained in [2] and [10] suggest that the Gabriel graph is the best in this respect. This procedure will be referred to as Gabriel-thinning in the following. Two points p and q are Gabriel neighbors if the "region of influence" of p and q is empty, i.e. there does not exist any point r of the set such that $d^2(p, q) > d^2(p, r) + d^2(r, q)$ where $d(p, q)$ denotes the distance measure between p and q. In the Euclidean space the region of influence of p and q is the smallest hypersphere that contains them. The Gabriel graph is obtained by connecting two points with an edge if they are Gabriel neighbors.

In 1998 Bhattacharya and Kaller [1] proposed a proximity graph they call the *k-Gabriel* graph and show how inexact thinning can be performed with this graph. The *k-Gabriel* graph is much easier to use than the *k-Delaunay* graph and yields good results.

4 Filtering the Training Data for Fine Tuning

Brighton and Mellish [3] proposed a new hybrid method and compared it to several other hybrid methods on 30 different classification data sets. Their elegant and simple algorithm, which appears to be the previous best in practice, is called *iterative case filtering* (ICF), and may be described as follows. The first part of the algorithm consists of preprocessing with the original Wilson editing scheme. The second part of their algorithm, their main contribution, is an adaptive condensing procedure. The rule for discarding an element (X_k, Y_k) of $\{X, Y\}$ depends on the relative magnitude of two functions of (X_k, Y_k) called the *reachable* set of (X_k, Y_k) and the *coverage* set of (X_k, Y_k). The *reachable* set of (X_k, Y_k) consists of all the data points contained in a hypersphere centered at X_k with radius equal to the distance from X_k to the nearest data point belonging to a class different from that of X_k. More precisely, let $S(X_k, Y_k)$ denote the hypersphere with center X_k and radius $r_k = min\{d(X_k, X_j)|Y_j \neq Y_k\}$ minimized over all j. Then all the data points of $\{X, Y\}$ that are contained in $S(X_k, Y_k)$ constitute the reachable set of (X_k, Y_k) denoted by $R(X_k, Y_k)$. The *coverage* set of (X_k, Y_k), denoted by $C(X_k, Y_k)$, consists of all the data points in $\{X, Y\}$ that have (X_k, Y_k) in their own reachable set. More precisely, $C(X_k, Y_k)$ consists of all data points $(X_i, Y_i), i = 1, 2, ..., n$ such that (X_k, Y_k) is a member of $R(X_i, Y_i)$. The condensing (thinning) step of the ICF algorithm of Brighton and Mellish [3] can now be made precise. First, for all i mark (X_i, Y_i) if $|R(X_i, Y_i)| > |C(X_i, Y_i)|$. Then discard all marked points. This condensing step is repeated until no marked points are discarded. We will refer to this second iterative step of their overall procedure as the *filtering* step of ICF.

5 The New Hybrid Gabriel Graph Algorithm

Our new approach to the problem of instance-based learning depends heavily on the use of the Gabriel graph. First we describe the approach using the exact Gabriel graph, and after that we turn to the practical version for high dimensional problems.

Step 1: The original training set $\{X, Y\}$ is subjected to editing with a modification of Wilson editing. Instead of editing with the k nearest neighbors of a point, we use the Gabriel neighbors, thus dispensing with the problem of choosing a value of k. Let $\{X, Y\}'$ denote the edited training set.

Step 2: The set $\{X, Y\}'$ is subjected to thinning (condensing) using the Gabriel graph rule: points are discarded (in parallel) if all their Gabriel neighbors belong to the same class. Let $\{X, Y\}''$ denote the resulting edited-thinned training set.

Step 3: Subject the set $\{X, Y\}''$ to the *filtering* step of the ICF algorithm of Brighton and Mellish [3]. Let $\{X, Y\}'''$ denote the final training set obtained.

Decision rule: A new query point Z is classified according to the majority vote among the Gabriel neighbors of Z in $\{X, Y\}'''$.

The above algorithm is called Hybrid Gabriel-Graph Algorithm. If $\{X, Y\}''$ is used instead of $\{X, Y\}'''$ in the decision rule, the algorithm is called Gabriel-Graph Algorithm. As discussed earlier, the set $\{X, Y\}''$ maintains the decision boundary of the the set $\{X, Y\}$ extremely well [2,10,11].

In high dimensional spaces computing the exact Gabriel graph may be costly for large training sets. The brute force approach to compute the Gabriel graph of n points in d dimension is $O(dn^3)$. There is no known faster algorithm for the exact computation of the Gabriel graph in arbitrary dimensions. In this paper a data structure called $GSASH$ is introduced that allows efficient computation of approximate Gabriel neighbors. The practical version of our algorithm uses the *approximate* Gabriel graph instead of the exact Gabriel graph at every step. The resulting algorithms are called Approximate Hybrid Gabriel-Graph Algorithm and Approximate Gabriel-Graph Algorithm.

5.1 GSASH Structure

The data structure GSASH [8] is a modification of SASH [5] to handle Gabriel neighbors rather than the originally intended k nearest neighbors. GSASH is basically a multi-level directed acyclic graph with the following characteristics:

(a) Each node in the graph corresponds to a data item.

(b) The graph consists of $O(\log n)$ levels for a dataset X of size n. Actually at most $2n$ nodes in GSASH are maintained, with n items at the bottom most level, say h, and one item at level 1. With the possible exception of the first level, each level i has half of the nodes as in level $i + 1$. The overall structure is as follows: All the n data items are stored at level h. We then "copy" half of these i.e. $\frac{n}{2}$ items uniformly at random to level $h - 1$. We repeat this process of "copying" all the way up to the root. If a level has at most c nodes (the constant c is chosen in advance of the construction), we pick one of these c nodes to be the root. The root is at level 1. The levels of GSASH are therefore numbered from 1 (the top level) to h (the bottom level). Let S_i denote the data items stored at level i.

(c) The nodes of S_i at level i have edges directed to the nodes of S_{i+1} at level $i + 1$. Each node (object p) links to at most c nodes at level below. These nodes represent the c closest approximate Gabriel neighbors of p among the points of the set S_{i+1}.

The issue of determining the value of c is an important one. One of the considerations behind the GSASH design is that connections to a given node v from a sufficient number of its approximate Gabriel neighbors could help guide a query search to v. In SASH a fixed choice of $c = 16$ led to good performance. In GSASH the same value for the parameter c has been used.

5.2 GSASH Querying

Given a query object q, the search for a set of closest k approximate Gabriel neighbors can be performed on GSASH. The query process starts at the root. Let $P_i(q)$ denote the set of k_i closest approximate Gabriel neighbors of q selected from among the elements of S_i. Let $C_i(q)$ denote the distinct child nodes of $P_i(q)$. $P_{i+1}(q)$ is then constructed by first determining the approximate Gabriel neighbors of q among the points of the set $C_i(q)$ with respect to the set S_{i+1} and then selecting the closest k_{i+1} of them to q.

We can easily use $k_i = k$ for all i. Like SASH [5] the GSASH experimentation uses a geometric search pattern that allows a larger proportion of the search to be devoted on the largest samples of the elements, namely those located closer to the bottom of the GSASH.

Theorem 1. *The GSASH data structure of n objects can be constructed in $O(dn \log n)$ time requiring $O(dn)$ extra storage space. An arbitrary k approximate Gabriel neighbors query can be answered in $O(dk \log n)$ time.*

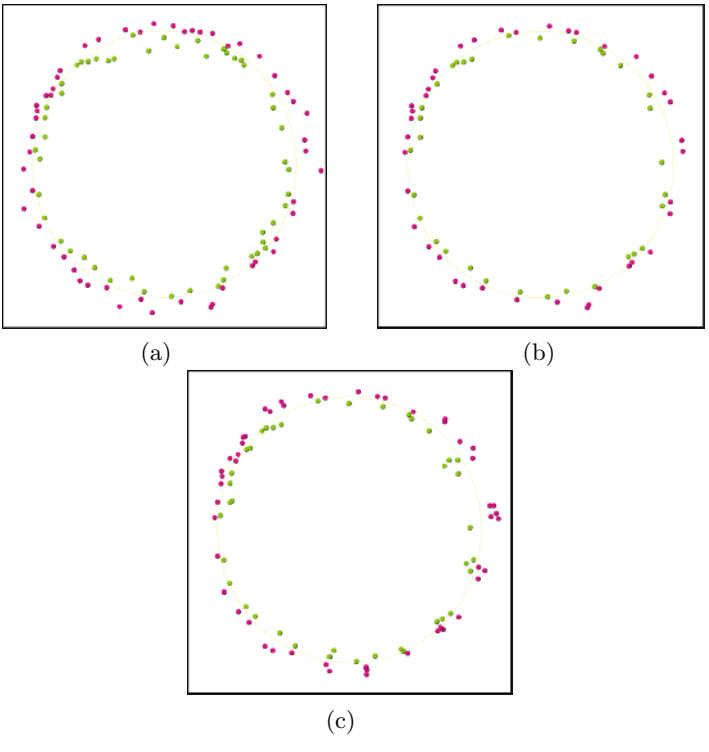

Fig. 1. (a) Gabriel thinned data set. 104 out of 1000 points remained. Error rate 2.75% on 200 randomly generated points (b)Hybrid thinned data set. 73 out of 1000 points remained. Error rate 3.25% on 200 randomly generated points (c)ICF thinned data set. 98 out of 1000 points remained. Error rate 4.75% on 200 randomly generated points.

6 Experimental Results

In this section we compare three algorithms: Approximate Gabriel-Graph Algorithm (called Gabriel), ICF-Algorithm (called ICF) and Approximate Hybrid Gabriel-Graph Algorithm (called Hybrid). In order to investigate the inner workings of these algorithms we generated some synthetic data sets in the plane. All data are generated inside a unit square. The decision boundary considered is circular. Points generated uniformly on either side of the boundary are given opposite class labels. Each generated set has 1000 points. The results obtained are 10-fold cross validated. Figs. 1(a), (b) and (c) show the points that are kept by the algorithms Gabriel, Hybrid and ICF respectively. The figures also show the error rates and storage reductions for the three algorithms. As expected, Gabriel produces the condensed subset of points that tightly maintains the decision boundary at the cost of slightly extra storage. The ICF distorts the boundary significantly. There are significant number of "non-border" points that are maintained which are not close enough to the boundary so as to classify the border points correctly. This is a significant weakness of the ICF algorithm. This

Table 1. Particulars of the data sets used

Data Set	Size	Number of Classes	Dimension
abalone	4177	29	8
anneal	898	6	38
audiology	226	24	69
auto-mpg	398	9	7
balance-scale	625	3	4
breast-cancer-w	699	2	9
bupa	345	2	6
car	1728	4	6
cell	2999	2	6
cmc	1473	3	10
crx	690	2	15
dermatology	366	6	33
ecoli	336	8	7
glass	214	7	10
haberman	306	2	3
heart-cleveland	303	5	13
hepatitis	155	2	19
house-votes-84	435	2	16
ionosphere	351	2	34
iris	150	3	3
pima-diabetes	768	2	8
thyroid	215	3	6
waveform	5000	3	21
wine	178	3	13
wisconson-bc-di	569	2	30

Table 2. Results on real-world data

Data Set	Gabriel			Hybrid			ICF-1			ICF-2		
	Accu	Sdev	Stor	Accu	Sdev	Stor	Accu	Sdev	Stor	Accu	Sdev	Stor
abalone	25.8%	1.1	28.7%	23.7%	1.7	14.8%	21.2%	1.8	24.3%	21.5%	2.3	17.8%
anneal	83.1%	3.2	23.2%	94.3%	2.2	10.3%	94.1%	3.3	20.7%	91.8%	2.0	13.7%
audiology	51.6%	4.3	23.2%	55.1%	8.4	15.0%	73.8%	6.2	31.3%	53.8%	5.3	14.8%
auto-mpg	47.3%	6.4	25.5%	53.2%	4.0	30.0%	50.9%	3.7	36.9%	48.9%	10.0	34.7%
balance-scale	85.1%	1.8	61.3%	80.0%	2.0	19.0%	78.1%	2.1	16.0%	79.7%	1.7	18.6%
breast-cancer-w	95.8%	2.9	10.4%	96.0%	2.1	1.8%	94.5%	1.9	3.0%	94.5%	3.4	2.5%
bupa	64.6%	5.5	46.6%	63.8%	6.4	22.9%	56.8%	4.4	22.4%	64.6%	5.5	26.2%
car	94.4%	1.4	81.5%	94.3%	1.2	13.7%	95.5%	1.3	17.4%	94.6%	1.2	13.7%
cell	94.1%	1.2	8.2%	94.6%	0.6	3.3%	94.2%	0.5	4.6%	93.8%	0.5	2.2%
cmc	51.2%	3.0	45.3%	52.4%	2.5	17.8%	47.8%	5.0	28.5%	51.8%	3.0	22.0%
crx	69.9%	2.7	27.5%	85.9%	3.5	8.2%	82.6%	2.5	8.5%	74.5%	4.6	16.8%
dermatology	97.3%	2.6	86.6%	94.5%	3.5	13.2%	94.0%	3.9	10.2%	95.6%	3.7	14.1%
ecoli	83.6%	4.7	34.0%	82.7%	4.8	11.8%	83.3%	6.9	14.2%	84.2%	4.3	13.8%
glass	64.3%	8.9	48.1%	66.2%	5.9	26.3%	63.8%	9.0	27.4%	68.6%	8.8	26.5%
haberman	74.8%	5.9	18.7%	70.5%	3.3	10.6%	68.9%	4.8	13.9%	69.2%	4.2	12.9%
heart-cleveland	54.0%	5.8	14.3%	50.0%	3.9	9.4%	54.7%	5.2	18.2%	54.7%	7.6	7.3%
hepatitis	76.1%	7.4	11.3%	74.8%	12.8	6.0%	80.0%	9.0	9.0%	80.0%	8.3	5.3%
house-votes-84	94.3%	3.5	48.1%	87.6%	4.8	8.0%	87.1%	6.9	10.9%	84.4%	7.6	7.8%
ionosphere	85.7%	4.3	54.7%	81.1%	10.1	8.5%	82.0%	6.2	5.3%	84.6%	2.6	8.8%
iris	95.3%	4.5	20.8%	92.0%	5.6	14.7%	92.0%	6.1	18.5%	90.0%	2.4	16.7%
pima-diabetes	73.2%	2.7	31.3%	72.3%	2.3	14.3%	70.5%	3.0	13.9%	71.9%	3.6	17.2%
thyroid	93.5%	5.0	19.2%	93.5%	3.0	9.9%	92.1%	2.8	9.1%	93.5%	3.4	11.5%
waveform	79.7%	1.0	84.9%	76.1%	1.9	14.6%	75.5%	0.8	13.2%	75.6%	1.0	14.8%
wine	73.1%	5.9	22.5%	94.9%	3.1	13.8%	89.1%	7.9	12.4%	92.0%	4.2	12.7%
wisconson-bc-di	92.9%	1.7	6.4%	93.3%	2.8	4.6%	93.5%	1.7	5.5%	92.6%	1.6	6.8%

is where Hybrid algorithm wins out. Not only does it reduce storage appreciably but also it faithfully maintains the decision boundary.

We have also compared the three algorithms using the data sets available in UCI machine learning depository [7]. The cell data was obtained from the Biomedical Image Processing laboratory at McGill University. The chosen data sets appeared to be the universal choice for performing experimental analysis of IBL algorithms (Wilson and Martinez [13] and Brighton and Mellish [3]). Table 1 shows the particulars of the data sets.

Table 2 gives us a measure of the accuracy of the algorithms. The results are obtained by the so-called cross validation technique. 20% of the data (for the testing purpose) is selected randomly from the data set. The remaining data points (called training set) are used to design the classifier. The experiment was repeated 10 times. The editing step of the algorithms Gabriel and Hybrid uses the approximate Gabriel neighbors. There are two versions of ICF reported in Table 2. The first version ICF-1 implements Wilson editing using *3-NN*-rule. This is similar to the one used in [3]. The second version of ICF-2 implements Wilson editing using *Gabriel-NN*-rule. Thus the edited sets used in Gabriel,

Hybrid and ICF are the same. In all cases the classification is done using the nearest neighbor rule. The distance between feature vectors (numerical, nominal or mixed) are computed using the VDM approach proposed by Wilson and Martinez [13]. The results indicate that ICF algorithm is able to aggressively throw away instances but has higher classification error. The Hybrid algorithm strikes the best balance between classification accuracy and storage reduction.

The results indicate that ICF algorithm is able to aggressively throw away instances but has higher classification error. The Hybrid algorithm strikes the best balance between classification accuracy and storage reduction.

7 Conclusion

The main objective of our study is to show that proximity graphs captures the appropriate proximity information in a data set. However, it is costly to exactly compute the proximity information. A data structure GSASH is proposed which allows one to compute the approximate Gabriel proximity information in an efficient manner. The Gabriel condensed set, using the approximate Gabriel graph, preserves the classification decision boundary remarkably well. We have shown here that ICF algorithm [3] aggressively removes instances from the training set thereby compromising on the quality of the classification correctness. The proposed Hybrid algorithm, based on the approximate Gabriel graph information, is shown to exhibit the best features of both of its constituents. It preserves the decision boundary remarkably well and is also able to aggressively reduce the storage space.

Acknowledgement

The authors acknowledge the help provided by Benjamin Lewis and Yuzhuang Hu.

References

1. Binay Bhattacharya and Damon Kaller. Reference set thinning for the k-nearest neighbor decision rule. In *Proceedings of the 14th International Conference on Pattern Recognition*, volume 1, 1998.
2. Binay K. Bhattacharya. Application of computational geometry to pattern recognition problems. Ph.d. thesis, School of Computer Science, McGill University, 1982.
3. Henry Brighton and Chris Mellish. Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery*, 6:153–172, 2002.
4. Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag New York, Inc., 1996.
5. Michael Houle. SASH: A spatial approximation sample hierarchy for similarity search. Tech. Report RT-0517, IBM Tokyo Research Laboratory, 2003.

6. Sanjeev R. Kulkarni, Gábor Lugosi, and Santosh S. Venkatesh. Learning pattern classification - a survey. *IEEE Transactions on Information Theory*, 44:2178–2206, 1998.
7. C. J. Merz and P. M. Murphy. UCI repository of machine learning database. Internet http://www.ics.uci.edu/mlearn/MLRepository.html, Department of Information and Computer Science, University of California.
8. Kaustav Mukherjee. Application of the gabriel graph to instance-based learning. M.sc. project, School of Computing Science, Simon Fraser University, 2004.
9. Demetri Psaltis, Robert R. Snapp, and Santosh S. Venkatesh. On the finite sample performance of the nearest neighbor classifier. *IEEE Transactions on Information Theory*, 40:820–837, 1994.
10. G. T. Toussaint, B. K. Bhattacharya, and R. S. Poulsen. The application of Voronoi diagrams to nonparametric decision rules. In *Computer Science and Statistics: The Interface*, pages 97–108, Atlanta, 1985.
11. G. T. Toussaint and R. S. Poulsen. Some new algorithms and software implementation methods for pattern recognition research. In *Proc. IEEE Int. Computer Software Applications Conf.*, pages 55–63, Chicago, 1979.
12. D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics*, 2:408–420, 1972.
13. D. Randall Wilson and Tony R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257–286, 2000.