LOCATING SUPPORT VECTORS VIA β-SKELETON TECHNIQUE

Wan Zhang and Irwin King

The Chinese University of Hong Kong
Department of Computer Science and Engineering
Shatin, N.T. Hong Kong

ABSTRACT

Recently, Support Vector Machine (SVM) has become a very dynamic and popular topic in the Neural Network community for its abilities to perform classification, estimation, and regression. One of the major tasks in the SVM algorithm is to locate the points, or rather support vectors, based on which we construct the discriminant boundary in classification task. In the process of studying the methods for finding the decision boundary, we conceive a method, β-skeleton algorithm, which reduces the size of the training set for SVM. We will describe their theoretical connections and practical implementation implications. In this paper, we also survey four different methods for classification: SVM method, K-Nearest neighbor method, β-skeleton algorithm used in the above two methods. Compared with the methods without using β-skeleton algorithm, prediction with the edited set obtained from β-skeleton algorithm as the training set, does not lose the accuracy too much but reduces the real running time.

1. INTRODUCTION

Given a data set which contains the data coming from two or more different classes, either linearly separable or non-separable, the classification problem is to find the optimal separating hyperplane (Decision Boundary) to separate the data according to their class type. Support Vector Machines (SVMs), introduced by Vapnik in the late seventies [10], have attracted wide interest as a means to implement structural risk minimization for the problem of classification and regression estimation. To understand SVMs better, a geometric interpretation from the dual perspective can be useful.

Convex hull could be a simple intuitive geometric explanation of SVM [2]. And also we know that Convex hull could be used to solve many problems, such as half space intersection, Delaunay triangulation, Voronoi diagrams, etc [1].

The Voronoi diagram and Delaunay triangulation are two of the possible representations for K-Nearest neighbor rule. The K-Nearest neighbor rule can be thought as the non-parametric decision rule which needs no prior knowledge of the data points' distributions [3]. Decision rules are used in many areas such as pattern recognition and database. Here, they are used to determine the class membership for a point based on some computational measurements for the point. Despite simplicity and good performance of K-Nearest neighbor rule, the traditional criticism of the method is that it needs a large storage space for the entire training data and the necessity to query the entire training set in order to make a single membership classification. As a result, there has been considerable interest in editing the training set to reduce its size.

Just as SVMs choose support vectors which are a small part of the whole training set to find the separating hyperplane, different proximity graphs (such as Delaunay triangulation and Gabriel graph) provide efficient geometric apparatus for solving the problem and finding the decision boundary. The Gabriel graph of a set of points is a subgraph of Delaunay triangulation for that set, which is a dual of Voronoi diagram [3]. Similarly, Relative neighborhood graph of a set of points is a subgraph of Gabriel graph of that set. And both of Gabriel graph and Relative neighborhood graph can be described by the β-skeleton with the different parameter setting.

This paper tries to bring all these mentioned concepts to a unified application domain. We use Convex hull as a bridge which connects SVM and β-skeleton [12]. Then we use the edited set as the training set for SVM and K-Nearest neighbor method. We also perform experiments to show the performance of the four methods and discuss their advantages and drawbacks.

In the next section, we will present the three main concepts: SVM, K-Nearest neighbor, and β-skeleton and then show the relationship among them. In Section 3, we conduct a series of experiments on different data sets and compare the performance of the different methods. The discussion of these results is shown in Section 4. Lastly, we conclude and make some final remarks in Section 5.

2. RELATED BACKGROUND

In this section, we survey three different concepts for doing classification in a data set. These concepts come from different disciplines in Computer Science, ranging from Computational Geometry to statistical learning theory. We aim to show the similar relationship among these different concepts arising from different disciplines.

2.1. Support Vector Machine

Given the training data, x_i, y_i, $i = 1, \cdots, l$, $y_i \in \{-1, 1\}$, $x_i \in \mathbf{R}^n$, suppose there exists a hyperplane separating the positive from the negative data set. It means that any point x which lies on the hyperplane satisfy $\omega \cdot x + b = 0$, where ω is normal to the hyperplane, $|b|/\|\omega\|$ is the perpendicular distance form the hyperplane to the origin, and $\|\omega\|$ is the Euclidean norm of ω. Define the margin as the sum of the distances of the separating hyperplane to the closest positive and negative points [4]. In fact, the basic concept behind SVM is to find the tradeoff between the largest margin(distance) and training error, so the generalized optimal separating hyperplane is regarded as the solution to Eq. (1) as follows,

$$\min(\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{m}\xi_i) \tag{1}$$

$$\text{subject to } y_i(x \cdot \omega + b) \geq 1 - \xi_i, \xi_i \geq 0, C > 0.$$

Let $\alpha = \alpha_1, \alpha_2, \cdots, \alpha_m$ be the m nonnegative Lagrange multipliers, one for each inequality constraints in Eq. (2), the solution to Eq. (1) equals to the solution to the constrained quadratic optimization problem using the Wolfe dual theory [4] as,

$$\min \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i x_j - \sum_i \alpha_i \tag{2}$$

$$\text{subject to } 0 \leq \alpha_i \leq C, \sum_i \alpha_i y_i = 0.$$

For solving high dimension problems, SVM maps the space of covariates X to a Hilbert space $\mathcal{H}$ of a higher dimension (maybe infinite), and fits an optimal linear classifier in $\mathcal{H}$. It does that by choosing a mapping function ϕ:$\mathbf{R}^n \rightarrow \mathcal{H}$ in such a way that $\phi(x) \cdot \phi(y) = K(x, y)$ for some known and easy-to-evaluate set of functions, K [11]. Set $Q_{ij} = y_i y_j K(x_i, x_j)$, such that $\alpha \cdot y = 0$, $0 \leq \alpha_i \leq C$, $i = 1, \cdots, m$, the objective function is changed as follows:

$$R(\alpha) = \frac{1}{2}\alpha \cdot (Q \cdot \alpha) - \alpha. \tag{3}$$

From a Computational Geometric viewpoint, the solution to the Convex hull problem provides a way to locate support vectors [2]. And it can be used to compute a Delaunay triangulation. Gabriel graph can be computed by discarding edges from Delaunay triangulation. However, according to the time consumed, this approach to obtain the Gabriel graph is not a very attractive one when number of dimensions is large.

2.2. K-Nearest neighbor and β-skeleton

The K-Nearest neighbor classifier is a conventional nonparametric classifier that provides good performance for optimal value of k. In K-Nearest neighbor classification, we classify an object (point) in d-dimensional space according to the dominant class among its k-nearest neighbors from the training data set. It is useful if we can find some representatives from the training set to classify new point while preserving a high accuracy. Both Voronoi diagram and Gabriel graph can be used for such a purpose. The general idea is found in [3].

A Voronoi diagram is a partition of special points into regions such that each region consists of points closer to one particular node than to any other nodes. Therefore, a new point in a Voronoi region must be closer to the region's node than to any other nodes. So, we can assign the new point to the class represented by the region's node. Moreover, the boundaries of the Voronoi regions separating those regions whose nodes are of different class can be used as the decision boundary of the classifier. However, it is clear that the nodes whose boundaries did not contribute to the decision boundary are redundant and can be safely deleted from the training data.

Gabriel graph of a set of points, S, has an edge between points p and q in S if and only if the diametral sphere of p and q does not contain any other points. The resulting points from the above process make up of the Gabriel edited set. We shall see that the decision boundary can be constructed from those Gabriel neighbors (p and q) such that p and q are of different classes.

The Gabriel edited set is always a subset of the Voronoi edited set because of the fact that a Gabriel graph of a set of points is a subgraph of Delaunay triangulation for that set. Thus, Gabriel editing which is the procedure of finding the Gabriel neighbors, reduces the size of the training set more than Voronoi editing. Although, the resulting Gabriel editing does not preserve the original decision boundary, the changes occur mainly outside of the zones of interest.

The parameterized family of neighborhood graphs, introduced by Kirkpatrick and Radke [7], is called β-skeleton. Let V be a set of points in $\mathbf{R}^n$, each pair of points $(p, q) \in V \times V$ with a neighborhood $U_{p,q} \subset \mathbf{R}^n$, let P be a property defined on $U = \{U_{p,q} | (p, q) \in V \times V\}$, $\delta(x, y)$ denotes the distance between point x and y, and $B(x, r)$ denotes the circle centered at x with the radius r. That is to say $B(x, r)$=$\{y | \delta(x, y) < r\}$. A neighborhood graph defined

Table 1: Training and Testing Time Complexity

Methods	Training			Testing
	Best Case	Average Case	Worst Case	Average
SVM	$O(n)$	uncertain	$O(n^5 \log^n / \xi_n)$	$O(c^2 m)$
β-skeleton	$O(dn^2)$	$O(dn^3)$	$O(dn^3)$	/
K-Nearest neighbor	/	/	/	$O(mnlogn)$

Table 2: Problems Description

Problem	class	attribute	training data	testing data
Iris Plants	3	4	150	0
Wine Cultivars	3	13	178	0
Glass	6	9	214	0
Satimage	6	36	4435	2000

on the property P consists of vertices V and the set of edges E, which is required to satisfy the condition that $(p, q) \in E$ if and only if $U_{p,q}$ has the property P. The neighborhood $U_{p,q}(\beta)$ is defined, for any fixed $\beta(1 \leq \beta < \infty)$ as the intersection of two spheres:

$$U_{p,q}(\beta) = B((1-\frac{\beta}{2})p+\frac{\beta}{2}q, \frac{\beta}{2}\delta(p,q)) \cap B((1-\frac{\beta}{2})q+\frac{\beta}{2}p, \frac{\beta}{2}\delta(p,q)).$$

So β-skeleton of V, $G_\beta(V)$, is a neighborhood graph with the set of edges:

$$(p, q) \in E \text{ if and only if } U_{p,q}(\beta) \cap V = \phi.$$

A special feature for this parameterized family is its monotonicity with respect to β, i.e. $G_{\beta_1}(V) \subset G_{\beta_2}(V)$ for $\beta_1 > \beta_2$. So we can easily see that β-skeletons contain Gabriel graph and Relative neighborhood graph. Specially, when β=1, $G_1(V) = GG(V)$, Gabriel graph of V; when β=2, $G_2(V) = RNG(V)$, Relative neighborhood graph of V. According to the feature of β-skeleton, it is easy to see that $RNG(V) \subset GG(V)$ [9].

If we change slightly the definition of the neighborhood by using the different intersection of the spheres, we can obtain a different class of graphs. In the following section, we will design a uniform algorithm for the whole spectrum of β-skeletons for $1 \leq \beta \leq 2$.

3. EXPERIMENTS AND RESULTS

3.1. Time Complexity

Table 1 summarizes the theoretical analysis from the research result of Hush and Scovel [8] and Bhattacharya [3], where n is the size of training set, m is the size of testing set, d is the dimension of samples, C is the number of the classes. Here, ξ_n is obtained through an appropriately normalized objective function(R) and depends on n.

In training ,

1. The best Case of any algorithm is to visit each data point once. For average case analysis with SVM, it typically requires some knowledge of the distribution over problem instances. Moreover, it is not uncommon to see run time estimates varying from n^2 to n^3 reported from experiments with these types of algorithms [8]. But the bounds of iteration steps to find the optimal solution are uncertain.

2. The β-skeleton algorithm requires $O(n^2)$ operations to yield $O(n^2)$ pairs of neighbors. For each such pair of points (A, B), the algorithm requires $O(nd)$ operations. Hence the overall average complexity of the algorithm is $O(dn^3)$.

   From Table 1, we see that β-skeleton is more stable in different cases (linear-separable or nonlinear-separable cases and different data sets) but has poor performance with high-dimension data. Up to now we have not found an exact way to measure the iterations that the algorithm of SVM will take to obtain the optimal solution. In other words, the performance of SVM is data-sensitive, i.e., it changes with different data sets. The dimension has little effect on SVM.

3. The worst case for SVM is that in this algorithm of SVM, the criterion function converge to the optimum solution in $O(\frac{Cn^4}{\xi_n})$ iterations. In each step it will take $O(n \log n)$ time to see whether the result satisfies the constraints for the optimal solution. Although this does not happen often in the real world, it should be considered when implementing the algorithm.

We do the following empirical experiments with several datasets: Iris dataset, Wine Cultivar discrimination, Glass identification data set and Satimage database. The three databases are available via anonymous file transfer protocol(ftp) from the University of California Irvine UCI Repository of machine learning databases. The last one is from Statlog collection. And in the Satimage database, there is one missing class. That is, there are no examples with one class in this dataset. We scale all training data to be in [-1,1] for SVM method. Table 2 gives out the number of classes, attributes and size of each database.

First, we use Libsvm [5], an integrated software for support vector classification, (C-SVC, nu-SVC), regression (xi-SVR, nu-SVR) and distribution estimation (one-class SVM) to implement SVM method. It provides both C++ and Java source codes. To reduce the search space of the parameter sets, we train all datasets only with the RBF kernel function. Better solutions may result with different choice of γ and C. For each problem, we estimate the generalized accuracy using different parameters C and γ: $\gamma = [2^4, 2^3, \ldots, 2^{-10}]$ and $C = [2^12, 2^11, \ldots, 2^{-2}]$ [6]. Then we can use the accuracy as criterion to choose the optimal parameters. For all

Table 3: SVM parameters setting and Accuracy for different dataset

Problem	Error Penalty	Gamma
Iris	2^{12}	2^{-9}
Wine	2^{7}	2^{-10}
Glass	2^{11}	2^{-2}
Satimage	2^{1}	2^{0}
Sat_{GG}	2^{1}	2^{-1}
$\text{Sat}_{\beta 1.5}$	2^{1}	2^{-1}
Sat_{RNG}	2^{0}	2^{-2}

the datasets where test data may not be available, we simply conduct a 10-fold cross-validation on the whole training data to estimate generalization on future data and report the cross-validation rate(See Table 3).

Second, we implement β-skeleton algorithm with Visual C++ in an uniform algorithm template and with different values of β, we can get the different proximity graph. When β is set as 1, it uses the Gabriel graph algorithm; while β equals to 2, it represents the Relative neighborhood graph algorithm. We try some different values between 1 and 2, then we can obtain some distinct graphs, which use the diverse nearest neighbor rules to define the neighbor in the graph.

Third, the K-Nearest neighbor algorithm is implemented in C language. For each problem we use the different value of k from 1 to 15 to obtain the optimal result, which has the highest prediction accuracy.

Then for the first three datasets, we choose the set of support vectors with highest cross-validation accuracy and do the experiments to record and compare the similarity among the set of support vectors and several edited sets of β-skeleton with different β value(See Table 4, 5). In the following step, we do the four kind of experiments on Satimage database, which include the training data and testing data, and then compare the prediction accuracy of the four experiments.(See Table 6, H means hour.) For each method, we need to do model selection step. First, the training set for SVM is scaled Satimage database. Second, the edited set obtained from the β-skeleton algorithm is used as the training set for SVM. Third, we use K-Nearest neighbor method to do prediction without training step. Fourth, after we obtain the edited set from the β-skeleton algorithm, we do prediction by K-Nearest neighbor method. All the experiments were done under WINNT operating system, Pentium4 1.4G, 512MB memory.

Table 4 shows the number of support vectors for SVM algorithm and the size of edited set for the other two algorithms, the edited size of β-skeleton with β=1.4/1.5(Satimage). Table 5 demonstrates the relationship among support vectors, points left in edited set of Gabriel graph, points in the edited set of Relative neighborhood graph, and points in edited set of β-skeleton, with β=1.4.

Table 4: Size of Database, Number of Support Vectors, Size of the edited sets

Dataset	Size of Data set	Support Vectors	GG Edited Set	RNG Edited Set	β-skeleton Edited Set
Iris	150	18	49	18	32
Wine	178	60	150	38	90
Glass Data	214	120	196	121	170
Satimage	4435	1615	3654	1191	2441

Table 5: Number of intersection between Support Vectors and Points in other Edited Sets

		Number of Intersection with SV		
Dataset	Number of SV	$V \in$GG(V)	$V \in$ RNG	$V \in \beta$=1.4
Iris	18	18	12	16
Wine	60	60	34	54
Glass	120	120	92	113

Table 6: Time Consuming and Accuracy Compare

Method	Training time	Testing Time	Accuracy
SVM	37	18	91.85
SVM_{GG}	2H+22	16	89.4
$SVM_{\beta 1.5}$	2H 15	16	89.7
SVM_{RNG}	2H+5	11	85.8
KNN(k=7)	0	137	72.1
$KNN_{GG}(k=4)$	2H	112	70.35

4. DISCUSSIONS

As described in above sections, the Convex hull could be one intuitive geometric explanation for SVM and it also can be used to solve the Voronoi diagram problem. In fact, the edited set of Voronoi diagram is the super set of the edited set of Gabriel graph. There exist some relationships between SVM and β-skeleton.

Observation #1 From the experiments on several dataset, it is always true that RNG(V) $\subset$ β-skeleton(V)($\beta = 1.4$) $\subset$ GG(V).

Observation #2 From the experiments(See Table 4), we observe that the number of support vectors is always smaller than the size of the Gabriel graph edited set, and approximately equals to or larger than the size of the Relative neighborhood edited set. When we record the exact points, which are the intersection of support vectors and the points in the GG edited set and RNG edited set, we find that support vectors are always in the subset of the Gabriel edited set, and have some intersection with the Relative neighbor-

hood edited set. And when we choose β value as 1.4, we can also obtain some edited set whose components have more common intersection with support vectors(See Table 5). If we choose some other β values between 1 and 2, whether the edited set of the corresponding graph will be more similar with the support vectors is a point worth to consider [12].

Observation #3 We may find that the prediction accuracy with the edited set as the training set for SVM is not lowered too much while the size of the training set is reduced hugely. While the speed of training data with β-skeleton is slower than that of SVM from Table reftimecompare. So We will do paralleling to speed up the β-skeleton algorithm in future work.

Observation #4 According to the observation 2,3 and if the conclusion can be the same with the experiments on most of the other datasets, we could improve SVM algorithm with the help of β-skeleton algorithm as follows:

1. Obtain the edited set with β-skeleton algorithm (paralleled).
2. Use the edited set instead of the training set in SVM method to construct SVM-model and do the prediction.

As a result of the steps above, the reduced training data will accelerate the convergence of finding the optimal quadratic solution.

5. CONCLUSION

In this paper, we have demonstrated how the SVM, β-skeleton and K-mean can be used to solve the classification problem. Moreover, we have shown the relationships among these algorithms. When used as training set in SVM method or K-Nearest neighbor method, β-skeleton will reduce the training set size and not lower the accuracy of prediction with the optimal parameters through empirical observations. We could improve SVM's performance in general by using the β-skeleton's training data set reduction algorithm. In light of this, we plan to investigate the following in future:

1. Given more dataset, compare the points in different edited sets with support vectors and prediction accuracy for different kind of training sets.
2. Theoretically prove that support vectors are the subset of Gabriel edited set.
3. Parallel the β-skeleton algorithm.

Acknowledgement

This research is supported in part by an Earmarked Grant from the Hong Kong Research Grants Committee (RGC), CUHK #4407/99E.

6. REFERENCES

[1] C. Bradford Barber, David P. Dobkin, and Hannu Huhanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.

[2] Kristin P. Bennett and Erin J. Bredensteiner. *Duality and Geometry in SVM Classifiers*. Morgan Kaufmann, San Francisco, CA, 2000.

[3] Binay K. Bhattacharya, Ronald S. Poulsen, and Godfried T. Toussaint. *Application of Proximity Graphs to Editing Nearest Neighbor Decision Rule*. International Symposium on Information Theory, Santa Monica, 1981.

[4] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[5] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a Library for Support Vector Machines (Version 2.31)*. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2001.

[6] Chih-Chung Chang and Chih-Jen Lin. A comparison on methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 2002.

[7] J. D. Radke. D.G.Kirkpatrick. *A Framework for computational morphology*. In G. T. Toussaint, editor, Computational Geometry, NorthHolland, Amsterdam, Netherlands, 1985.

[8] Don Hush and Clint Scovel. Polynomial-time decomposition algorithms for support vector machines, 2000.

[9] J.W. Jaromczyk and G.T. Toussaint. Relative neighborhood graphs and their relatives. *Proceedings IEEE*, 80(9):1502–1517, 1992.

[10] V. N. Vapnik. *Estimation of dependencies based on empirical Data*. (in Russian), Nauka, Moscow, 1979.

[11] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, Heidelberg, DE, 1995.

[12] wan Zhang and Irwin King. A study of the relationship between support vector machine and gabriel graph, May 2002.