A Support Vector Method for Hierarchical Clustering

Asa Ben-Hur
Faculty of IE and Management
Technion, Haifa 32000, Israel

David Horn
School of Physics and Astronomy
Tel Aviv University, Tel Aviv 69978, Israel

Hava T. Siegelmann
Faculty of IE and Management
Technion, Haifa 32000, Israel

Vladimir Vapnik
AT&T Labs Research
100 Schultz Dr., Red Bank, NJ 07701, USA

Abstract

We present a novel method for clustering using the support vector machine approach. Data points are mapped to a high dimensional feature space, where support vectors are used to define a sphere enclosing them. The boundary of the sphere forms in data space a set of closed contours containing the data. As the kernel parameter is varied these contours fit the data more tightly and splitting of contours occurs. The contours are interpreted as cluster boundaries and the points within each disconnected contour are defined as a cluster. Cluster boundaries can take on arbitrary geometrical shapes and clusters are separated by valleys in the underlying probability distribution. As in other SV algorithms, outliers can be dealt with by introducing a soft margin constant leading to smoother cluster boundaries. The hierarchical structure of the data is explored by varying the two parameters. We investigate the dependence of our method on these parameters and apply it to several data sets.

1 Introduction

Clustering is an ill-defined problem for which there exist numerous methods [1, 3]. These can be based on parametric models or on non-parametric criteria. Parametric algorithms are usually limited in their expressive power, i.e. a certain cluster structure is assumed, including the number of clusters. Two recent examples of non-parametric hierarchical methods are the paramagnetic model of [4], with temperature serving as the hierarchy parameter and magnetic regions defining the clusters, and the approach of [5] based on successive normalization of the dissimilarity matrix between pairs of points.

In two recent papers [6, 7] it was suggested that the Support Vector (SV) approach, used until then only in learning of labeled data [10] can be used for data exploration of unlabeled data. They propose an algorithm, for characterizing the support of a high dimensional distribution. As a by-product of the algorithm one can compute a set of contours that

enclose the data points. These contours were interpreted by us as cluster boundaries [8], where we formulate an approach like [9] in the SVM language. The shape of the contours is regulated by the width parameter of the kernel function. The points inside each separate piece are interpreted as beloging to the same cluster. The hierarchical organization of the data is explored by varying the width parameter and the soft margin constant. In this paper we identify clusters by valleys in the probability distribution of the data. Other work that uses this paradigm to identify clusters is found in [3].

2 Describing Cluster Boundaries with Support Vectors

In this section we present an algorithm for describing the support of a probability distribution represented by a finite data set [7, 6] that forms the basis of our clustering algorithm. Let $\{\mathbf{x}_i\} \subseteq \chi$ be a data-set of N points, with $\chi \subseteq \mathbb{R}^d$, the input space. Using a nonlinear transformation Φ from χ to some high dimensional feature-space, we look for the smallest enclosing sphere of radius R. This is described by the constraints: $||\Phi(\mathbf{x}_i) - \mathbf{a}||^2 \leq R^2 \ \forall i$, where $||\cdot||$ is the Euclidean norm and $\mathbf{a}$ is the center of the sphere. Outliers can be allowed by incorporating soft constraints [10, 12]:

$$||\Phi(\mathbf{x}_j) - \mathbf{a}||^2 \leq R^2 + \xi_j, \tag{1}$$

with $\xi_j \geq 0$. To solve this problem we introduce the Lagrangian

$$L = R^2 - \sum_j (R^2 + \xi_j - ||\Phi(\mathbf{x}_j) - \mathbf{a}||^2)\beta_j - \sum \xi_j \mu_j + C \sum \xi_j \ , \tag{2}$$

where $\beta_j \geq 0$ and $\mu_j \geq 0$ are Lagrange multipliers, C is a constant, and $C \sum \xi_j$ is a penalty term. Minimization with respect to R, ξ_j and $\mathbf{a}$ leads to:

$$\sum_j \beta_j = 1, \qquad \beta_j = C - \mu_j, \tag{3}$$

$$\mathbf{a} = \sum_j \beta_j \Phi(\mathbf{x}_j). \tag{4}$$

The KKT complementarity conditions are: $\xi_j \mu_j = 0$ and $(R^2 + \xi_j - ||\Phi(\mathbf{x}_j) - \mathbf{a}||^2)\beta_j = 0$. An outlier is a point $\mathbf{x}_i$ with $\xi_i > 0$. Equation (3) states that outliers have $\mu_i = 0$, hence $\beta_i = C$. A point with $\xi_i = 0$ is inside or on the surface of the sphere in feature space. If its $\beta_i > 0$ then it follows from the KKT conditions the point $\mathbf{x}_i$ is on the surface of the sphere. These are defined by us as support vectors. Next we eliminate the variables R, $\mathbf{a}$ and μ_j, turning the Lagrangian into the Wolfe dual form:

$$W = \sum_j \Phi(\mathbf{x}_j)^2 \beta_j - \sum_{i,j} \beta_i \beta_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{5}$$

Since the variables μ_j don't appear in the Lagrangian they may be replaced with the constraints $0 \leq \beta_j \leq C$. This is the SV minimization problem that we are solving.

We follow the SV method and represent the dot products $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ by a Mercer kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ [10]. As noted in [7], poynomial kernels do not yield tight contour representations of a cluster. In this paper we use the Gaussian kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-q||\mathbf{x}_i - \mathbf{x}_j||^2} \ , \tag{6}$$

or the Laplacian kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-q'||\mathbf{x}_i - \mathbf{x}_j||} \ , \tag{7}$$

with width parameters q and q'. The Lagrangian W is now written as:

$$W = \sum_j K(\mathbf{x}_j, \mathbf{x}_j)\beta_j - \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j). \tag{8}$$

For a translationally invariant kernel (e.g. Gaussian), this minimization problem is equivalent to the standard SV optimization problem [6].

At each point $\mathbf{x}$ we define the distance of its feature space image from the center of the sphere:

$$R^2(\mathbf{x}) = \|\Phi(\mathbf{x}) - \mathbf{a}\|^2 \ . \tag{9}$$

In view of (4) and the definition of the kernel we have:

$$R^2(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}) - 2\sum_j \beta_j K(\mathbf{x}_j, \mathbf{x}) + \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j). \tag{10}$$

The radius of the sphere is:

$$R = \{R(\mathbf{x}_i) \mid \mathbf{x}_i \text{ is a support vector } \} \ , \tag{11}$$

and the contour that encloses the points in data space is the set

$$\{\mathbf{x} \mid R(\mathbf{x}) = R\} \ . \tag{12}$$

The shape of the contour is governed by q and C. Figure 3 shows that as q is increased, the enclosing contour forms a tighter fit to the data, and the number of SVs increases. For fixed q, as C is decreased the number of SVs decreases since ignoring outliers gives a smoother shape. This will be further discussed in section 3.

Let us denote by n_{sv}, n_{out} the number of support vectors and outliers, respectively, and note the following results that are a consequence of the constraints:

Proposition 2.1 [6]

$$n_{out} + n_{sv} \geq 1/C, \ \ n_{out} < 1/C \tag{13}$$

We now discuss the generalization ability of this algorithm: given data points generated from the same probability distribution as the original data set, the probability of error is the expected fraction of points which are not contained in the high dimensional sphere. It can be estimated by

$$P(\text{error}) = \frac{n_{out} + n_{sv}}{N} \ , \tag{14}$$

using the standard leave one out argument, with the addition of outliers [10, 7].

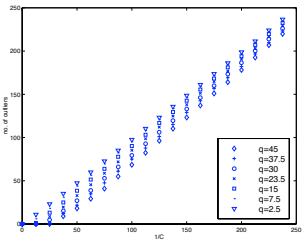

Figure 1: Number of outliers as a function of $1/C$.

We next characterize the dependence of n_{out} and n_{sv} on the parameters q and C. The dependence was found empirically: we generated data sets of 500 points sampled from a uniform distribution on the unit square. For each data set n_{out} and n_{sv} were computed and

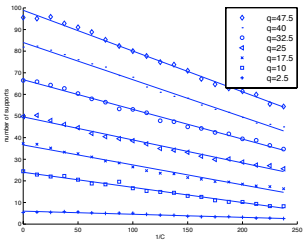

Figure 2: Number of supports as a function of $1/C$.

averaged over 20 data sets. We found similar behavior for other probability distributions as well. The first observation is that the number of outliers depends only weakly on q (Figure 1), and satisfies:

$$n_{out}(q, C) = \max(0, 1/C - n_0) \,, \tag{15}$$

where $n_0 > 0$ is a function of q and N. We found that n_0 has a linear dependence on q. The linear behavior of n_{out} continues until $n_{out} + n_{sv} = N$.

From Figure 2 we find:

$$n_{sv} = a/C + b \,, \tag{16}$$

where a and b are functions of q and N.

3 Hierarchical Clustering

In this section we go through a set of examples to demonstrate the clustering ability of our algorithm. We begin with a data set in which the separation into clusters can be seen without allowing for outliers, i.e. $1/C = 1$. As seen in Figure 3, as q is increased the shape of the boundary curve in data-space varies. At several q values the enclosing contour splits forming an increasing number of connected components. We regard each component as representing a single cluster.

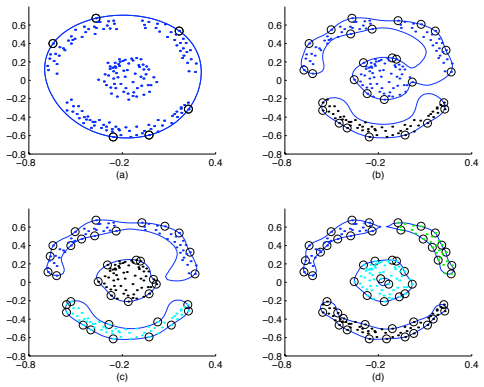

Figure 3: Data set contains 183 points. A Gaussian kernel was used with $1/C = 1.0$ (a): $q = 1$ (b): $q = 20$ (c): $q = 24$ (d): $q = 48$.

We define an adjacency matrix A_{ij} between pairs of points $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$A_{ij} = \begin{cases} 1 & \text{if for all } \mathbf{y} \text{ on the line segment connecting } \mathbf{x}_i \text{and } \mathbf{x}_j \; R(\mathbf{y}) \leq R \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

Clusters are now defined as the connected components of the graph induced by A. This labeling procedure is justified by the empirical observation that the line segment connecting points in different components contain points outside the sphere whereas the line connecting "close neighbors" in the same component lies inside the sphere. Checking the line segment is implemented by sampling a number of points (a value of 10 was used in the numerical experiments). Outliers can be left unclassified, or labeled according to the cluster to which they are closest to, as we choose here.

3.1 The role of C

In this sub-section we demonstrate the importance of allowing outliers by setting a value of $1/C > 1$. The clusters in the data in Figure 3 were distinguished without outliers since they are well separated. We now consider the task of separating a mixture of two Gaussians. When no outliers are allowed small single point clusters are formed while the two main clusters are distinguished (Figure 4(a)). Figure 4(b) demonstrate a clustering solution for the same value of q for a higher value of $1/C$ that allows for enough outliers. Another example is given in Figure 4(c-d). When no outliers are allowed only the smallest

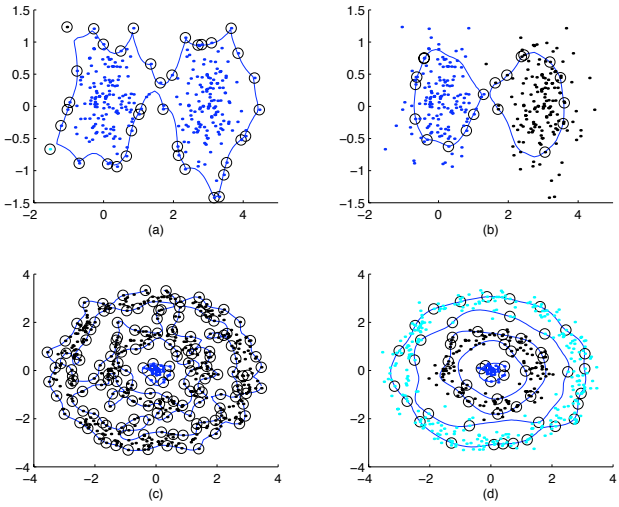

Figure 4: Upper figures: data is a 200 point mixture of two gaussians. (a) Laplacian kernel with $q = 0.7, 1/C = 1$. (b) Laplacian kernel with $q = 0.7, 1/(CN) = 0.2$. Lower figures: The inner sphere is composed of 50 pts with components with a Gaussian distribution; the two concentric rings contain 150/300 pts with uniform angular distribution and Gaussian radial distribution. (c) Gaussian kernel with $q = 3.5, 1/C = 1$. (d) Gaussian kernel with $q = 1.0, 1/(CN) = 0.3$.

cluster is distinguished regardless of the value of q. For $1/(NC) = 0.3$ separation occurs at $q = 1.0$

3.2 The iterative process

There are two types of hierarchical clustering algorithms: agglomerative algorithms and divisive ones [1]. We can use our cluster identification algorithm both ways: In an agglomerative manner, starting from a large value of q, where each point is in a different cluster, and decreasing its value until there is a single cluster, or in the divisive approach, start-

ing from a small value of q and increasing it. The latter seems to be more efficient since "meaningful" clustering solutions usually have a relatively small number of clusters.

Here we describe a qualitative schedule for varying the parameters. We start with a value of q where one cluster occurs, and increase it to detect cluster splitting. When single point clusters start to break off (Figure 4(a)) or a large number of support vectors is obtained (overfitting, as in Figure 4(c)) $1/C$ is increased.

For a decision when to stop dividing the approach described in [5] can be used: After clustering they partition the data into two sets with a sizable overlap, they perform clustering on these smaller data sets and compute the average overlap between the two clustering solutions for a number of partitions. Such validation can be performed here as well. Another approach is to halt when the fraction of outliers and support vectors exceeds some threshold. This is justified by the relation between this fraction and generalization. Alternatively we can use relations such as equations (16)and (15) to pick their values in advance without the iterative process.

3.3 Complexity and Performance

The quadratic programming problem of equation (2) can be solved by the SMO algorithm [14] which was recently proposed as an efficient tool for SVM training. Some minor modifications are required to adapt it to the problem that we solve here [6]. Benchmarks reported in [14] show that this algorithm converges after $O(N) - O(N^{2.3})$ kernel evaluations, depending on the type of data and the parameters. The complexity of the labeling part of our algorithm is $O(N^2 d)$, so that the overall complexity is $O(N^{2.3} d)$. We also note that the memory requirements of the SMO algorithm are low - it can be implemented using $O(1)$ memory at the cost of a decrease in efficiency, which makes our algorithm useful even for very large data-sets.

To compare the performance of our algorithm with other hierarchical algorithms we ran it on the Iris data set [15], which is a standard benchmark in the pattern recognition literature. It can be obtained from the UCI repository [16]. The data set contains 150 instances each containing four measurements of an iris flower. There are three types of flowers, represented by 50 instances each. One of the clusters is linearly separable from the other two, and was easily separated for low values of q. At higher values the two remaining clusters were separated ($q' = 3.4$ for the Laplacian kernel and $q = 10.7$ for the Gaussian kernel). In this case the clustering solution was not perfect: 15/25 points were misclassified for the Gaussian and Laplacian kernels, respectively. All misclassifications were in the outliers. This can be compared with 15 outliers for the super-paramagnetic clustering algorithm [4], and 26 misclassifications for the iterative algorithm in [5].

4 Discussion

The algorithm described in this paper finds clustering solutions by discovering gaps in the probability distribution of the data. Thus clusters with a sizable overlap cannot be distinguished using our algorithm.

An advantage of our algorithm is that it can represent clusters of arbitrary shape, whereas other algorithms that use a geometric representation are most often limited to hyper-ellipsoids [1]. In this respect it is reminiscent of the method of [9]. Our algorithm has a distinct advantage over this algorithm in that it is a kernel method, hence explicit calculations in feature-space are avoided, leading to higher efficiency.

Finally we wish to stress the possibility of our method to deal with outliers. In some algorithms which minimize error criteria, such as a square-error criterion, outliers can alter

a clustering solution [1], whereas these do not affect the clustering solution obtained by our algorithm when $1/C$ is chosen large enough. Using our estimates for the numbers of outliers and support-vectors as functions of q and C it can be designed to best fit any clustering problem.

Our empirical results indicate that the cluster description algorithm can be used to obtain a better theoretical understanding of SVMs in general - the number of support vectors and outliers has a well defined dependence on the kernel and soft margin parameters. In addition we found interesting differences between cluster descriptions produced by the Laplacian and Gaussian kernels: in numerical experiments we performed we found that domains generated by the Laplacian kernel didn't have "holes" as observed in those of the Gaussian kernel, leading to better generalization performance in some cases. However, for problems with many local features, the Gaussian kernel was found to perform better.

References

[1] A.K. Jain and R.C. Dubes. *Algorithms for clustering data*. Prentice Hall, Englewood Cliffs, NJ, 1988.

[2] R.O. Duda and P.E. Hart. *Pattern classification and scene analysis*. Wiley-Interscience, 1973.

[3] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, CA, 1990.

[4] M. Blatt, S. Wiseman, and E. Domany. Data clusteringusinga model granula magnet. *Neural Computation*, 2000.

[5] S. Dubnov, R. El-Yaniv, Y. Gdalyahu, E. Schneidman, N. Tishby, and G. Yona. A new nonparametric pairwise clustering algorithm. *Machine Learning*.

[6] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high dimensional distribution. In *Proceedings of the Annual Conference on Neural Information Systems 1999 (NIPS*99)*. MIT Press, 2000.

[7] D.M.J. Tax and R.P.W. Duin. Support vector domain description. *Pattern Recognition letters*, 20:1991–1999, 1999.

[8] A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik. A support vector clustering method.

[9] H. Lipson and H.T. Siegelmann. Clustering irregular shapes using high-order neurons. *Neural Computation*, 1999.

[10] V Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[11] B. Schölkopf. *Support Vector Learning*. R. Oldenburg Verlag, 1997.

[12] B. Schölkopf, C.J.C. Burgess, and A.J. Smolla, editors. *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

[13] S. Saitoh. *Theory of reproducing kernels and its applications*. Longman Scientific & Technical, 1988.

[14] J. Platt. Fast training of SVMs using sequential minimal optimization. In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 185–208. MIT Press, Cambridge, MA, 1999.

[15] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annual Eugenics*, 7, Part II:179–188, 1936.

[16] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.