Journal of Machine Learning Research 2 (2001) 125-137 Submitted 3/04; Published 12/01

Support Vector Clustering

Asa Ben-Hur ASA@BARNHILLTECHNOLOGIES.COM
BIOwulf Technologies
2030 Addison st. suite 102, Berkeley, CA 94704, USA

David Horn HORN@POST.TAU.AC.IL
School of Physics and Astronomy
Raymond and Beverly Sackler Faculty of Exact Sciences
Tel Aviv University, Tel Aviv 69978, Israel

Hava T. Siegelmann HAVA@MIT.EDU
Lab for Information and Decision Systems
MIT Cambridge, MA 02139, USA

Vladimir Vapnik VLAD@RESEARCH.ATT.COM
AT&T Labs Research
100 Schultz Dr., Red Bank, NJ 07701, USA

Editor: Nello Critianini, John Shawe-Taylor and Bob Williamson

Abstract

We present a novel clustering method using the approach of support vector machines. Data points are mapped by means of a Gaussian kernel to a high dimensional feature space, where we search for the minimal enclosing sphere. This sphere, when mapped back to data space, can separate into several components, each enclosing a separate cluster of points. We present a simple algorithm for identifying these clusters. The width of the Gaussian kernel controls the scale at which the data is probed while the soft margin constant helps coping with outliers and overlapping clusters. The structure of a dataset is explored by varying the two parameters, maintaining a minimal number of support vectors to assure smooth cluster boundaries. We demonstrate the performance of our algorithm on several datasets.

Keywords: Clustering, Support Vectors Machines, Gaussian Kernel

1. Introduction

Clustering algorithms group data points according to various criteria , as discussed by Jain and Dubes (1988), Fukunaga (1990), Duda et al. (2001). Clustering may proceed according to some parametric model, as in the k-means algorithm of MacQueen (1965), or by grouping points according to some distance or similarity measure as in hierarchical clustering algorithms. Other approaches include graph theoretic methods, such as Shamir and Sharan (2000), physically motivated algorithms, as in Blatt et al. (1997), and algorithms based on density estimation as in Roberts (1997) and Fukunaga (1990). In this paper we propose a non-parametric clustering algorithm based on the support vector approach of

Vapnik (1995). In Schölkopf et al. (2000, 2001), Tax and Duin (1999) a support vector algorithm was used to characterize the support of a high dimensional distribution. As a by-product of the algorithm one can compute a set of contours which enclose the data points. These contours were interpreted by us as cluster boundaries in Ben-Hur et al. (2000). Here we discuss in detail a method which allows for a systematic search for clustering solutions without making assumptions on their number or shape, first introduced in Ben-Hur et al. (2001).

In our Support Vector Clustering (SVC) algorithm data points are mapped from data space to a high dimensional feature space using a Gaussian kernel. In feature space we look for the smallest sphere that encloses the image of the data. This sphere is mapped back to data space, where it forms a set of contours which enclose the data points. These contours are interpreted as cluster boundaries. Points enclosed by each separate contour are associated with the same cluster. As the width parameter of the Gaussian kernel is decreased, the number of disconnected contours in data space increases, leading to an increasing number of clusters. Since the contours can be interpreted as delineating the support of the underlying probability distribution, our algorithm can be viewed as one identifying valleys in this probability distribution.

SVC can deal with outliers by employing a soft margin constant that allows the sphere in feature space not to enclose all points. For large values of this parameter, we can also deal with overlapping clusters. In this range our algorithm is similar to the scale space clustering method of Roberts (1997) that is based on a Parzen window estimate of the probability density with a Gaussian kernel function.

In the next Section we define the SVC algorithm. In Section 3 it is applied to problems with and without outliers. We first describe a problem without outliers to illustrate the type of clustering boundaries and clustering solutions that are obtained by varying the scale of the Gaussian kernel. Then we proceed to discuss problems that necessitate invoking outliers in order to obtain smooth clustering boundaries. These problems include two standard benchmark examples.

2. The SVC Algorithm

2.1 Cluster Boundaries

Following Schölkopf et al. (2000) and Tax and Duin (1999) we formulate a support vector description of a data set, that is used as 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 data 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:

$$||\Phi(\mathbf{x}_j) - \mathbf{a}||^2 \leq R^2 \quad \forall j \ ,$$

where $||\cdot||$ is the Euclidean norm and $\mathbf{a}$ is the center of the sphere. Soft constraints are incorporated by adding slack variables ξ_j:

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

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. Setting to zero the derivative of L with respect to R, $\mathbf{a}$ and ξ_j, respectively, leads to

$$\sum_j \beta_j = 1 \tag{3}$$

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

$$\beta_j = C - \mu_j \ . \tag{5}$$

The KKT complementarity conditions of Fletcher (1987) result in

$$\xi_j \mu_j = 0, \tag{6}$$

$$(R^2 + \xi_j - ||\Phi(\mathbf{x}_j) - \mathbf{a}||^2)\beta_j = 0. \tag{7}$$

It follows from Eq. (7) that the image of a point $\mathbf{x}_i$ with $\xi_i > 0$ and $\beta_i > 0$ lies outside the feature-space sphere. Eq. (6) states that such a point has $\mu_i = 0$, hence we conclude from Eq. (5) that $\beta_i = C$. This will be called a *bounded support vector* or BSV. A point $\mathbf{x}_i$ with $\xi_i = 0$ is mapped to the inside or to the surface of the feature space sphere. If its $0 < \beta_i < C$ then Eq. (7) implies that its image $\Phi(\mathbf{x}_i)$ lies on the surface of the feature space sphere. Such a point will be referred to as a *support vector* or SV. SVs lie on cluster boundaries, BSVs lie outside the boundaries, and all other points lie inside them. Note that when $C \geq 1$ no BSVs exist because of the constraint (3).

Using these relations we may eliminate the variables R, $\mathbf{a}$ and μ_j, turning the Lagrangian into the Wolfe dual form that is a function of the variables β_j:

$$W = \sum_j \Phi(\mathbf{x}_j)^2 \beta_j - \sum_{i,j} \beta_i \beta_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j). \tag{8}$$

Since the variables μ_j don't appear in the Lagrangian they may be replaced with the constraints:

$$0 \leq \beta_j \leq C, \ j = 1, \ldots, N. \tag{9}$$

We follow the SV method and represent the dot products $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ by an appropriate Mercer kernel $K(\mathbf{x}_i, \mathbf{x}_j)$. Throughout this paper we use the Gaussian kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-q||\mathbf{x}_i - \mathbf{x}_j||^2} \ , \tag{10}$$

with width parameter q. As noted in Tax and Duin (1999), polynomial kernels do not yield tight contours representations of a cluster. The Lagrangian W is now written as:

$$W = \sum_j K(\mathbf{x}_j, \mathbf{x}_j)\beta_j - \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j). \tag{11}$$

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

$$R^2(\mathbf{x}) = ||\Phi(\mathbf{x}) - \mathbf{a}||^2 \ . \tag{12}$$

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

$$R^2(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}) - 2\sum_j \beta_j K(\mathbf{x}_j, \mathbf{x}) + \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \ . \tag{13}$$

The radius of the sphere is:

$$R = \{R(\mathbf{x}_i) \mid \mathbf{x}_i \text{ is a support vector } \} \ . \tag{14}$$

The contours that enclose the points in data space are defined by the set

$$\{\mathbf{x} \mid R(\mathbf{x}) = R\} \ . \tag{15}$$

They are interpreted by us as forming cluster boundaries (see Figures 1 and 3). In view of equation (14), SVs lie on cluster boundaries, BSVs are outside, and all other points lie inside the clusters.

2.2 Cluster Assignment

The cluster description algorithm does not differentiate between points that belong to different clusters. To do so, we use a geometric approach involving $\mathbf{R}(\mathbf{x})$, based on the following observation: given a pair of data points that belong to different components (clusters), any path that connects them must exit from the sphere in feature space. Therefore, such a path contains a segment of points $\mathbf{y}$ such that $R(\mathbf{y}) > R$. This leads to the definition of the adjacency matrix A_{ij} between pairs of points $\mathbf{x}_i$ and $\mathbf{x}_j$ whose images lie in or on the sphere in feature space:

$$A_{ij} = \begin{cases} 1 & \text{if, for all } \mathbf{y} \text{ on the line segment connecting } \mathbf{x}_i \text{ and } \mathbf{x}_j, \ R(\mathbf{y}) \leq R \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Clusters are now defined as the connected components of the graph induced by A. Checking the line segment is implemented by sampling a number of points (20 points were used in our numerical experiments).

BSVs are unclassified by this procedure since their feature space images lie outside the enclosing sphere. One may decide either to leave them unclassified, or to assign them to the cluster that they are closest to, as we will do in the examples studied below.

3. Examples

The shape of the enclosing contours in data space is governed by two parameters: q, the scale parameter of the Gaussian kernel, and C, the soft margin constant. In the examples studied in this section we will demonstrate the effects of these two parameters.

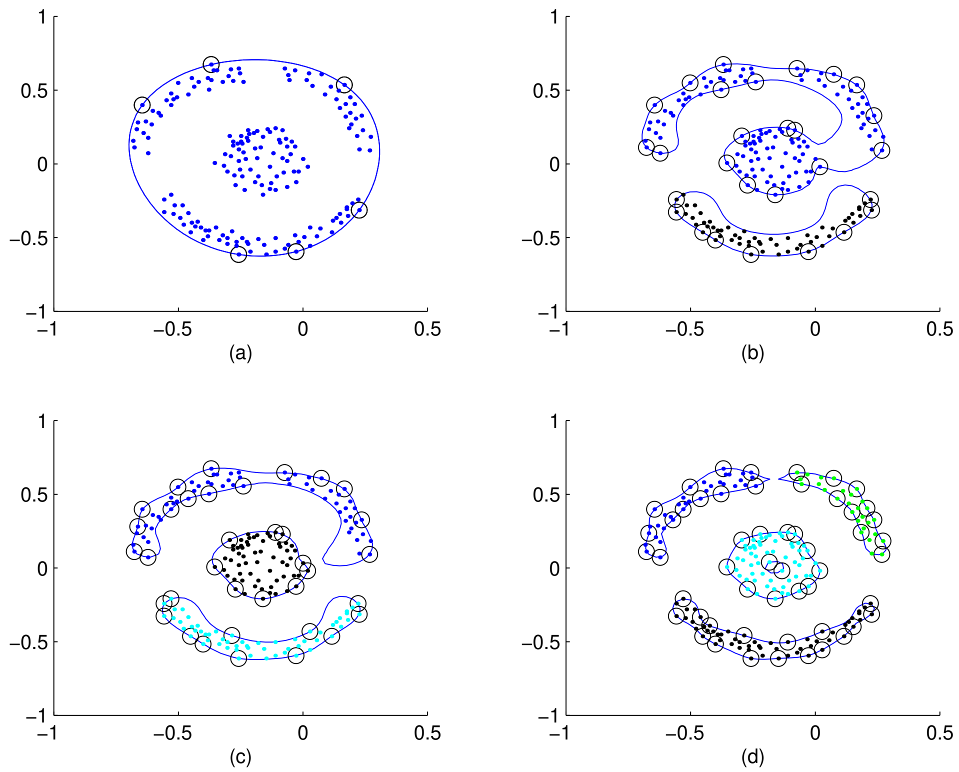

Figure 1: Clustering of a data set containing 183 points using SVC with $C = 1$. Support vectors are designated by small circles, and cluster assignments are represented by different grey scales of the data points. (a): $q = 1$ (b): $q = 20$ (c): $q = 24$ (d): $q = 48$.

3.1 Example without BSVs

We begin with a data set in which the separation into clusters can be achieved without invoking outliers, i.e. $C = 1$. Figure 1 demonstrates that as the scale parameter of the Gaussian kernel, q, is increased, the shape of the boundary in data-space varies: with increasing q the boundary fits more tightly the data, and at several q values the enclosing contour splits, forming an increasing number of components (clusters). Figure 1a has the smoothest cluster boundary, defined by six SVs. With increasing q, the number of support vectors n_{sv} increases. This is demonstrated in Figure 2 where we plot n_{sv} as a function of q for the data considered in Figure 1.

3.2 Example with BSVs

In real data, clusters are usually not as well separated as in Figure 1. Thus, in order to observe splitting of contours, we must allow for BSVs. The number of outliers is controlled

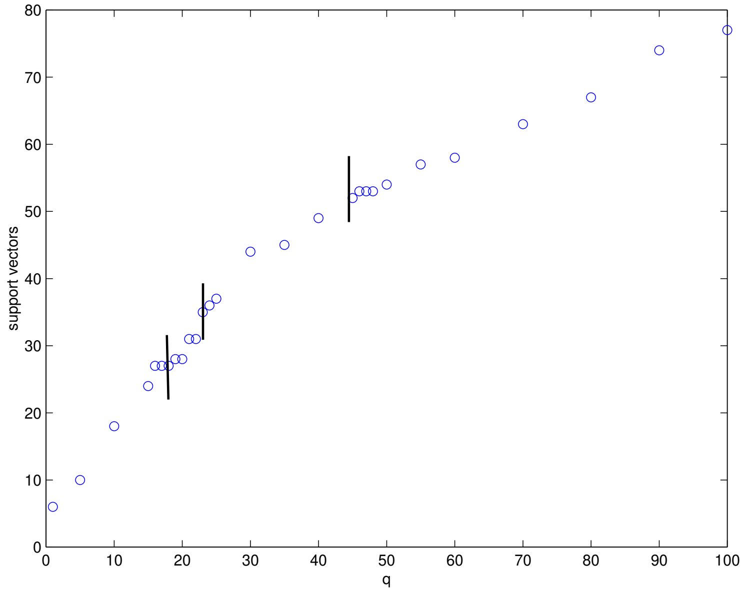

Figure 2: Number of SVs as a function of q for the data of Figure 1. Contour splitting points are denoted by vertical lines.

by the parameter C. From the constraints (3,9) it follows that

$$n_{bsv} < 1/C\,, \tag{17}$$

where n_{bsv} is the number of BSVs. Thus $1/(NC)$ is an upper bound on the fraction of BSVs, and it is more natural to work with the parameter

$$p = \frac{1}{NC}\,. \tag{18}$$

Asymptotically (for large N), the fraction of outliers tends to p , as noted in Schölkopf et al. (2000).

When distinct clusters are present, but some outliers (e.g. due to noise) prevent contour separation, it is very useful to employ BSVs. This is demonstrated in Figure 3a: without BSVs contour separation does not occur for the two outer rings for any value of q. When some BSVs are present, the clusters are separated easily (Figure 3b). The difference between data that are contour-separable without BSVs and data that require use of BSVs is illustrated schematically in Figure 4. A small overlap between the two probability distributions that generate the data is enough to prevent separation if there are no BSVs.

In the spirit of the examples displayed in Figures 1 and 3 we propose to use SVC iteratively: Starting with a low value of q where there is a single cluster, and increasing it, to observe the formation of an increasing number of clusters, as the Gaussian kernel describes the data with increasing precision. If, however, the number of SVs is excessive,

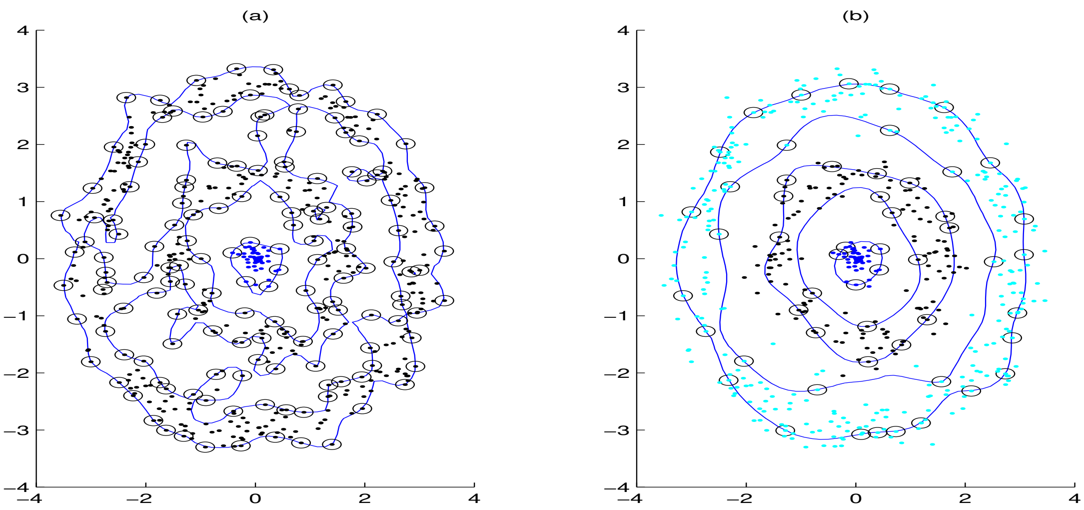

Figure 3: Clustering with and without BSVs. The inner cluster is composed of 50 points generated from a Gaussian distribution. The two concentric rings contain 150/300 points, generated from a uniform angular distribution and radial Gaussian distribution. (a) The rings cannot be distinguished when $C = 1$. Shown here is $q = 3.5$, the lowest q value that leads to separation of the inner cluster. (b) Outliers allow easy clustering. The parameters are $p = 0.3$ and $q = 1.0$.

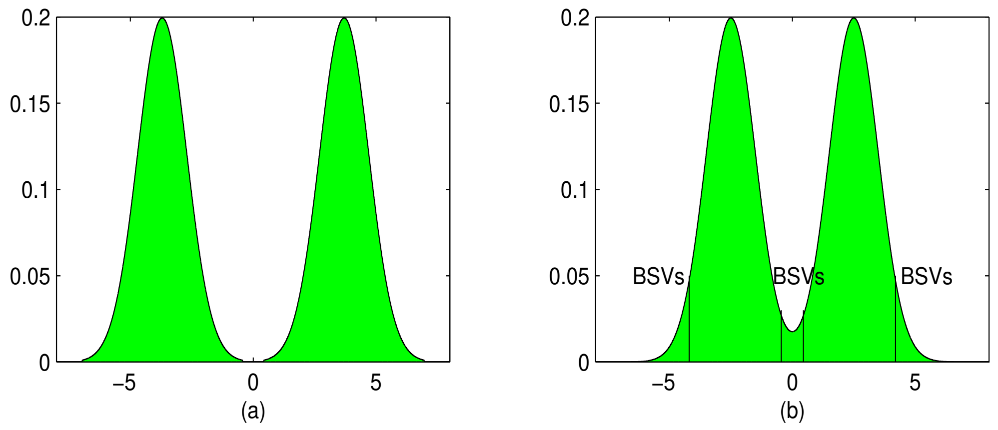

Figure 4: Clusters with overlapping density functions require the introduction of BSVs.

i.e. a large fraction of the data turns into SVs (Figure 3a), or a number of singleton clusters form, one should increase p to allow these points to turn into outliers, thus facilitating contour separation (Figure 3b). As p is increased not only does the number of BSVs increase, but their influence on the shape of the cluster contour decreases, as shown in Ben-Hur et al. (2000). The number of support vectors depends on both q and p. For fixed

q, as p is increased, the number of SVs decreases since some of them turn into BSVs and the contours become smoother (see Figure 3).

4. Strongly Overlapping Clusters

Our algorithm may also be useful in cases where clusters strongly overlap, however a different interpretation of the results is required. We propose to use in such a case a high BSV regime, and reinterpret the sphere in feature space as representing cluster cores, rather than the envelope of all data.

Note that equation (15) for the reflection of the sphere in data space can be expressed as

$$\{\mathbf{x} \mid \sum_i \beta_i K(\mathbf{x}_i, \mathbf{x}) = \rho\}, \tag{19}$$

where ρ is determined by the value of this sum on the support vectors. The set of points enclosed by the contour is:

$$\{\mathbf{x} \mid \sum_i \beta_i K(\mathbf{x}_i, \mathbf{x}) > \rho\}. \tag{20}$$

In the extreme case when almost all data points are BSVs ($p \to 1$), the sum in this expression,

$$P_{svc} = \sum_i \beta_i K(\mathbf{x}_i, \mathbf{x}) \tag{21}$$

is approximately equal to

$$P_w = \frac{1}{N} \sum_i K(\mathbf{x}_i, \mathbf{x}). \tag{22}$$

This last expression is recognized as a Parzen window estimate of the density function (up to a normalization factor, if the kernel is not appropriately normalized), see Duda et al. (2001). In this high BSV regime, we expect the contour in data space to enclose a small number of points which lie near the maximum of the Parzen-estimated density. In other words, the contour specifies the *core* of the probability distribution. This is schematically represented in Figure 5.

In this regime our algorithm is closely related to the scale-space algorithm proposed by Roberts (1997). He defines cluster centers as maxima of the Parzen window estimator $P_w(\mathbf{x})$. The Gaussian kernel plays an important role in his analysis: it is the only kernel for which the number of maxima (hence the number of clusters) is a monotonically non-decreasing function of q. This is the counterpart of contour splitting in SVC. As an example we study the crab data set of Ripley (1996) in Figure 6. We plot the topographic maps of P_w and P_{svc} in the high BSV regime. The two maps are very similar. In Figure 6a we present the SVC clustering assignment. Figure 6b shows the original classification superimposed on the topographic map of P_w. In the scale space clustering approach it is difficult to identify the bottom right cluster, since there is only a small region that attracts points to this local maximum. We propose to first identify the contours that form cluster cores, the dark contours in Figure 6a, and then associate points (including BSVs) to clusters according to their distances from cluster cores.

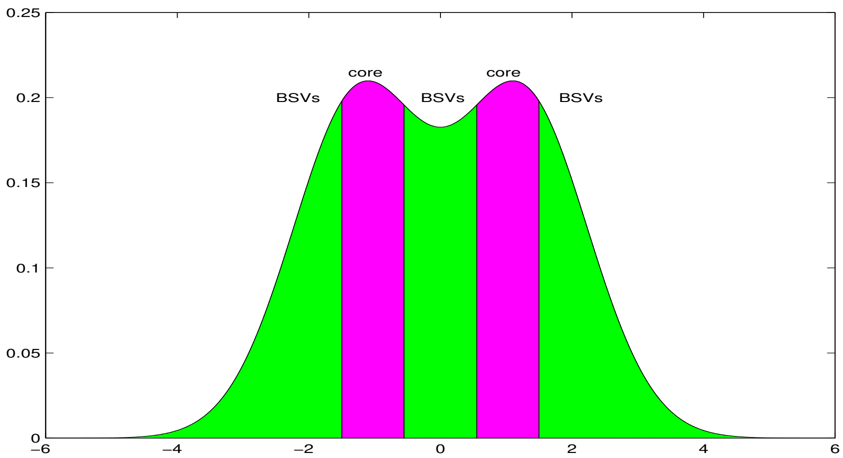

Figure 5: In the case of significant overlap between clusters the algorithm identifies clusters according to dense cores, or maxima of the underlying probability distribution.

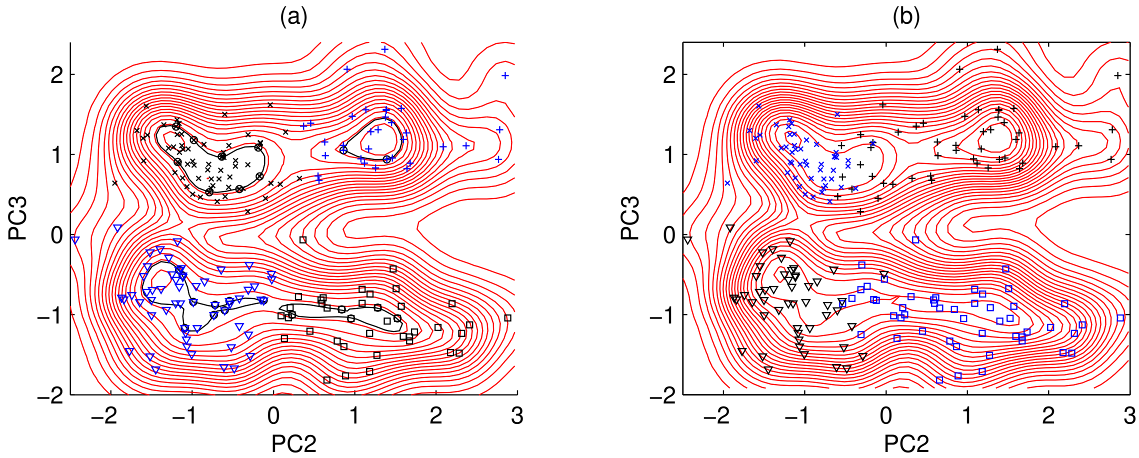

Figure 6: Ripley's crab data displayed on a plot of their 2nd and 3rd principal components: (a) Topographic map of $P_{svc}(\mathbf{x})$ and SVC cluster assignments. Cluster core boundaries are denoted by bold contours; parameters were $q = 4.8, p = 0.7$. (b) The Parzen window topographic map $P_w(\mathbf{x})$ for the same q value, and the data represented by the original classification given by Ripley (1996).

The computational advantage of SVC over Roberts' method is that, instead of solving a problem with many local maxima, we identify core boundaries by an SV method with a global optimal solution. The conceptual advantage of our method is that we define a region, rather than just a peak, as the core of the cluster.

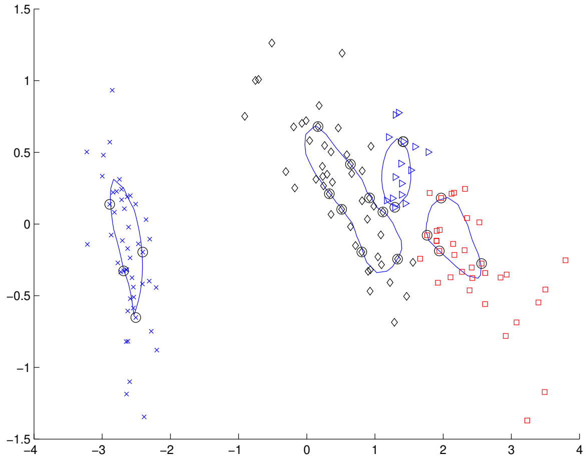

Figure 7: Cluster boundaries of the iris data set analyzed in a two-dimensional space spanned by the first two principal components. Parameters used are $q = 6.0$ $p = 0.6$.

4.1 The Iris Data

We ran SVC on the iris data set of Fisher (1936), which is a standard benchmark in the pattern recognition literature, and can be obtained from Blake and Merz (1998). The data set contains 150 instances each composed of four measurements of an iris flower. There are three types of flowers, represented by 50 instances each. Clustering of this data in the space of its first two principal components is depicted in Figure 7 (data was centered prior to extraction of principal components). One of the clusters is linearly separable from the other two by a clear gap in the probability distribution. The remaining two clusters have significant overlap, and were separated at $q = 6$ $p = 0.6$. However, at these values of the parameters, the third cluster split into two (see Figure 7). When these two clusters are considered together, the result is 2 misclassifications. Adding the third principal component we obtained the three clusters at $q = 7.0$ $p = 0.70$, with four misclassifications. With the fourth principal component the number of misclassifications increased to 14 (using $q = 9.0$ $p = 0.75$). In addition, the number of support vectors increased with increasing dimensionality (18 in 2 dimensions, 23 in 3 dimensions and 34 in 4 dimensions). The improved performance in 2 or 3 dimensions can be attributed to the noise reduction effect of PCA. Our results compare favorably with other non-parametric clustering algorithms: the information theoretic approach of Tishby and Slonim (2001) leads to 5 misclassifications and the SPC algorithm of Blatt et al. (1997), when applied to the dataset in the original data-space, has 15 misclassifications. For high dimensional datasets, e.g. the Isolet dataset which has 617 dimensions, the problem was obtaining a support vector description: the number of support vectors jumped from very few (one cluster) to all data points being support vectors (every point in a separate cluster). Using PCA to reduce the dimensionality produced data that clustered well.

4.2 Varying q and p

We propose to use SVC as a "divisive" clustering algorithm, see Jain and Dubes (1988): starting from a small value of q and increasing it. The initial value of q may be chosen as

$$q = \frac{1}{\max_{i,j} ||\mathbf{x}_i - \mathbf{x}_j||^2} \,. \tag{23}$$

At this scale all pairs of points produce a sizeable kernel value, resulting in a single cluster. At this value no outliers are needed, hence we choose $C = 1$.

As q is increased we expect to find bifurcations of clusters. Although this may look as hierarchical clustering, we have found counterexamples when using BSVs. Thus strict hierarchy is not guaranteed, unless the algorithm is applied separately to each cluster rather than to the whole dataset. We do not pursue this choice here, in order to show how the cluster structure is unraveled as q is increased. Starting out with $p = 1/N$, or $C = 1$, we do not allow for any outliers. If, as q is being increased, clusters of single or few points break off, or cluster boundaries become very rough (as in Figure 3a), p should be increased in order to investigate what happens when BSVs are allowed. In general, a good criterion seems to be the number of SVs: a low number guarantees smooth boundaries. As q increases this number increases, as in Figure 2. If the number of SVs is excessive, p should be increased, whereby many SVs may be turned into BSVs, and smooth cluster (or core) boundaries emerge, as in Figure 3b. In other words, we propose to systematically increase q and p along a direction that guarantees a minimal number of SVs. A second criterion for good clustering solutions is the stability of cluster assignments over some range of the two parameters.

An important issue in the divisive approach is the decision when to stop dividing the clusters. Many approaches to this problem exist, such as Milligan and Cooper (1985), Ben-Hur et al. (2002) (and references therein). However, we believe that in our SV setting it is natural to use the number of support vectors as an indication of a meaningful solution, as described above. Hence we should stop SVC when the fraction of SVs exceeds some threshold.

5. Complexity

The quadratic programming problem of equation (2) can be solved by the SMO algorithm of Platt (1999) which was proposed as an efficient tool for SVM training in the supervised case. Some minor modifications are required to adapt it to the unsupervised training problem addressed here, see Schölkopf et al. (2000). Benchmarks reported in Platt (1999) show that this algorithm converges after approximately $O(N^2)$ kernel evaluations. The complexity of the labeling part of the algorithm is $O((N - n_{bsv})^2 n_{sv} d)$, so that the overall complexity is $O(N^2 d)$ if the number of support vectors is $O(1)$. We use a heuristic to lower this estimate: we do not compute the whole adjacency matrix, but only adjacencies with support vectors. This gave the same results on the data sets we have tried, and lowers the complexity to $O((N - n_{bsv}) n_{sv}^2)$. 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. This makes SVC useful even for very large datasets.

6. Discussion

We have proposed a novel clustering method, SVC, based on the SVM formalism. Our method has no explicit bias of either the number, or the shape of clusters. It has two parameters, allowing it to obtain various clustering solutions. The parameter q of the Gaussian kernel determines the scale at which the data is probed, and as it is increased clusters begin to split. The other parameter, p, is the soft margin constant that controls the number of outliers. This parameter enables analyzing noisy data points and separating between overlapping clusters. This is in contrast with most clustering algorithms found in the literature, that have no mechanism for dealing with noise or outliers. However we note that for clustering instances with strongly overlapping clusters SVC can delineate only relatively small cluster cores. An alternative for overlapping clusters is to use a support vector description for each cluster. Preliminary results in this direction are found in Ben-Hur et al. (2000).

A unique advantage of our algorithm is that it can generate cluster boundaries of arbitrary shape, whereas other algorithms that use a geometric representation are most often limited to hyper-ellipsoids, see Jain and Dubes (1988). In this respect SVC is reminiscent of the method of Lipson and Siegelmann (2000) where high order neurons define a high dimensional feature-space. Our algorithm has a distinct advantage over the latter: being based on a kernel method it avoids explicit calculations in the high-dimensional feature space, and hence is more efficient.

In the high p regime SVC becomes similar to the scale-space approach that probes the cluster structure using a Gaussian Parzen window estimate of the probability density, where cluster centers are defined by the local maxima of the density. Our method has the computational advantage of relying on the SVM quadratic optimization that has one global solution.

References

A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. in Pacific Symposium on Biocomputing, 2002.

A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik. A support vector clustering method. in International Conference on Pattern Recognition, 2000.

A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik. A support vector clustering method. in Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference, Todd K. Leen, Thomas G. Dietterich and Volker Tresp eds., 2001.

C.L. Blake and C.J. Merz. Uci repository of machine learning databases, 1998.

Marcelo Blatt, Shai Wiseman, and Eytan Domany. Data clustering using a model granular magnet. *Neural Computation*, 9(8):1805–1842, 1997.

R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* John Wiley & Sons, New York, 2001.

R.A. Fisher. The use of multiple measurments in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

R. Fletcher. *Practical Methods of Optimization.* Wiley-Interscience, Chichester, 1987.

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

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

H. Lipson and H.T. Siegelmann. Clustering irregular shapes using high-order neurons. *Neural Computation*, 12:2331–2353, 2000.

J. MacQueen. Some methods for classification and analysis of multivariate observations. in Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, 1965.

G.W. Milligan and M.C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50:159–179, 1985.

J. Platt. Fast training of support vector machines using sequential minimal optimization. in Advances in Kernel Methods — Support Vector Learning, B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, 1999.

B.D. Ripley. *Pattern recognition and neural networks.* Cambridge University Press, Cambridge, 1996.

S.J. Roberts. Non-parametric unsupervised cluster analysis. *Pattern Recognition*, 30(2): 261–272, 1997.

B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. in Advances in Neural Information Processing Systems 12: Proceedings of the 1999 Conference, Sara A. Solla, Todd K. Leen and Klaus-Robert Muller eds., 2000.

Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, , Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

R. Shamir and R. Sharan. Algorithmic approaches to clustering gene expression data. in T. Jiang, T. Smith, Y. Xu, and M.Q. Zhang, editors, Current Topics in Computational Biology, 2000.

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

N. Tishby and N. Slonim. Data clustering by Markovian relaxation and the information bottleneck method. in Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference, Todd K. Leen, Thomas G. Dietterich and Volker Tresp eds., 2001.

V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.