Locally Constrained Support Vector Clustering

Dragomir Yankov, Eamonn Keogh, Kin Fai Kan
Computer Science & Engineering Department
University of California, Riverside, USA
{dyankov, eamonn, kkan}@cs.ucr.edu

Abstract

Support vector clustering transforms the data into a high dimensional feature space, where a decision function is computed. In the original space, the function outlines the boundaries of higher density regions, naturally splitting the data into individual clusters. The method, however, though theoretically sound, has certain drawbacks which make it not so appealing to the practitioner. Namely, it is unstable in the presence of outliers and it is hard to control the number of clusters that it identifies. Parametrizing the algorithm incorrectly in noisy settings, can either disguise some objectively present clusters in the data, or can identify a large number of small and nonintuitive clusters.

Here, we explore the properties of the data in small regions building a mixture of factor analyzers. The obtained information is used to regularize the complexity of the outlined cluster boundaries, by assigning suitable weighting to each example. The approach is demonstrated to be less susceptible to noise and to outline better interpretable clusters than support vector clustering alone.

1 Introduction

One-class support vector machine (SVM) is an efficient approach for estimating the density of a population [15, 17]. It works by applying a transformation $\Phi : X \rightarrow \Phi(X)$ from the input space to a high dimensional feature space, such that points within denser neighborhoods are projected further from the origin of the coordinate system. The support vectors in the feature space are then used to outline closed contours around the dense regions in the input space, defining a binary decision function which is positive inside the contours and negative elsewhere. The method has been demonstrated to be applicable for tasks, such as novelty and fault detection, context change detection, learning in image retrieval, etc.

One can easily extend one-class classification to a clustering scheme, by labeling each closed contour as a different cluster. Elements, not enclosed by any contour, correspond to regions that are estimated to have lower density support in the high dimensional feature space. Such elements can be assigned the label of their closest contour in the original space. This extension, called support vector clustering (SVC), was initially proposed in [3].

Despite its theoretical soundness the SVC method has remained relatively unpopular among the practitioners. There are several specific characteristics of SVC that diminish its appeal. For instance, the map Φ requires a parametrized kernel to be provided as an input from the user. The radial basis function $k(x_i, x_j) = e^{-\gamma\|x_i - x_j\|^2}$ has been recognized as a preferred kernel function because of its ability to form closed contours [3, 18]. This means that the user needs to provide a suitable kernel width γ. However, small values of γ (i.e. large kernel widths) may disguise or merge some of the clusters, while very large γ may create multiple closed contours which outline some rather nonintuitive clusters. The effect of multiple emerging clusters is especially strong in the presence of noise. This becomes an issue, in many practical application where the examples lie near the surface of a lower dimensional nonlinear manifold. For example, such noisy manifolds may be defined by a sample of facial images [13, 14, 19], or by the walking motions of a human [10]. Though a soft margin can be introduced to alleviate the impact of the outliers, there is again the issue of how to specify the correct parameter ν, that controls the tradeoff between the generalization performance of the learner and its tolerance to the noisy examples.

To improve the performance of SVC in the case of Gaussian distributed noise and to obtain better control over the number of detected clusters, we explore the density variability of the data in very small regions. For the purpose, a Mixture of Factor Analyzers (MFA) [9] is used. The mixture model, when learned with a large number of analyzers, implicitly detects points that deviate from the main trajectory of the data. The information about those locally deviating points is used to determine the soft margin tradeoff between the outliers and the accuracy of the one-class SVM learner, as well as, to regularize the complexity of the induced decision boundary. The regularization results

in smoother contours, which are shrunk towards the dense regions in the data, rather than trying to accommodate all outliers. The subsequent clustering often allows for easier interpretation too. Because of the local dimensionality reduction performed by MFA and the nonlinear feature map Φ, the "locally constrained" SVC method is further demonstrated to correctly identify the topological structure of the data, when the clusters reside on a lower dimensional nonlinear manifold.

2 Related Work

A number of clustering algorithms have been demonstrated to be particularly suitable for learning of non-convex formations, e.g. spectral clustering [12], spectral graph partitioning [8], or kernel K-means [16]. A close relation between all of these approaches has been pointed out before [4]. We focus on one of these algorithms - spectral clustering. Interestingly, the algorithm shares a lot of commonalities with SVC. They both start by computing a Gaussian kernel matrix, emulating the high dimensional nonlinear feature map Φ. From here on, however, spectral clustering performs an eigen decomposition of the data in the feature space. The projected examples are then clustered, again in the feature space, using K-means clustering. Instead, SVC computes the optimal plane that separates the projected data from the origin in the feature space. In this way a simpler problem is solved by only isolating the higher density regions. This comes at the price of not knowing the actual clusters in the data, so a subsequent labeling and assignment step is carried out by SVC.

A different set of unsupervised learning approaches try to infer the nonlinear structure of the data by considering small regions around each example. Some popular methods following this paradigm are, for example, the Laplacian eigenmaps [2] and Isomap [19]. The general idea behind these algorithms is to compute a neighborhood graph G, where each example x_i is connected only to examples in its close proximity. The graph is then augmented to a full affinity matrix, by propagating the neighboring distances, e.g. by solving an all pairs shortest path problem (Isomap) or by applying a Laplacian operator (Laplacian eigenmaps). Both methods proceed by computing an eigen decomposition and projecting the data using a small subset of the eigenvectors. As they preserve the convexity of the data, the algorithms can easily be extended for clustering by using a partitioning scheme as K-means or Expectation Maximization (EM) for a mixture of Gaussians. While local reduction methods have been demonstrated to be unstable in the presence of noise [1], they remain to be the preferred tool for unsupervised learning from nonlinear manifolds.

In the proposed approach we combine the best features that can be obtained from global methods, such as SVC and local approximations as the ones discussed above. The underlying idea is that a global view of the data can be inferred by looking at the overall density distribution. The density estimate alone, however, provides for a very coarse reconstruction of the underlying sample space. Local methods, on the other hand, can smoothen this estimate by looking at the data statistics in some small regions. This is especially important if density fluctuations are observed in the data and yet an obvious clustering is available. In this sense, the proposed method is closest in spirit to the manifold reconstruction method proposed by Roweis et al. [14]. They use a mixture of factor analyzers to infer the local structure of the underlying manifold, but then a global constraint is imposed, so that all local models are aligned to follow a consistent trajectory.

3 Support Vector Density Estimation

3.1 One-Class Classification

Let us have a set of n independent and identically distributed observations: $X = \{x_i\}_{i=1}^{n}$. The problem addressed by one-class classification is to find a minimal region R, which encloses the data (Figure 1). Assuming that the data are generated from the same distribution p, an additional to the minimization of R is the requirement that future test examples generated by p should also fall with high probability within R. Therefore, apart of being minimal, R should also generalize well on unseen data, which implies that it should have a non-complex boundary.

Following similar reasoning as in support vector classification, rather than exploring the nonlinear boundary in the original space, one could describe it as a hyper plane in the high dimensional feature space defined by $\Phi(X)$. All examples, which in the original space are enclosed within R, are going to be projected in the same half-space with respect to the hyper plane. If $\mathbf{w} \cdot \Phi(x) = b$ is the equation of the plane, this is equivalent to the requirement that for all examples x_i, the inequality $\mathbf{w} \cdot \Phi(x_i) \geq b$ should hold. The two parameters that define the plane uniquely, $\mathbf{w}$ and b, are its normal vector and its displacement from the origin respectively. Finally, the plane that corresponds to the smoothest boundary in the original space is the one with smallest norm of the normal vector $\mathbf{w}$ [16]. The equation of this plane is given by the solution of the optimization problem:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & \mathbf{w} \cdot \Phi(x_i) \geq b,\ i = 1..n \end{aligned} \tag{1}$$

It may be useful to restrict R to enclose only a subregion of X that has higher support for the probability density function. This will be the case, for example, if we are not interested in the noisy points on the periphery of the distribution (see Figure 1). In the feature space, the points that

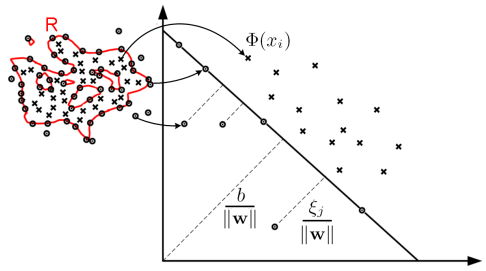

Figure 1: One-class SVMs detect a region R in the data with higher density support. Points inside the region are projected in the same half-space defined by the separation hyper plane $(\mathbf{w}, b)$.

fall outside of R will satisfy $\mathbf{w} \cdot \Phi(x_i) < b$. To account for such points the constraints for them in (1) should be changed to $\mathbf{w} \cdot \Phi(x_i) \geq b - \xi_i$, where we have additionally introduced the *slack variables* $\xi_i \geq 0$. The regularization term that guarantees the smoothness of the boundary also changes, yielding the new formulation:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & q(\mathbf{w}, b, \xi) = \tfrac{1}{2}\|\mathbf{w}\|^2 + \tfrac{1}{n\nu}\textstyle\sum_{i=1}^{n} \xi_i - b \quad (2) \\ \text{subject to} \quad & \mathbf{w} \cdot \Phi(x_i) \geq b - \xi_i,\ \xi_i \geq 0,\ i = 1..n \end{aligned}$$

Formulations (1) and (2) produce the so called *hard* and *soft margin* decision planes respectively. The penalty parameter ν in (2) controls the tradeoff between the allowed slack for some of the examples and the complexity of the region boundary. It takes values in the interval $(0, 1]$ with $\nu \to 1$ allowing for a lot of examples to lie outside the region R, and $\nu \to 0$ penalizing significantly the slack variables, converting the problem effectively into a hard margin decision problem. The latter case leads to a very tight and complex boundary for the density region R.

Minimizing the quadratic function $q(\mathbf{w}, b, \xi)$ in problem (2) is hard, because of the available constraints. Instead, if we write all constraints in the form $q_i(\mathbf{w}, b, \xi) \leq 0$, the solution is obtained by minimizing the Lagrangian $L(\mathbf{w}, b, \xi, \alpha) = q(\mathbf{w}, b, \xi) + \sum_i \alpha_i q_i(\mathbf{w}, b, \xi)$. To minimize L, one sets the derivatives of L with respect of $\mathbf{w}$, b and ξ to zero, which allows for expressing them as a function solely of the introduced Lagrangian multipliers α_i ($\alpha_i \geq 0$, $\sum_i \alpha_i = 1$) and the data in the feature space $\Phi(x_i)$. Substituting the values back in the Lagrangian, we obtain the dual optimization problem of problem (2):

$$\begin{aligned} \min_{\alpha} \quad & \tfrac{1}{2}\textstyle\sum_{ij} \alpha_i \alpha_j \Phi(x_i) \cdot \Phi(x_j) \quad (3) \\ \text{subject to} \quad & \textstyle\sum_{i=1}^{n} \alpha_i = 1,\ 0 \leq \alpha_i \leq \frac{1}{n\nu} \end{aligned}$$

The class of feature mappings $\Phi(X)$ that linearly separate the data from the origin is not available in parametric form, yet it is selected so that the dot products in the feature space correspond to a computable kernel function in the input space, i.e. $k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$. In SVC the Gaussian kernel $k(x_i, x_j) = e^{-\gamma\|x_i - x_j\|^2}$ is used, as it defines smooth closed contours [3, 18]. All multipliers $\alpha_i > 0$ in the solution of (3) correspond to the support vectors, i.e. the examples which in the feature space lie on the separating hyper-plane (see Figure 1). For the rest of the points x_i the corresponding α_i is equal to zero. To test on which side of the hyper plane such examples are projected, one needs to substitute them in the equation of the plane as defined by the computed support vectors:

$$f(x) = sgn[\sum_{x_i \in SVs} \alpha_i k(x_i, x) - b] \quad (4)$$

Positive $f(x)$ implies that x falls within the dense subspace R, whereas negative values of the decision function imply a sparsely populated region. Observations $x_i \in X$ for which $f(x_i) < 0$ are called *bounded support vectors*. The value of the displacement b can be computed using the fact that any support vector x_s lies on the separation plane, and thus it satisfies the equality $\mathbf{w} \cdot \Phi(x_s) = b$, which can also be expressed in terms of the kernel function as $\sum_{x_i \in SVs} \alpha_i k(x_i, x_s) = b$.

The formalization defined so far is not the only way for computing high dimensional density support. For instance, instead of looking for the optimal separation plane, Ben-Hur et al. [3] study the class of spheres in the feature space that enclose the projected examples. They derive an alternative formulation of problem (3), which instead minimizes the volume of the enclosing hyper sphere. An equivalence of the two formulations has been demonstrated in [16]. In the current work, the density estimation step is carried out as in the original one-class SVM formulation.

3.2 Support Vector Clustering

The one-class density estimation method can easily be extended to a clustering scheme by computing a matrix A for the data, where $A_{ij} = 1$ if x_i and x_j are enclosed within the same contour and 0 otherwise. Whether x_i and x_j lie within the same contour can be determined by computing the SVM decision function (4) for all points on the line that connects them. In the original SVC formulation (and also in our implementation) 20 regularly spaced points between x_i and x_j are tested. An always positive decision function guarantees that x_i and x_j are part of the same dense region. The opposite, however, is not necessarily true. For some points, on the line between two examples, f may be negative, but the examples may still be within the same contour. This is often the case if the contours are too complex.

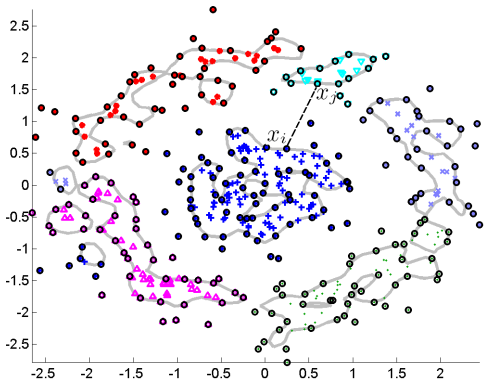

Figure 2: One-class SVMs can be extended to a clustering scheme, by assigning the same label to all points enclosed within the same contour. For example, x_i and x_j are within the same contour if for any point x on the line between them the decision function $f(x)$ is non-negative.

Therefore, one needs to detect the connected components in the graph induced by A. This determines the number of clusters in the data as well as the labels for each example that is enclosed by a contour. Finally, the bounded support vectors (i.e. the examples outside the contours) are assigned to their closest cluster (see Figure 2).

While precise parametrization is not so essential when only density estimation is required, it becomes of crucial importance in the case of clustering. Consider, for example, Figure 2. Selecting a large kernel width (i.e. small γ) would disguise the fact that there is large fluctuation between the density of the inner and the outer circles. Large values of γ or too small tradeoff terms ν, on the other hand, can produce decision boundary of a very high capacity, which leads to multiple tight contours in the original space. Apart of obtaining too many small and nondescriptive clusters, the complex decision function impedes the proper labeling even of elements that are within the same contour. For some examples x_p all lines connecting them to other examples x_q within the same contour, would pass through regions where the decision function has negative value. Such examples will be assumed to belong to a different cluster.

The lack of control over the number of clusters produced by different parametrizations is a significant drawback of the scheme. A common requirement in clustering is that the users provide the number of clusters that they want to be detected in their data. Such a requirement is easily handled by partition clustering (e.g. K-means), agglomerative clustering and even kernel based algorithms as spectral clustering. Unfortunately there is no clear unsupervised strategy of how such user imposed constraint can be incorporated in SVC. One reasonable way to emulate such behavior, would be to start exploring kernels with monotonically decreasing widths until at least as many clusters as the users require emerge from the data. Such iterative approach is followed for example in [11]. As will be shown in the experimental evaluation, this strategy, though pretty robust in the case of well separated and dense clusters, can cause the occurrence of some rather uninformative formations when the clusters are sparse and noise is present in the data.

Next, we introduce a modification of the SVC approach, which improves on its stability in the presence of noise. The method is further demonstrated to be less sensitive to slight changes in the parametrization.

4 Locally Constrained SVC

The intuition followed in the current work is that both global density estimation methods as SVC, and local reconstruction methods as Isomap [19] or LLE [13] introduce information about the data, which is somewhat complementary. For example, support vector clustering provides some very important information about the overall structure of the data. Namely, an estimate of its density. A local method can complement this with additional region boundary smoothing and can evaluate locally which points are likely to deviate from the unknown distribution that has generated the data. The method that we utilize here to obtain such local statistics is based on the Mixture of Factor Analyzers framework introduced by Ghahramani et al. in [9]. We term the algorithm derived in this section Locally constraint Support Vector Clustering (LSVC).

4.1 Mixture of Factor Analyzers

Factor analysis (FA) is a technique for linearly projecting the data $X \subset R^D$ into a lower dimensional space R^d [7]. Ghahramani et al. [9] derive an EM procedure for learning the projecting dimensions $\mathbf{z}$. They make the simplifying assumption that the dimensions $\mathbf{z}$ are normally distributed with zero means and variance one, i.e. $\mathbf{z} \in \mathcal{N}(0, I)$ (I here marks the identity matrix). Furthermore, each example is allowed to have some residual noise $\mathbf{u}$, which is also assumed to be normally distributed with covariance Ψ, i.e. $\mathbf{u} \in \mathcal{N}(0, \Psi)$. The following relation is now enforced: $\mathbf{x} = \Lambda\mathbf{z} + \mathbf{u}$, where Λ is the so called *factor loading matrix*, and the noise covariance matrix Ψ is required to be diagonal. The *common factors* $\mathbf{z}$ are used as latent variables to iteratively obtain an improved likelihood estimate for the observed data $\mathbf{x}$ (E-step of the algorithm), recomputing on each iterations more optimal values for the matrices Ψ and Λ (M-step of the algorithm).

Ghahramani et al. [9] also suggest that one could have a mixture of factor analyzers, rather than a single one, where every component in the mixture can have different mean μ_j and loading matrix Λ_j. The noise term in the mixture is preserved the same across all factor analyzers, i.e. $\mathbf{z}_j \in \mathcal{N}(\mu_j, \Psi)$. The goal now becomes to find a maximum likelihood estimate for the observed data $\mathbf{x}$, using the latent

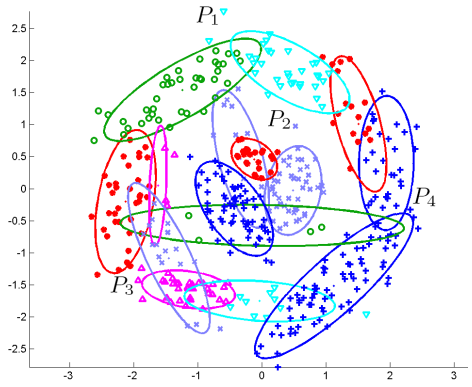

Figure 3: The topology of the data is closely approximated with a mixture of 20 analyzers. The ellipses outline two standard deviations from the center of the analyzers. The mixture can be used to detect "local" outliers (e.g. P_2) that bridge the existing clusters.

variables $\mathbf{z}_j$, and the probability that it has been projected using the j-th factor analyzer (E-step of the mixture model). On every iteration the MFA algorithm, apart of computing some more optimal estimates of the matrices Ψ and Λ_j, also improves on the estimate for the mean of the analyzers μ_j too (M-step of the mixture model).

Figure 3 illustrates the MFA algorithm when applied with twenty components. Apart of clustering the data, MFA also estimates the optimal lower dimensional representation for the examples in each cluster. This is an essential characteristic when the data follow the structure of a lower dimensional manifold embedded in the original space R^D. The locally constrained SVC method suggested here exploits this property.

4.2 Regularizing the One-Class SVMs

In the proposed approach we are going to use the fact that MFA can single out the majority of the outliers, which fall outside the main trajectory followed by the data. In Figure 3 the ellipses outline a two standard deviations region around the mean of the corresponding local clusters. Points, such as P_1 and P_2, that are too distant from their cluster centers, are indeed among the noisy points bridging the two global concentric clusters. Cleaning the data set from these points can significantly improve the performance of the SVC method. Note also, that using only the MFA method for reconstructing the underlying distribution will not provide a good enough solution either. Applied as a local method, similarly to Isomap and LLE, MFA can be instable because of the noise [1]. For instance, the two analyzers that bridge the two clusters on Figure 3 will impede the proper identification of the present formations. This comes to illustrate the importance of having an additional input from the global density method too.

Before we show how the information obtained through MFA can improve the one-class SVMs, it would be useful to understand how the outliers impact the detected contours. In the soft margin formulation (2), every example is allowed to cross the decision boundary with a penalty controlled by the slack variables ξ. This makes the decision function less complex, at the price of some misclassified examples x_i, which in this case means that the function underestimates the density around these examples. Misclassification of all such x_i is penalized proportionally to their distance to the separation plane (ξ_i), but with the same weighting factor $\frac{1}{n\nu}$. Assuming that there is an additional, possibly uncertain, knowledge about which examples are actually outliers, the procedure might instead be changed to use different weighting factors. The idea is similar to the weighted SVM classification, that has been demonstrated to be suitable in the case of imbalanced classes [6], with the difference being that the weights now should be determined based on the confidence that a certain example is an outlier.

A confidence estimate of the importance of each example can be obtained by measuring the example's deviation from the mean of the factor analyzer that it belongs to. If $\mathbf{z}_j = (z_1^j, z_2^j, \ldots, z_{r_j}^j)'$ are the projections of the examples that are assigned to the j-th mixture component, then the deviation of each example projection z_i^j can be expressed through the Mahalanobis distance:

$$\mathbf{d}_j = [(\mathbf{z}_j - \mu_j)' C_j (\mathbf{z}_j - \mu_j)]^{1/2} \tag{5}$$

In the above, the covariance of the j-th factor analyzer is estimated as $C_j = \Lambda_j \Lambda_j' + \Psi$ (see [9]). Now we adjust the penalty for misclassifying examples that are believed to be outliers (i.e. examples with large distance d_{ij} to their corresponding center μ_j) to be small, so that the decision function is not so influenced by them. This will smooth the separation boundary inferred by function (4), and hence will decrease the chance of having multiple small contours around sparser neighborhoods. To achieve that, each individual penalty term is modified to be inversely proportional to its Mahalanobis distance d_i. Now (2) is written as follows:

$$\begin{aligned} \min_{\mathbf{w},\mathbf{b},\xi} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 + \tfrac{1}{n\nu}\textstyle\sum_{i=1}^{n} \tfrac{1}{d_i}\xi_i - b \\ \text{subject to} \quad & \mathbf{w} \cdot \Phi(x_i) \geq b - \xi_i,\ \xi_i \geq 0,\ i = 1..n \end{aligned} \tag{6}$$

For brevity of notation in (6), we have omitted the indicator showing which factor analyzer the projection of an example x_i belongs to, yet it should be kept in mind that the distances d_i are computed based on the individual mixture components. Note, that the feature map Φ is applied on the original variables x_i rather than the projections z_i. The lat-

ter is done because the projecting dimensions for every factor analyzer are different. As density estimation in higher dimensional spaces has degrading effectiveness, it may still be necessary to perform a dimensionality reduction of the space X before solving the optimization problem (6). For that purpose, one could detect a global coordination for all factor analyzers [14], or just use a linear reduction as PCA as suggested by Ben-Hur et al. [3]. Here we use the second approach, which does not diminish the importance of MFA in the overall scheme, as the example weights have been computed based on the intrinsic dimensionality inferred by the method.

The Lagrangian now has the form:

$$L = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{n\nu}\sum_{i=1}^{n}\frac{1}{d_i}\xi_i - b - \sum_{i=1}^{n}\alpha_i(\Phi(x_i) - b + \xi_i) - \sum_{i=1}^{n}\beta_i\xi_i \qquad (7)$$

Taking the derivatives with respect to the primal variables $\mathbf{w}$, b, and ξ_i and substituting in (7) we obtain the dual optimization problem which we now try to maximize with respect to the dual variables α_i. This yields the constraint optimization problem:

$$\begin{aligned} \min_{\alpha} \quad & \tfrac{1}{2}\textstyle\sum_{ij}\alpha_i\alpha_j\Phi(z_i)\cdot\Phi(z_j) \\ \text{subject to} \quad & \textstyle\sum_{i=1}^{n}\alpha_i = 1,\ 0 \le \alpha_i \le \frac{1}{d_i n\nu} \end{aligned} \qquad (8)$$

In [16] the one-class SVM optimization problem is demonstrated to be solvable with a fast iterative technique called sequential minimal optimization (SMO). What makes the method applicable is the special form of the objective function and the linear equality constraints $\sum_{i=1}^{n}\alpha_i = 1$. Both, the function and the equality constraints in (8), are similar to the ones in problem (3), which means that we can perform the optimization using SMO again. Formulations (3) and (8) differ only the constraints imposed on α_i, which are now allowed to be upper-bounded by different values. That upper-bound is determined based on the confidence for the corresponding examples to be outliers.

It may be argued that the described process will also identify as noisy points that are not necessarily outliers. For instance, the points P_3 and P_4 in Figure 3. They are part of denser regions, yet they deviate from their component centers too. In this sense we say that the feedback obtained from MFA is uncertain, yet this will not necessarily have a detrimental effect, as the collaboration with the density estimation procedure again comes into play. The decision function evaluated for the denser region where P_3 resides will be positive for a large set of kernel widths, and the optimal slack variable for this point will most likely be zero, regardless of what constraint is imposed on its weight.

The number of mixture components that we use in the evaluation procedure is set to be larger than the number of clusters that we would like to be detected in the data. In general, we find it as a good practice to use at least several analyzers for each cluster that we want to detect. This ensures that if there are non-convex clusters present, each cluster may be covered with more than one component on average, which would better outline the cluster's topology. This may seem like very loose specification, yet we observe that even providing a relatively large number of components, the LSVC algorithm still correctly detects as bounded support vectors points that are indeed outliers. We could also specify the number of analyzers as a fraction of the total number of examples. In this mode MFA would roughly approximate methods, such as Isomap or LLE which use neighborhoods of certain size to reconstruct the underlying structure. For example, if we set the number of analyzers to be equal to $\frac{n}{10}$, then most components in the mixture will on average have ten elements and will resemble the neighborhoods constructed by the local methods.

Before we conclude this section, we note another interesting estimate that can be obtained through the MFA algorithm, namely, that of the tradeoff parameter ν. [15] demonstrates that the optimal ν to be specified in the one-class optimization problem (2) should be an upper bound on the fraction of outliers that are assumed to be present in the data. This fact by itself is not very helpful, as the number of outliers is unknown in advance. Using the factor analyzers, however, such an estimate can be obtained for example by counting the elements which deviate significantly from the mean of their mixture component. For the purpose, we compute the empirical standard deviation of the Mahalanobis distances d_{ij} within each analyzer. Then we set $\nu = \sum_j s_j/n$, where s_j is the number of examples that are more than two standard deviations away from the mean of the j-th analyzer.

5 Discussion

Using an example, we will elaborate on the effect that the introduced weighting scheme has on the detected contours. We run the two algorithms, SVC and the LSVC, on the synthetic "target data set" from Figure 4 (see Section 6 for details about its generation). The parameters used for both algorithms are $\gamma = 8$ and $\nu = 0.1$. Ten factor analyzers were used in the weight computing step for LSVC.

The black diamonds on the graphs represent bounded support vectors or support vectors which were found to form no connected components with any of the other examples (i.e. they form a one point cluster). As Figure 4 left

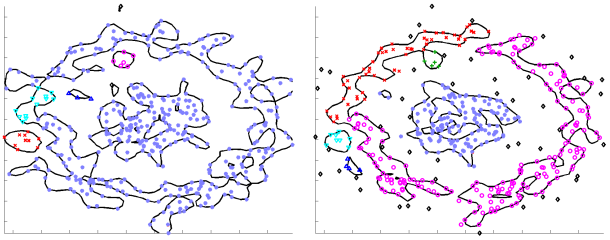

Figure 4: $\gamma = 8$ and $\nu = 0.1$. *Left*: SVC tries to accommodate all examples building complex contours and incorrectly bridging the two concentric clusters. *Right*: LSVC, the proposed here method, detects most outliers. The contours shrink towards the dense regions and the two main clusters are separated correctly.

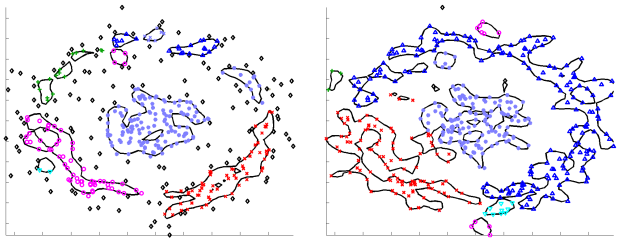

Figure 5: *Left*: SVC for $\gamma = 8$ and $\nu = 0.4$. Many outliers are now correctly identified, but the rest of the points are split into multiple uninformative clusters. *Right*: SVC for $\gamma = 9$ and $\nu = 0.1$. Increasing γ also cannot achieve the LSVC effect. The contours become very tight and complex and start splitting into multiple clusters.

shows, SVC tries to learn a decision boundary that accounts for almost all of the examples. This results in bridging the two concentric clusters present in the data. For the same parameters, LSVC (see Figure 5 right) forms contours that are shrunk towards the means of the data distribution. Multiple points, with lower density around them, are identified as bounded support vectors. Such points are identified as noise during the MFA step, and their weights in building the decision function have been decreased. The central circle is now detected as a separate cluster, while the outer circle has approximately as many clusters as in the SVC case.

It could be argued that we give an advantage to the LSVC algorithm by allowing the penalty to vary due to the different weights, while for SVC it is fixed with the constant ν. It is true, that if we relax the penalty for all examples (i.e. increase ν), some of the noisy points will be identified as bounded support vectors by SVC too. Yet, there is the problem of how exactly ν should be determined to improve the performance of SVC. In this case the value $\nu = 0.1$ was automatically computed using the previously described procedure of counting the deviating points for the ten factor analyzers. Furthermore, a suitable value for ν may not exist for the currently selected γ. For example, increasing ν twice produces almost identical results as $\nu = 0.1$. Increasing it four times leads to the graph on Figure 5 left.

SVC detects the internal circle as a separate cluster now, but the outer circle is split into multiple nonintuitive clusters. Another alternative to isolating the noisy points would be to keep ν unchanged and decrease the kernel width instead. However, there is again the issue of what kernel width would be more accurate. Furthermore, decreasing the width increases the complexity of the boundary, forming some very tight contours (see Figure 5 right) that at some point may also split into multiple clusters.

6 Experimental Evaluation

To demonstrate the performance of the proposed method we employ the following unsupervised procedure, which we run with both algorithms SVC and LSVC. For every data set we specify the number of clusters k that we would like the algorithm to detect. For all experiments the number of factor analyzers in LSVC is set to 10. The value of ν is determined as the fraction of outliers detected in the MFA step. The same value of ν is used in parameterizing SVC too. We vary $\log \gamma$ within the interval $[-16, 16]$ starting with -16 and incrementing it with step 1 at a time. This gradually increases γ (i.e. decreases the kernel width) and causes for more clusters to emerge. We stop the procedure when the number of clusters $\hat{k}$ detected by the algorithm surpasses k (i.e. $\hat{k} \geq k$). The procedure is suitable for comparing the robustness of the two algorithms, as the rate with which the clusters emerge when slowly decreasing the kernel width is highly correlated to the stability of the density estimation procedure in the presence of noise.

Though SVC and LSVC are primarily density estimation methods, rather than clustering algorithms for detection of fixed number of classes, we also check which would be the k clusters that the algorithms will return to the users. For the purpose, if $\hat{k}$ is larger than k, we start appending smaller clusters to the k largest clusters. The merging is done based on the minimal pairwise distance between the different clusters. Though not formal enough, and prone to certain errors, this merging step is suitable for detecting whether the clusters identified by the algorithms are well separated or there are dense regions that bridge them. The bounded support vectors are also assigned to their closest cluster.

6.1 Synthetic Data Sets

We first study the performance of SVC and LSVC on the synthetic data set used throughout this exposition. The data represents two concentric circles (see Figure 6), and is generated similarly to one of the data sets used by Ben-Hur et al. in [3]. The inner concentric circle contains 150 points from a Gaussian distribution. The outer circle is composed of 300 points from a radial Gaussian distribution and a uniform angular distribution.

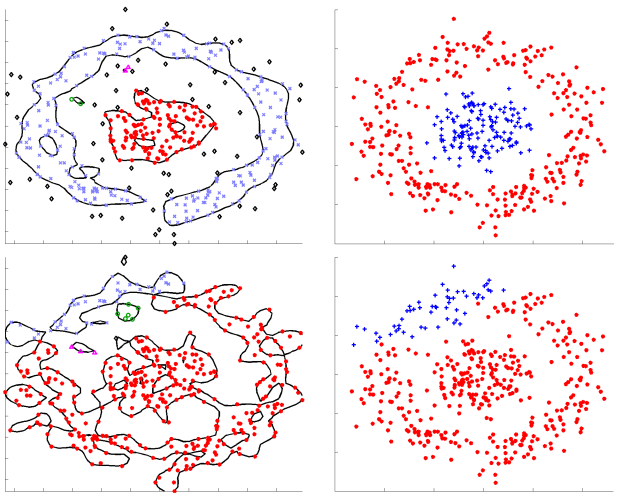

Figure 6: *Top*: the proposed LSVC algorithm; *left*: the contours and the clusters identified by the automatic procedure (the black diamonds indicate the bounded support vectors detected as noise); *right*: merging to obtain only two clusters. *Bottom*: the SVC algorithm; *left*: identified contours and clusters; *right*: merging to obtain only two clusters.

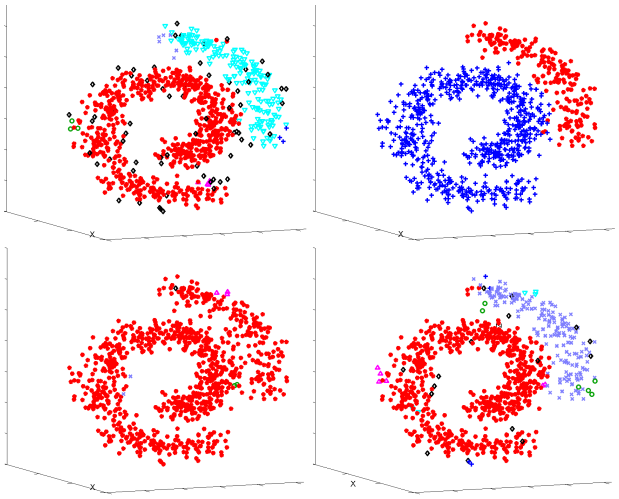

Figure 7: *Top*: the proposed LSVC algorithm; *left*: clusters identified by the automatic procedure; *right*: merging to obtain only two clusters. *Bottom*: the SVC algorithm; *left*: 5 small nonrepresentative clusters are identified with the automatic procedure; *right*: using supervision we detect parameters that lead to better clustering, which still fails to isolate the noise.

We set $k = 2$ and run the described automatic procedure. The ν value is computed to be 0.1. For $\log\gamma < 2$ both SVC and LSVC detect only one cluster. For $\log\gamma = 2$ LSVC and SVC detect four clusters (see Figure 6 left) and as $\hat{k} > k$ the procedure terminates. LSVC identifies 62 bounded support vectors (the black diamonds on the graph) against only 2 for SVC. The merging of the detected clusters results in 99% accuracy for LSVC and only 54% for SVC (see Figure 6 right). Manually probing among a larger set of (γ, ν)-pairs we managed to identify values for SVC that also produced high accuracy after the merging procedure, but for those values there were multiple nonintuitive clusters detected by the algorithm and some rather complex contour boundaries.

The *Swiss roll* data set is a standard benchmark data for evaluating local unsupervised techniques for clustering and dimensionality reduction [13, 19]. We have removed some of the examples from the original data set to obtain two disconnected non-convex clusters (see Figure 7). The data is three dimensional and contains 900 examples to which we have additionally added some Gaussian noise.

For this experiment, the MFA step of the LSVC algorithm is set to use a two dimensional projection $\mathbf{z}$. The number of required clusters is set to $k = 2$. The tradeoff term is computed as $\nu = 0.07$. $\log\gamma = -1$ is the first value for which the LSVC method detects more than one cluster ($\hat{k} = 9$). The number of bounded support vectors is 65 (see Figure 7 left top). Note that the bounded support vectors are positioned on the periphery of the two clusters, detecting much of the bridging noise that could degrade the clustering approach. Applying the merging procedure yields the clustering presented on Figure 7 top right. The accuracy is again approximately 99%.

The SVC algorithm detects $\hat{k} = 5$ clusters for $\log\gamma = -2$, and thus the automatic procedure terminates. Four of the clusters, however, correspond to some small dense neighborhoods and do not detect the two large point formations in the data (see Figure 7 bottom left). Only one bounded support vector was found, underestimating significantly the amount of noise present. The accuracy after merging is 78% with most points from the smaller cluster being assigned to the larger one. We again manually probe for other possible parameters that can produce a more accurate merging step for SVC. We find that the pair $(\log\gamma = -1, \nu = 0.07)$ identifies 14 clusters and 16 bounded support vectors (see Figure 7 bottom right), which after merging do lead to high accuracy as in the LSVC algorithm. Again, in this case, the detection of the suitable values required additional supervision and still produced larger number of not very representative small clusters.

6.2 Face Data Set

The *Frey face* images have been demonstrated by Roweis et al. [14] to reside on a smooth two dimensional manifold. Several examples of the images are presented in Figure 8, top right. The position of the examples on the manifold is determined by the expression of the face and the rotation of the head. Those are the features that separate the data into the two dense clouds seen in the figure. Every example is

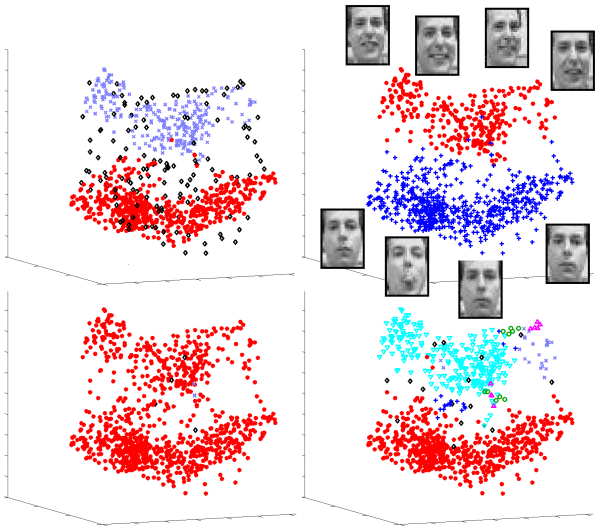

Figure 8: *Top*: the proposed LSVC algorithm; *left*: clusters identified by the automatic procedure; *right*: merging to obtain only two clusters. *Bottom*: the SVC algorithm; *left*: 1 large and 1 small nonrepresentative cluster are identified with the automatic procedure; *right*: using supervision we detect parameters that lead to better separation, but still with some nonrepresentative clusters.

recorded as a 560 dimensional vector (the images are 20x28 pixels), where the dimensions correspond to the greyscale intensities of each pixel. In the evaluation here we randomly select 1000 examples from the original data set.

The data are very high dimensional and, as previously mentioned, the density estimation approach in this case may not lead directly to reasonable results. Therefore, we first reduce the dimensionality using PCA and we work instead with the three dimensional projection along the top three eigenvectors. We further require that $k = 2$, aiming to detect the two dense formations that can be observed on the PCA projection in Figure 8.

The MFA step is again set to use two dimensional projections $\mathbf{z}$. The tradeoff ν is computed to be 0.07 and $\log\gamma = -14$ is the first γ for which LSVC detects more than one cluster. The algorithm identifies exactly $\hat{k} = 2$ clusters and 129 bounded support vectors which again outline correctly the bridging noise between the two distributions (see Figure 8 top left). Assigning the bounded support vectors to the closest dense region results in the clustering demonstrated in Figure 8 top right.

For the SVC algorithm $\log\gamma = -13$ yields the kernel width that first detects more than one cluster ($\hat{k} = 2$). One of the clusters, however, is a small region of just a few elements (see Figure 8 bottom left). The merging step does not change this result either. Increasing $\log\gamma$ twice did lead us to better cluster assignment (see Figure 8 bottom right), which after merging the multiple clusters produced

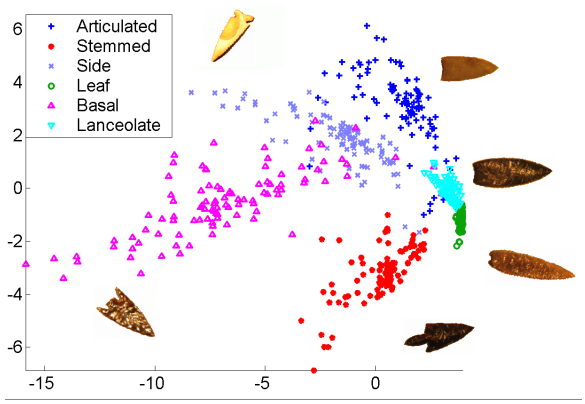

Figure 9: *Arrowheads* data set. 2D MDS projection with representative examples for the six classes present in the data.

two clusters similar to the ones identified with LSVC. However, the value required additional supervision and also detected multiple non-representative clusters. Moreover, very few of the scattered examples between the two dense formations were detected as noise (i.e. bounded support vectors).

6.3 Arrowheads Data Set

The *Arrowheads* data set contains time series extracted from the shape contours of 600 projectile images. There are six classes of projectiles labeled in the collection. The time series were formed by computing the distance from every point of the shape's contour to its centroid [5]. To allow for rotation and scale invariance, we have further aligned and resampled all time series in the data set, representing them with 340 dimensional vectors. The data is then projected using the two largest eigenvectors (see Figure 9).

The data set is rather difficult to discriminate, with many bridging elements between the available classes, and with some classes (*leaf* and *lanceolate*) significantly overlapping. We run SVC and LSVC with $k = 6$. The MFA projection $\mathbf{z}$ is again two dimensional. The value for ν is computed to be 0.09. The contours detected by the two methods and the clusters after the merging procedure are presented in Figure 10.

Both methods detect less than six clusters for $\log\gamma < 1$. For $\log\gamma = 1$, LSVC finds 19 clusters and isolates 60 bounded support vectors (see Figure 10 top left). After the merging procedure, we map the six clusters that we identify to the original labels that yield highest accuracy. The result is presented in Figure 10 top right. The accuracy of the method is $\sim 73\%$. In summary, the LSVC method performs well and succeeds in capturing the objectively dense regions in the data.

The SVC approach fails to separate the stemmed class, and hence the worse accuracy of the clustering $\sim 60\%$ (see Figure 10 bottom right). The number of clusters detected by

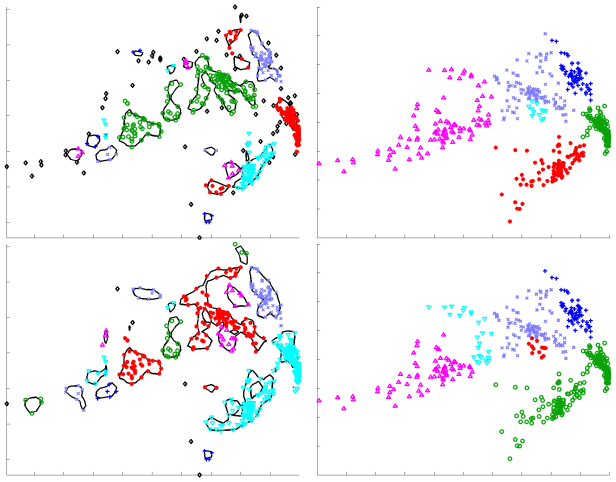

Figure 10: *Top*: the proposed LSVC algorithm; *left*: contours and clusters identified by the automatic procedure (colors are assigned agnostically); *right*: merging to obtain six clusters. *Bottom*: SVC algorithm; *left*: identified contours and clusters (colors are assigned agnostically). The method tries to accommodate much of the noise building more complex boundaries; *right*: merging to obtain six clusters. The accuracy is significantly lower compared to the LSVC algorithm: 60% vs 73%.

the method is 18 and the number of bounded support vectors is six (see Figure 10 bottom left). SVC also identifies some objectively dense regions in the data, but the contours are again more complex and tend to accommodate most of the bridging elements between the different classes.

7 Conclusions and Future Work

We presented a method for improving the stability of the support vector clustering (SVC) algorithm in the presence of noise and bridging elements between the available clusters. The introduced algorithm uses a mixture of factor analyzers (MFA) to learn a weighting, representing the confidence that a certain example is an outlier. The weights are later used to regularize the complexity of the decision function computed for the clustering. On synthetic and real data sets, we demonstrated that our method produces superior results than SVC alone. The results also indicate that complementing the best features from local and global clustering approaches can provide for a powerful tool for learning of clusters sampled from nonlinear manifolds.

Though the algorithm is fairly robust to a not very precise specification of the number of factor analyzers, it would be useful to have an automatic procedure that removes the need of specifying this parameter. The Dirichlet processes have been demonstrated as suitable means for inferring the number of components in mixture models. We are currently exploring their applicability in the settings of the LSVC algorithm.

References

[1] M. Balasubramanian and E. Schwartz. The Isomap algorithm and topological stability. *Science*, 295(5552):7, 2002.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.

[3] A. Ben-Hur, D. Horn, H. Siegelmann, and V. Vapnik. Support vector clustering. *J. Mach. Learn. Res.*, 2:125–137, 2002.

[4] M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering. *Proc. of the 9-th International Workshop on Artificial Intelligence and Statistics*, 2003.

[5] C. Chang, S. Hwang, and D. Buehrer. A shape recognition scheme based on relative distances of feature points from the centroid. *Pattern Recognition*, 24(11):1053–1063, 1991.

[6] S. Du and S. Chen. Weighted support vector machine for classification. In *Proc. Internation Conference on Systems, Man and Cybernetics*, volume 4, pages 3866– 3871, 2005.

[7] B. Everitt. *An Introduction to Latent Variable Models (Monographs on Statistics and Applied Probability)*. Chapman & Hall, 1984.

[8] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

[9] Z. Ghahramani and G. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, 1996.

[10] C. Lee and A. Elgammal. Simultaneous inference of view and body pose using torus manifolds. In *Proc. of the 18th International Conference on Pattern Recognition (ICPR)*, pages 489–494, 2006.

[11] S. Lee and K. Daniels. Gaussian kernel width generator for support vector clustering. In *Proc. of the International Conference on Bioinformatics and its Applications. Series in Mathematical Biology and Medicine*, volume 8, pages 151–162, 2005.

[12] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, 2001.

[13] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[14] S. Roweis, L. Saul, and G. Hinton. Global coordination of local linear models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, 2002.

[15] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471, 2001.

[16] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[17] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. *Advances in Neural Information Processing Systems (NIPS)*, pages 582–588, 2000.

[18] D. Tax and R. Duin. Support vector data description. *Mach. Learn.*, 54(1):45–66, 2004.

[19] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.