The Entire Regularization Path for the Support Vector Domain Description

Karl Sjöstrand[1,2] and Rasmus Larsen[1]

[1] Informatics and Mathematical Modelling, Technical University of Denmark
[2] Department of Radiology, VAMC, University of California-San Francisco, USA
kas@imm.dtu.dk, rl@imm.dtu.dk

Abstract. The support vector domain description is a one-class classification method that estimates the shape and extent of the distribution of a data set. This separates the data into outliers, outside the decision boundary, and inliers on the inside. The method bears close resemblance to the two-class support vector machine classifier. Recently, it was shown that the regularization path of the support vector machine is piecewise linear, and that the entire path can be computed efficiently. This paper shows that this property carries over to the support vector domain description. Using our results the solution to the one-class classification can be obtained for any amount of regularization with roughly the same computational complexity required to solve for a particularly value of the regularization parameter. The possibility of evaluating the results for any amount of regularization not only offers more accurate and reliable models, but also makes way for new applications. We illustrate the potential of the method by determining the order of inclusion in the model for a set of corpora callosa outlines.

1 Introduction

The support vector domain description (SVDD) [1, 2] is a one-class classification method for unlabeled data that is closely related to the support vector machine (SVM) [3] for labeled (training) data with two or more classes. While SVM separates two classes using a hyperplane, SVDD separates data from outliers using a hypersphere. The idea is to find the minimal sphere that encapsulates the data, allowing for some points to be outside the boundary. Formally, this amounts to,

$$\min_{R^2,\mathbf{a},\xi_i} \sum_i \xi_i + \lambda R^2, \quad \text{subject to} \quad (\mathbf{x}_i - \mathbf{a})(\mathbf{x}_i - \mathbf{a})^T \leq R^2 + \xi_i, \quad \xi_i \geq 0 \quad \forall i, \tag{1}$$

where $\mathbf{X} = [\mathbf{x}_1^T \dots \mathbf{x}_n^T]^T$ is the $n \times p$ data matrix with $\mathbf{x}_i \in \mathbb{R}^p$, R is the radius, $\mathbf{a}$ is the center of the sphere, and ξ_i is the amount by which point i is allowed to be outside the sphere. The parameter λ determines the amount of regularization. A large value of λ puts focus on the radius, leading to a smaller sphere while the ξ_i are allowed to grow large. If λ is small, the resulting sphere will be large

while the total distance of outlying points shrinks. The setup in Equation 1 is convex and can be solved using Lagrange multipliers $\alpha_i \geq 0$ and $\gamma_i \geq 0$,

$$L_p : \sum_i \xi_i + \lambda R^2 + \sum_i \alpha_i(\mathbf{x}_i\mathbf{x}_i^T - 2\mathbf{a}\mathbf{x}_i^T + \mathbf{a}\mathbf{a}^T - R^2 - \xi_i) - \sum_i \gamma_i\xi_i. \quad (2)$$

At the minimum, the derivative of each variable is zero, giving

$$\frac{\partial L_p}{\partial R^2} = 0 \Leftrightarrow \lambda = \sum_i \alpha_i, \quad (3)$$

$$\frac{\partial L_p}{\partial \mathbf{a}} = 0 \Leftrightarrow \mathbf{a} = \frac{\sum_i \alpha_i \mathbf{x}_i}{\sum_i \alpha_i} = \frac{\sum_i \alpha_i \mathbf{x}_i}{\lambda}, \quad (4)$$

$$\frac{\partial L_p}{\partial \xi_i} = 0 \Leftrightarrow \alpha_i = 1 - \gamma_i. \quad (5)$$

Equation 5 and the fact that $\alpha_i \geq 0, \gamma_i \geq 0, \forall i$, gives that $0 \leq \alpha_i \leq 1$. Furthermore, we have the Karush-Kuhn-Tucker complimentary conditions,

$$\alpha_i(\mathbf{x}_i\mathbf{x}_i^T - 2\mathbf{a}\mathbf{x}_i^T + \mathbf{a}\mathbf{a}^T - R^2 - \xi_i) = 0, \quad (6)$$

$$\gamma_i\xi_i = 0. \quad (7)$$

Equations 5, 6 and 7 give that $\alpha_i = 1$ for points outside the sphere and $\alpha_i = 0$ for points on the inside. By continuity, α_i will travel from 1 to 0 as point i passes the boundary from outside the sphere to the inside. Inserting Equations (3-5) into (2) gives the Wolfe dual form which is to be maximized w.r.t. (3-5),

$$L_w : \max_{\alpha_i} \sum_i \alpha_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{\lambda}\sum_i\sum_j \alpha_i\alpha_j \mathbf{x}_i\mathbf{x}_j^T \quad : \quad 0 \leq \alpha \leq 1, \quad \sum_i \alpha_i = \lambda. \quad (8)$$

This is a quadratic optimization problem with linear constraints. As such, it can be solved using some quadratic programming algorithm. This is a slight reformulation of the original setup [1] which uses a regularization parameter $C = 1/\lambda$. As in [4] we favor the description above since $0 \leq \alpha \leq 1$ instead of $0 \leq \alpha \leq C$ which facilitates the interpretation of the coefficient paths $\alpha_i(\lambda)$.

Equation 3 determines the valid range for the regularization parameter to $\lambda \in [0, n]$.

For most data sets, a hypersphere is an unsuitable representation of the data. Increasing the dimensionality using basis expansions $h(\mathbf{x})$ allows for more flexible decision boundaries. Replacing $\mathbf{x}$ by $h(\mathbf{x})$ in Equation 8, we note that the dual can be expressed in terms of inner products $\langle h(\mathbf{x}_i), h(\mathbf{x}_j)\rangle$. These can be replaced by $K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$, where K is some suitable kernel function,

$$L_w : \max_{\alpha_i} \sum_i \alpha_i K_{i,i} - \frac{1}{\lambda}\sum_i\sum_j \alpha_i\alpha_j K_{i,j} \quad : \quad 0 \leq \alpha \leq 1, \quad \sum_i \alpha_i = \lambda. \quad (9)$$

In the remainder of this paper, the more general kernel notation will be used.

The squared distance from the center of the sphere to a point $\mathbf{x}$ is,

$$f(\mathbf{x}) = \|h(\mathbf{x}) - \mathbf{a}\|^2 = K(\mathbf{x},\mathbf{x}) - \frac{2}{\lambda}\sum_i \alpha_i K(\mathbf{x},\mathbf{x}_i) + \frac{1}{\lambda^2}\sum_i\sum_j \alpha_i\alpha_j K_{i,j} \quad (10)$$

The squared radius of the sphere can therefore be written $R^2 = f(\mathbf{x}_k)$, where index k belongs to any point on the boundary ($\alpha_k \in (0,1)$). Note that $f(\mathbf{x})$, R^2, α_i and index k are all dependent on λ.

2 Computing the Regularization Path

In this section we will prove that the coefficient path of each α_i is piecewise linear in λ, and purpose an algorithm for their calculation using standard matrix algebra. The increased interest in coefficient paths and their computation originates from the seminal work by Efron et al. [5], where the regularization path of the LASSO regression algorithm is derived, leading to a highly efficient algorithm. These results have since been generalized to hold for a range of regularized problems [6, 7]. Specifically, it holds for support vector machines as described by Hastie et al. [4]. Due to the alikeness of SVMs and the SVDD, the following derivation of the regularization path for the SVDD is noticeably similar to that of SVMs.

Define by $\mathcal{I}$, $\mathcal{O}$ and $\mathcal{B}$ the sets containing indices i corresponding to points on the inside, outside and on the boundary respectively, and let $n_\mathcal{I}$, $n_\mathcal{O}$, and $n_\mathcal{B}$ be the number of elements in these sets. The set $\mathcal{A} = \mathcal{I} \cup \mathcal{O} \cup \mathcal{B}$ contains the indices of all points.

As discussed above, $\alpha_i = 1$ for $i \in \mathcal{O}$, $\alpha_i = 0$ for $i \in \mathcal{I}$, and $0 < \alpha_i < 1$ for $i \in \mathcal{B}$. There are four types of events where these sets change.

1. Point i leaves $\mathcal{B}$ and joins $\mathcal{I}$; $\alpha_i \in (0,1) \longrightarrow \alpha_i = 0$.
2. Point i leaves $\mathcal{B}$ and joins $\mathcal{O}$; $\alpha_i \in (0,1) \longrightarrow \alpha_i = 1$.
3. Point i leaves $\mathcal{O}$ and joins $\mathcal{B}$; $\alpha_i = 1 \longrightarrow \alpha_i \in (0,1)$.
4. Point i leaves $\mathcal{I}$ and joins $\mathcal{B}$; $\alpha_i = 0 \longrightarrow \alpha_i \in (0,1)$.

To determine which set a point $\mathbf{x}$ belongs to, we define a decision function,

$$g(\mathbf{x}) = f(\mathbf{x}) - f(\mathbf{x}_k) = K(\mathbf{x},\mathbf{x}) - K_{k,k} - \frac{2}{\lambda}\sum_i \alpha_i\big(K(\mathbf{x},\mathbf{x}_i) - K_{k,i}\big),\ k \in \mathcal{B}, \quad (11)$$

which has $g = 0$ for $\mathbf{x}$ on the boundary, $g < 0$ for $\mathbf{x}$ inside and vice versa.

The algorithm starts at $\lambda = n$, corresponding to the minimal sphere radius with $\mathcal{O} = \mathcal{A}$ and $\alpha_i = 1$, $\forall i$.

From an arbitrary configuration of $\mathcal{I}$, $\mathcal{O}$ and $\mathcal{B}$, λ is allowed to decrease until the next event occurs. As in [4], let λ_l be the value of the regularization parameter at step l. While $\lambda_l < \lambda < \lambda_{l+1}$, the sets remain static. Hence, $g(\mathbf{x}_m) = 0, \forall\, m \in \mathcal{B}$ in this interval. Using this, Equation 11 can be expanded and rearranged into

$$\sum_{i\in\mathcal{B}} \alpha_i(K_{m,i} - K_{k,i}) = \frac{\lambda}{2}(K_{m,m} - K_{k,k}) - \sum_{i\in\mathcal{O}}(K_{m,i} - K_{k,i})\ \ \forall m \in \mathcal{B}. \quad (12)$$

This results in $n_{\mathcal{B}}$ equations with $n_{\mathcal{B}}$ unknowns $\alpha_i, i \in \mathcal{B}$. However, for $m = k$, it is seen that (12) degenerates into $0 = 0$, making the system of equations rank deficient. We therefore replace the equation for $m = k$ by the auxiliary condition in Equation 3.

This procedure can be summarized in matrix form. Let $\mathbf{Y}$ be an $n \times n$ matrix where $\mathbf{Y}_{i,j} = K_{i,j} - K_{k,j},\ \forall\,(i,j) \in \mathcal{A}$ and let $\mathbf{y}$ be a length n vector with $y_i = K_{i,i} - K_{k,k}\ \forall i \in \mathcal{A}$. With the obvious definitions of submatrices, Equation 12 can be written

$$\mathbf{Y}_{\mathcal{B},\mathcal{B}}\alpha_{\mathcal{B}} = \frac{\lambda}{2}\mathbf{y}_{\mathcal{B}} - \mathbf{Y}_{\mathcal{B},\mathcal{O}}\mathbf{1}_{n_{\mathcal{O}}}, \tag{13}$$

where $\mathbf{1}_{n_{\mathcal{O}}}$ is a vector of ones of length $n_{\mathcal{O}}$. This expression can be expanded to include the conditions $\alpha_i = 0$ for $i \in \mathcal{I}$ and $\alpha_i = 1$ for $i \in \mathcal{O}$. It also needs to be augmented to replace the degenerate equation corresponding to index k with the relation from Equation 3. We will now define matrices that implement this.

Let $\mathcal{B}_{-k}$ be the boundary set with index k removed. Let $\mathbf{Z}$ be the $n \times n$ identity matrix with $\mathbf{Z}_{\mathcal{B}_{-k},\mathcal{B}} = \mathbf{Y}_{\mathcal{B}_{-k},\mathcal{B}}$ and $\mathbf{Z}_{k,\mathcal{A}} = \mathbf{1}_n^T$. Let $\mathbf{z}$ be the length n zero vector with $\mathbf{z}_{\mathcal{B}_{-k}} = \mathbf{y}_{\mathcal{B}_{-k}}$ and $z_k = 2$. Finally, let $\mathbf{W}$ be the $n \times n$ zero matrix with $\mathbf{W}_{\mathcal{B}_{-k},\mathcal{O}} = -\mathbf{Y}_{\mathcal{B}_{-k},\mathcal{O}}$ and $\mathbf{W}_{\mathcal{O},\mathcal{O}} = \mathbf{I}_{n_{\mathcal{O}}}$ where $\mathbf{I}_{n_{\mathcal{O}}}$ is the identity matrix of size $n_{\mathcal{O}}$. The complete system of n equations for n unknowns is then

$$\mathbf{Z}\alpha = \frac{\lambda}{2}\mathbf{z} + \mathbf{W}\mathbf{1}_n. \tag{14}$$

Providing $\mathbf{Z}$ is invertible, the resulting expression for α becomes,

$$\alpha = \frac{\lambda}{2}\mathbf{Z}^{-1}\mathbf{z} + \mathbf{Z}^{-1}\mathbf{W}\mathbf{1}_n \equiv \lambda\mathbf{p} + \mathbf{q}, \tag{15}$$

an expression that is linear in λ.

Now that the functional form of each coefficient between two events has been established, it remains to disclose the valid range $[\lambda_l, \lambda_{l+1}]$ of λ. That is, we wish to find λ_{l+1} at which the next event occurs. We treat each of the four types of events defined above separately.

The first event type occurs for α_i, $i \in \mathcal{B}$ when $\alpha_i \to 0$. By setting (15) equal to 0 and solving for each value of λ, we get,

$$\lambda_i = -\frac{q_i}{p_i}, \quad i \in \mathcal{B}. \tag{16}$$

Similarly for event two, $\alpha_i = 1$ when

$$\lambda_i = \frac{1 - q_i}{p_i}, \quad i \in \mathcal{B}. \tag{17}$$

For either of the other two event types to occur, a point i in either $\mathcal{I}$ or $\mathcal{O}$ must join the boundary. At this stage, $g(\mathbf{x}_i) = 0$. Inserting (15) into (11), the value of the decision function at point i for some value of λ can be expressed as

$$g(\mathbf{x}_i, \lambda) = y_i - 2\mathbf{Y}_{i,\mathcal{A}}(\mathbf{p} + \frac{1}{\lambda}\mathbf{q}). \tag{18}$$

To find the values of λ at which each point joins the boundary, $g(\mathbf{x}_i, \lambda_i) = 0$ is solved for λ_i,

$$\lambda_i = \frac{2\mathbf{Y}_{i,\mathcal{A}}\mathbf{q}}{y_i - 2\mathbf{Y}_{i,\mathcal{A}}\mathbf{p}}. \tag{19}$$

Out of the candidates $\{\lambda_i\}$ for λ_{l+1}, the largest candidate smaller than λ_l must be the point at which the sets first change. Therefore, $\lambda_{l+1} = \max_i \lambda_i$ subject to $\lambda_i < \lambda_l$.

There is one final consideration. Event 1 may at any stage of the algorithm lead to the boundary set $\mathcal{B}$ becoming empty, resulting in a violation of Equation 3. One or more points from $\mathcal{O}$ must therefore join $\mathcal{B}$ concurrently. The calculation of candidates for λ_{l+1} in (19) will fail in this case, as a consequence of the new point not being placed on the current boundary. This behavior forces a discontinuity in the radius function, which must increase discretely to encompass the next point. Since $\alpha(\lambda)$ is a continuous function, Equation 4 shows that the position of the hypersphere center $\mathbf{a}(\lambda)$ is also continuous. Hence, despite the discontinuity of the boundary function, the next point to join $\mathcal{B}$ can be established by finding the point in $\mathcal{O}$ with the smallest radius,

$$\min_{i\in\mathcal{O}} f(\mathbf{x}_i) = \min_{i\in\mathcal{O}} K_{i,i} - \frac{2}{\lambda}K_{i,\mathcal{A}}\alpha + \frac{1}{\lambda^2}\alpha^T K_{\mathcal{A},\mathcal{A}}\alpha. \tag{20}$$

The entire process is summarized in Algorithm 1.

Algorithm 1 SVDD coefficient paths

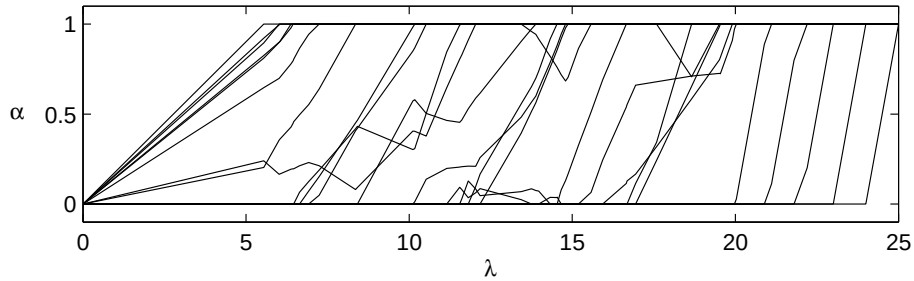
Initialize $\lambda = n$ and $\alpha_i = 1\ \forall i$.
while $\lambda > 0$ **do**
  if $n_{\mathcal{B}} = 0$ **then**
    Add index i to boundary set $\mathcal{B}$ that satisfies (20).
    Remove i from $\mathcal{O}$.
  end if
  Given sets $\mathcal{I}$, $\mathcal{O}$ and $\mathcal{B}$, compute $\mathbf{p} = \mathbf{Z}^{-1}\mathbf{z}/2$ and $\mathbf{q} = \mathbf{Z}^{-1}\mathbf{W}\mathbf{1}_n$.
  Calculate λ candidates according to event 1 using (16).
  Calculate λ candidates according to event 2 using (17).
  Calculate λ candidates according to event 3 using (19) with $i \in \mathcal{O}$.
  Calculate λ candidates according to event 4 using (19) with $i \in \mathcal{I}$.
  Choose candidate λ_{l+1} with the largest value smaller than λ_l.
  Calculate new coefficients, $\alpha = \lambda_{l+1}\mathbf{p} + \mathbf{q}$.
  Update sets accordingly.
end while

3 Examples

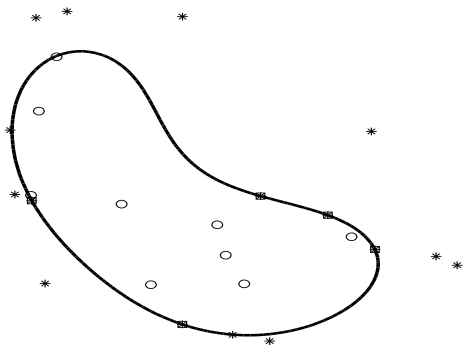
The algorithm was implemented using MATLAB and tested on both synthetic and real data sets. For the results presented here, a Gaussian kernel with $K_{i,j} =$

$\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\gamma)$ has been used. High values of γ produce smooth and coherent decision boundaries while smaller values give more wiggly and clustered results. The choice of this parameter is application specific and is not discussed in this paper. For the examples shown here, $\gamma = 1$ is used which results in smooth, but not excessively constrained boundaries.

Figure 1 shows a small data set of 25 points in two dimensions. Figure 1(a) shows the resulting regularization path for λ values ranging from 0 to $n = 25$. The



(a) Sample regularization path for the small data set below.



(b) Decision boundary at $\lambda = 13$. Circles, stars and squares respectively denote inliers, outliers and points on the boundary.

Fig. 1. Example description of 25 points in two dimensions with the corresponding regularization path.

The second experiment is an application that uses the asset of knowing the entire regularization path. The goal is to order a large set of Procrustes aligned shapes in ascending order according to the density of the corresponding distribu-

tion at each observation. This is done using two approaches. The first is based on successive maximization of Mahalanobis distance. At each step, the observation with the largest distance w.r.t. the current data set is removed. For n shapes, this is performed $n-1$ times, thus establishing an ordering. The second method uses the SVDD and its regularization path. The order is established directly from $\mathcal{O}$ as λ grows from 0 to n. The data consists of 582 outlines of the mid-sagittal cross-section of the corpus callosum brain structure. This data set is part of the LADIS (Leukoaraiosis and DISability) study [8], a pan-European study involving 12 hospitals and more than 600 patients. Figures 2 and 3 show the first and last twelve observations of each ordering. The Mahalanobis distance measure is based on the shape of the covariance matrix and assumes an ellipsoidal distribution. Due to the use of kernels, the SVDD is able to model more complex distributions, giving better estimates of the density at each observation. This is particularly apparent among the inliers in Figure 3. The variance is clearly lower for the SVDD-based ordering than for the Mahalanobis-based counterpart.

(a) First 8 selected (outliers)

(b) Last 8 selected (inliers)

Fig. 2. Ordering established by successive maximization of Mahalanobis distance.

(a) First 8 selected (outliers)

(b) Last 8 selected (inliers)

Fig. 3. Ordering established by the SVDD regularization path. Note the increased dissimilarity among the outliers, as well as the increased similarity among later samples.

4 Discussion

The computational complexity of computing the entire path is low. Most of the effort goes into solving the linear system in (15). To increase efficiency, we solve (15) for points on the boundary only, i.e. using submatrices $\mathbf{Z}_{\mathcal{B},\mathcal{B}}$, $\mathbf{z}_{\mathcal{B}}$ and $\mathbf{W}_{\mathcal{B},\mathcal{O}}$. The remaining values of α_i ($i \in \mathcal{I} \cup \mathcal{O}$) remain static. The resulting burden for finding $\mathbf{p}$ and $\mathbf{q}$ is roughly $O(n_{\mathcal{B}}^3)$ where typically $n_{\mathcal{B}} \ll n$. The most prominent

addition to this is the work involved in finding λ_{l+1} which includes the multiplication of several length n vectors. In our experience, the number of iterations is generally less than $2n$, although more than $5n$ iterations is possible for very dense data sets. In comparison, algorithms for solving quadratic programming problems have $O(n^k)$, with k dependent on the choice of implementation.

Due to the exclusive use of kernels, the method handles data with many variables well. The memory usage level is mainly due to the matrix $\mathbf{Y}_{\mathcal{B},\mathcal{B}}$, which can grow large for data sets with many observations and the use of very unconstrained decision boundaries.

Knowledge of the entire regularization path is an important basis for picking the appropriate amount of regularization. As shown in this paper, information from the path itself can also be used directly as a data description method. Furthermore, the SVDD is the basis for other methods such as support vector clustering (SVC) [9], which may benefit from these results.

Acknowledgement

The authors extend their gratitude to the LADIS work group for supplying the corpus callosum data. In particular we acknowledge the annotation effort of Charlotte Ryberg and Egill Rostrup from the Danish Research Center for Magnetic Resonance, Copenhagen University Hospital, Hvidovre, Denmark.

References

1. Tax, D.M., Duin, R.P.: Support vector domain description. Pattern Recognition Letters **20**(11-13) (1999) 1191–1199
2. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Computation **13** (2001) 1443–1471
3. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
4. Hastie, T., Rosset, S., Tibshirani, R., Zhu, J.: The entire regularization path for the support vector machine. JMLR **5** (2004) 1391–1415
5. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Annals of Statistics **32**(2) (2004) 407–451
6. Rosset, S., Zhu, J.: Piecewise linear regularized solution paths. Technical report, Stanford University (2003)
7. Rosset, S.: Tracking curved regularized optimization solution paths. NIPS (2004)
8. Pantoni, L., Basile, A.M., Pracucci, G., Asplund, K., Bogousslavsky, J., Chabriat, H., Erkinjuntti, T., Fazekas, F., Ferro, J.M., Hennerici, M., O'brien, J., Scheltens, P., Visser, M.C., Wahlund, L.O., Waldemar, G., Wallin, A., Inzitari, D.: Impact of age-related cerebral white matter changes on the transition to disability - the LADIS study: Rationale, design and methodology. Neuroepidemiology **24**(1-2) (2005) 51–62
9. Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V.: Support vector clustering. Journal of Machine Learning Research **2** (2001) 125–137