# A Geometric Viewpoint of the Selection of the Regularization Parameter in Some Support Vector Machines

Nandyala Hemachandra and Puja Sahu

Indian Institute of Technology Bombay, Mumbai, India
{nh,puja.sahu}@iitb.ac.in

**Abstract.** The regularization parameter of support vector machines is intended to improve their generalization performance. Since the feasible region of binary class support vector machines with finite dimensional feature space is a polytope, we note that classifiers at vertices of this unbounded polytope correspond to certain ranges of the regularization parameter. This reduces the search for a suitable regularization parameter to a search of (finite number of) vertices of this polytope. We propose an algorithm that identifies neighbouring vertices of a given vertex and thereby identifies the classifiers corresponding to the set of vertices of this polytope. A classifier can then be chosen from them based on a suitable test error criterion. We illustrate our results with an example which demonstrates that this path can be complicated. A portion of the path is sandwiched between two finite intervals of path, each generated by separate sets of vertices and edges.

**Keywords:** Support vector machines, regularization path, polytopes, neighbouring vertices, prediction error, parameter tuning, linear programming.

## 1 Introduction

A classical learning problem is that of binary classification wherein the learner is trained on a given data set (training set) and predicts the class of a new data point. Let the $n$ point training set be $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^m$ is a vector of $m$ features and $y_i \in \{-1, +1\}$ is the label of $\mathbf{x}_i,\ i \in \{1, \cdots, n\}$. We consider the class of linear classifiers, $(\mathbf{w}, b)$, with $\mathbf{w} \in \mathbb{R}^m$ and $b \in \mathbb{R}$. The classifier predicts the class of data point $\mathbf{x}$ as $-1$ if $\mathbf{w} \cdot \mathbf{x} + b < 0$ and predicts the class as $+1$ otherwise, i.e., the predicted class for $\mathbf{x}$ is $\text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$. Such classifiers are called linear Support Vector Machines (SVMs).

Among finite dimensional models for binary class prediction, the class of polynomial kernels form an important class. These are quite popular in natural language processing (NLP) because fast linear SVM methods can be applied to the polynomially mapped data and can achieve accuracy close to that of using highly nonlinear kernels [2].

The standard soft-margin SVM optimization problem (SVM QP), for a given $\lambda > 0$ [6] is:

$$\begin{aligned} &\min_{\mathbf{w},b,\boldsymbol{\xi}} \lambda||\mathbf{w}|| + \sum_{i=1}^{n} \xi_i \\ &\text{s. t. } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i \in \{1\dots n\} \\ &\qquad\qquad \xi_i \geq 0 \quad \forall i \in \{1\dots n\} \end{aligned} \tag{1}$$

The objective function is a sum of the regularization penalty term with regularization parameter $\lambda \in \mathbb{R}^+$ and the classification error (measured from the margins, $\mathbf{w}\cdot\mathbf{x}+b = \pm 1$, of the classifier) as captured by $\{\xi_i\}_{i=1}^n$.We are working with linearly inseparable data. Therefore, $\xi_i > 0$ for at least one $i \in \{1,\dots,n\}$. The points lying on the margins of the classifier are called support vectors. Hence the name support vector machines.

The purpose of the regularization parameter $\lambda$ is to improve the generalization error of the SVM. It is known that a proper choice is needed; see, for example, Figure 4 of [7]. The purpose of this paper is to investigate this choice in fairly basic SVMs by considering the polyhedral nature of the feasible region of the above SVM QP.

The main results of this paper are summarized as follows: We characterize, to the best of our knowledge for the first time, the polytope, $P$, associated with the feasible space of (1), in terms of its vertices and give an algorithm that lists all its vertices. We notice that, starting off from a vertex, the path is generated by vertices and edges (one-dimensional facets) as well as facets of higher dimensions. The regularization parameter, $\lambda$, for any classifier can be identified by linear programs; and for classifiers corresponding to vertices, this is an interval. The SVMs are generally assessed in terms of their performance on $0-1$ loss criterion. We find that the vertex classifiers dominate other boundary classifiers on a single test point using this $0-1$ loss function. This means that for the SVMs that we consider, a suitable choice of $\lambda$ as a design parameter can be replaced by a search among the finite but large number of the vertices of $P$.

Different approaches have been employed to select an optimal regularization parameter, $\lambda$, for the SVM QP. The task of tracing an entire regularization path was pioneered by [7]. The sets $E, L$ and $R$ of [7] in the feature space $(\mathbf{w}, b)$ correspond to a vertex $v$ of the polytope $P$ in the lifted space in $(\mathbf{w}, b, \boldsymbol{\xi})$. Another approach [3] considers finding the optimal parameters for SVM based classifiers with kernel functions that could be infinite dimensional, where $\lambda$ is included in the parameter vector. Bounds on the test error are obtained, based on the leave-one-out testing scheme and these are differentiable with respect to the parameter vector. A gradient based scheme is proposed for finding optimal parameters. Apart from tracing the path, various other aspects have been studied regarding the design of the SVMS, such as the feature selection problem [3], [9].

Note that, to trace the path, [7] use the dual optimization program to the SVM QP to study the trajectories of the primal and dual variables as a function of the regularization parameter; whereas the polytope considered in this paper

resides in the primal space itself. As a consequence, we need to search among a finite, albeit a large, set of vertices. And unlike [3], we restrict our analysis to the case of finite dimensional kernels, which can be handled using fast algorithms.

## 2 The Polytope of the Feasible Region of SVMs

First we notice that the feasible region is unbounded; hence it admits a Minkowski decomposition into a base polytope, $P$, and a recession cone. We want to concentrate on characterizing $P$ in terms of its vertices and more importantly, the role of these vertices in the regularization path of the SVM.

**Theorem 1.** *For a given $\lambda \geq 0$, the optimal point (the classifier for the SVM) lies on the boundary of the polytope $P$.*

*Proof.* Consider the unconstrained problem with the same objective function of SVM QP: $\lambda||\mathbf{w}|| + \sum_i \xi_i$. This optimization is separable into two optimization problems: $\min_{\mathbb{R}^m} \lambda||\mathbf{w}||$ and $\min_{\mathbb{R}^n} \sum_i \xi$. While the first one has the optimal value zero at $\mathbf{w} = 0$, the second one is unbounded. Hence the unconstrained problem has an unbounded value, whereas the SVM QP has a finite non-negative optimal value. The SVM QP is a convex minimization problem and hence its finite optimal solution will lie on the boundary of its feasible region, the polytope $P$. □

**Theorem 2.** *A classifier on the vertex of the polytope dominates a boundary classifier, i.e., a classifier corresponding to an edge or a facet, on $0-1$ loss function.*

*Proof.* Consider two classifiers $(\mathbf{w}^1, b^1)$ and $(\mathbf{w}^2, b^2)$ on the $\lambda$-path, lying on two different vertices of the polytope. Suppose, for a test data $\hat{\mathbf{x}}$, we have

$$\begin{aligned} \text{sign}(\mathbf{w}^1 \cdot \hat{\mathbf{x}} + b^1) &= +1 \\ \text{and } \text{sign}(\mathbf{w}^2 \cdot \hat{\mathbf{x}} + b^2) &= -1. \end{aligned}$$

A classifier $(\tilde{\mathbf{w}}, \tilde{b})$ on the related edge can be identified as $(\tilde{\mathbf{w}}, \tilde{b}) = \alpha(\mathbf{w}^1, b^1) + (1-\alpha)(\mathbf{w}^2, b^2)$ for some $\alpha \in (0,1)$. We can see that

$$\text{sign}(\tilde{\mathbf{w}} \cdot \hat{\mathbf{x}} + \tilde{b}) = \begin{cases} 1 & \text{if } \alpha \geq \alpha_0, \\ -1 & \text{if } \alpha < \alpha_0, \end{cases}$$

where $\alpha_0 \in (0,1)$ is the normalized distance of $\hat{\mathbf{x}}$ from $\mathbf{w}^1$. Thus, $(\tilde{\mathbf{w}}, \tilde{b})$ can be dominated on the grounds of 0-1 loss function by one of the vertices depending on the true label of $\hat{\mathbf{x}}$. Dominance over the classifiers belonging to a facet of the polytope can be shown using similar arguments.

*Remark 1.* The above result was shown for a single test point. When we have a collection of the points, we will have a 'dominating set' of vertices, which may or may not lie on the $\lambda$-path.

### 2.1 Characterization of the Vertices in Terms of Active Constraints

We recall the following (see [1], [8], [10], etc.): For a polytope in $\mathbb{R}^k$, a vertex is a point of zero dimension. A vertex in $\mathbb{R}^k$ can be identified as a solution of $k$ linearly independent linear equation. A vertex is an extreme point of the polytope, and can not be obtained as a convex combination of any two distinct points. An edge is a facet of dimension one and is a convex combination of two vertices of the polytope. A facet of dimension two is a convex combination of three vertices and so forth. We define a vertex classifier as a classifier corresponding to a vertex on the polytope of the feasible region of the standard SVM model. The edge and facet classifiers are defined in a similar fashion. Henceforth, these notations will be used in the rest of the paper.

The dimension of $(\mathbf{w}, b, \boldsymbol{\xi})$ is $(m+1+n)$ and we have $n$ linear inequalities with $n$ positively constrained variables, $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)$. (We make the assumption that $(m+1) < n$.) Rewriting them as $n$ equalities with positive slack variables $s_i$, we get a set of $2n$ linear constraints whose intersection gives us a polytope as the feasible region. Hence, if at least $(m+n+1)$ of these $2n$ constraints are active and are linearly independent, the resulting unique solution is a vertex.

So, with $s_i$, we have the following:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 = s_i - \xi_i \quad \forall i \in \{1 \ldots n\}, \tag{2}$$

where $s_i, \xi_i \geq 0 \; \forall i \in \{1 \ldots n\}$.

At a vertex $v$, then, for a given $i \in \{1, \cdots, n\}$, if $\xi_i - s_i \neq 0$, then only one of $\xi_i$ or $s_i$ is non-zero. It can be noted that,

$$s_i - \xi_i = \begin{cases} (-\infty, -2) & \text{if } \mathbf{x}_i \text{ is misclassified by } (\mathbf{w}, b) \text{ and outside the margin} \\ -2 & \text{if } \mathbf{x}_i \text{ is misclassified by } (\mathbf{w}, b) \text{ and on the margin} \\ (-2, -1) & \text{if } \mathbf{x}_i \text{ is misclassified by } (\mathbf{w}, b) \text{ and within the margin} \\ -1 & \text{if } \mathbf{x}_i \text{ is correctly classified by } (\mathbf{w}, b) \text{ and on the classifier} \\ (-1, 0) & \text{if } \mathbf{x}_i \text{ is correctly classified by } (\mathbf{w}, b) \text{ and within the margin} \\ 0 & \text{if } \mathbf{x}_i \text{ is correctly classified by } (\mathbf{w}, b) \text{ and on the margin} \\ & \text{i.e., } \mathbf{x}_i \text{ is a support vector} \\ (0, \infty) & \text{if } \mathbf{x}_i \text{ is correctly classified by } (\mathbf{w}, b) \text{ and outside the margin.} \end{cases}$$

We can identify three different categories of classifiers based on the above values of $(s_i - \xi_i)$, as mentioned in the following theorem:

**Theorem 3.** *There are three types of linear SVM classifiers for the case of binary classification problem:-*

*(i) The classifiers for which the points can be within, on or outside the margin. Thus, $(s_i - \xi_i) \in (-\infty, \infty) \; \forall i \in \{1, \ldots, n\}$ for such classifiers.*

*(ii) The $(\mathbf{0}, 1)$ and $(\mathbf{0}, -1)$ classifiers, for which $\xi_i$ is either 0 or 2 and $(s_i - \xi_i) \in \{0, -2\} \; \forall i \in \{1, \ldots, n\}$.*

*(iii) The classifiers for which all the points are within or on the margins. For such classifiers $(s_i - \xi_i) \in (-2, 0) \; \forall i \in \{1, \ldots, n\}$.*

We have another important characterization of a vertex of the polytope in terms of support vectors of the classifier.

**Theorem 4.** *A vertex $v = (\mathbf{w}, b, \boldsymbol{\xi})$ of the polytope $P$ has at least one correctly classified point on its margin, also known as the support vector. Therefore, for $v \in P$, we have*

$$|\{i \in \{1, \ldots, n\} | \xi_i = s_i = 0\}| \geq 1. \tag{3}$$

*Proof.* For the type (ii) classifiers, the above is trivially true using (2). For the type (i) and type (iii) classifiers, using the definition of a vertex of a polytope, at least $(m + n + 1)$ of the $2n$ constraints in (1) have to be active. Thus, for any set of at least $(m + 1 + n)$ active constraints, $I^*$

$$\begin{aligned} \exists \text{ at least one } i \in I^* \text{ such that } & y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1 - \xi_i \\ & \text{and,} \quad \xi_i = 0. \end{aligned}$$

□

Rewriting the constraints in (1) in a matrix form, we have: $A \cdot (\mathbf{w}, b, \boldsymbol{\xi}) \geq b$, where $A$ corresponds to the coefficient matrix of the two sets of constraints of the SVM QP and $b = \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \end{pmatrix}$.

Let us denote by $I^*(v)$, the set of (indices of) active constraints at vertex $v$. So, given a set of active constraints, $I^*(v)$, we can find the corresponding vertex $v = (\mathbf{w}, b, \boldsymbol{\xi})$ by solving the following equation:

$$A[I^*(v), ] \cdot (\mathbf{w}, b, \boldsymbol{\xi}) = b[I^*(v)], \tag{4}$$

where $A[I^*(v), ]$ is a sub-matrix of $A$ with rows corresponding to $I^*(v)$. Such a vertex corresponds to a basic solution of the feasible region. It is a feasible vertex if and only if it satisfies (1). Equivalently, it is not a feasible point if $s_i$ is strictly negative. A feasible vertex corresponds to a basic feasible solution [1].

Given an active set of constraints, $I^*(v)$, Algorithm VERTEX($Active$)) computes a vertex $v$ corresponding to these active constraint set, if it exists and is feasible.

**Algorithm 1.** Finding a vertex corresponding to an active set of constraints, VERTEX(*Active*)

**Require:** $\{\mathbf{x}_i, y_i\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^m, y_i \in \{+1, -1\}$

**procedure** VERTEX(*Active*)
  Solve for $\tilde{v} = (\tilde{\mathbf{w}}, \tilde{b}, \tilde{\boldsymbol{\xi}})$ using the system of linear equations s.t.

$$y_i \tilde{\mathbf{w}} \cdot \mathbf{x}_i + y_i \tilde{b} + \tilde{\xi}_i = 1 \quad \forall i \in Active \,\&\, i \leq n$$
$$\tilde{\xi}_i = 0 \quad \forall i \in Active \,\&\, i > n$$

  Set $feasible \leftarrow 1$
  **for** $k$ in 1 to $n$ **do**
    **if** $(y_k \tilde{\mathbf{w}} \cdot \mathbf{x}_k + y_k \tilde{b} + \tilde{\xi}_k - 1) < 0$ **then**
      $feasible \leftarrow 0$
      **break**
    **end if**
  **end for**
  **if** $(feasible = 1)$ **then return** $\tilde{v}$
  **else return** $\phi$
  **end if**
**end procedure**

### 2.2 Neighbours of a Vertex of the Polytope, $P$

Given that a vertex is characterized by the set of constraints, $I^*(v)$, that are active at that point, we can find a neighbouring vertex $\tilde{v}$ by changing $I^*(v)$ in the following way:

Replace an active constraint by the one that is currently inactive at $v$. The constraint $i \in I^*(v)$ to leave the active set is the one such that $\xi_i = s_i = 0$. The existence of such a constraint $i \in I^*(v)$ is guaranteed by Theorem 3. The incoming constraint $j \in I^*(v)$ is chosen so that $\{j \mid (s_j > 0 \,\&\, \xi_j = 0)$ or $(\xi_j > 0 \,\&\, s_j = 0)\}$ at $v$. And at the neighbour $\tilde{v}$, we set $\xi_j = s_j = 0$ to ensure a support vector for $\tilde{v}$.

If the solution to (4) with these new active constraints is feasible, then it is a valid neighbour of $v$. Note that if the given vertex $v$ is degenerate, then, the above change in active constraint set $I^*(v)$ can lead to another degenerate vertex and hence not a neighbouring vertex. Such degenerate vertices need to be ignored in the list of neighbours of $v$.

Such a careful updating of the set of neighbouring vertices avoids potential cycling while listing the set of all vertices of the polytope $P$. The set of all such neighbours of given $v$ is denoted by $N(v)$ and can be found as in Algorithm NEIGHBOUR($v$).

**Algorithm 2.** Finding the set of neighbours for a given vertex, NEIGHBOUR($v$)

**Require:** $v \in P$ where $v = (\mathbf{w}, b, \boldsymbol{\xi})$; $\{\mathbf{x}_i, y_i\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^m, y_i \in \{+1, -1\}$

**procedure** NEIGHBOUR($v$)
  Initialize $N(v) \leftarrow \phi$
  Set $degenerate \leftarrow 0$
  Find $I^*(v)$, the set of active constraints at $v$ s.t.

$$y_i \mathbf{w} \cdot \mathbf{x}_i + y_i b + \xi_i = 1 \quad \forall i \in I^*(v) \,\&\, i \leq n$$
$$\xi_i = 0 \quad \forall i \in I^*(v) \,\&\, i > n$$

  **if** $|I^*(v)| > (m + n + 1)$ **then**
    $degenerate \leftarrow 1$
  **end if**
  Let $Leaving(v) := \{i \cup (i + n) \mid \xi_i = s_i = 0\}$
  Let $Incoming(v) := \{j \mid (s_j > 0 \,\&\, \xi_j = 0) \text{ or } (\xi_j > 0 \,\&\, s_j = 0)\}$
  **for all** $j \in Incoming(v)$ **do**
    **if** $degenerate = 0$ **then**
      **for all** $i \in Leaving(v)$ **do**
        $I^*(\tilde{v}) \leftarrow I^*(v) \setminus \{i\} \cup \{j\}$
        $N(v) \leftarrow N(v) \cup$ VERTEX($I^*(\tilde{v})$)
      **end for**
    **else**
      **for all** $S \subset Leaving(v)$ s.t $|S| = |I^*(v)| - (m + n)$ **do**
        $I^*(\tilde{v}) \leftarrow I^*(v) \setminus S \cup \{j\}$
        **if** $(det(A[I^*(\tilde{v}), ]) \neq 0)$ **then**
          $N(v) \leftarrow N(v) \cup$ VERTEX($I^*(\tilde{v})$)
        **end if**
      **end for**
    **end if**
  **end for**
  **return** $N(v)$
**end procedure**

### 2.3 Vertices of the Polytope, $P$

We observe that for $\lambda = 0$, the SVM QP is a linear program and its optimal solution by a simplex type algorithms will be at a vertex, say $v_0$, and the set of active constraints, $I^*(v_0)$, can be easily obtained. Intializing with this vertex $v_0$ and its active set $I^*(v_0)$, Algorithm 3 VERTEX SEARCH finds the set of all vertices of polytope $P$ using Algorithm NEIGHBOUR($v$)) as required, which in turn calls procedure VERTEX($Active$) with $I^*(v)$.

**Algorithm 3.** VERTEX SEARCH($P$)

Solve the SVM QP at (1) with $\lambda = 0$. Let this optimal classifier be $(\mathbf{w}^0, b^0, \boldsymbol{\xi}^0)$. This corresponds to a vertex, say $v_0 \in P$
Set $Current \leftarrow \{v_0\}, N(P) \leftarrow \phi$
**while** $(Current \neq \phi)$ **do**
    $Next \leftarrow \phi$
    **for all** $v \in Current$ **do**
        $Next \leftarrow Next \cup \text{NEIGHBOUR}(v)$
    **end for**
    $N(P) \leftarrow N(P) \cup Current$
    $Current \leftarrow Next \setminus N(P)$
**end while**
**return** $N(P)$

## 3 The Regularization Path

As the optimal classifiers for SVM QP are on the boundary of the polytope, by Theorem 1, the set of classifiers given by the regularization path is a subset of the set of vertices and related edges of the polytope of feasible region. Since the classifier is chosen by 0-1 loss function, using this in SVM design phase itself, one can argue that vertex classifiers on the path dominate those at the related edges.

However, for some set of test points, the dominating vertex classifier (as in Theorem 2) may or may not be on the path (see the example in Section 4). In the following discussion, we focus on the classifiers at vertices that generate some portions of the regularization path.

Before describing the procedure to identify the vertices on the path traced by the parameter $\lambda$, we mention a few results which will be used by this procedure.

Using the fact that, at optimality, the gradient of the objective function in a convex setting needs to be a member of the normal cone at that point and the KKT system gives an algebraic representation of this geometric phenomena, we have the following:

**Theorem 5.** *The bounds $[\lambda_l, \lambda_u]$ on the range of values of the regularization parameter $\lambda$ for which a given classifier $(\mathbf{w}, b, \boldsymbol{\xi}) \in P$ is optimal for SVM QP at (1), can be obtained as solutions to the following two linear programs, respectively:*

$$\lambda_l = \min_{\lambda \geq 0} \lambda \quad (5) \qquad \lambda_u = \max_{\lambda \geq 0} \lambda \quad (6)$$
$$over \quad S(\mathbf{w}, b, \boldsymbol{\xi}) \qquad over \quad S(\mathbf{w}, b, \boldsymbol{\xi})$$

*where* $S(\mathbf{w}, b, \boldsymbol{\xi}) = \{(\lambda, \alpha_1, \ldots, \alpha_n)\}$ *such that*

$$\begin{aligned}
\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i &= 2\lambda \mathbf{w} \\
\sum_{i=1}^{n} \alpha_i y_i &= 0 \\
\alpha_i \left[ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - \xi_{\mathbf{i}} \right] &= 0 \quad \forall i \in \{1, \ldots, n\} \\
(1 - \alpha_i)\xi_i &= 0 \quad \forall i \in \{1, \ldots, n\} \\
0 \leq \alpha_i &\leq 1 \quad \forall i \in \{1, \ldots, n\}
\end{aligned}$$

*Proof.* For $(\mathbf{w}, b, \boldsymbol{\xi})$ to be optimal for a given $\lambda$ in the SVM QP, the Karush-Kuhn-Tucker (KKT) system should hold with Lagrange multipliers $\{\alpha_i\}_{i=1}^n$. The set $S(\mathbf{w}, b, \boldsymbol{\xi})$ has been defined using the KKT system for the SVM QP. Hence the bounds on the range of $\lambda$ are given by the above two LPs. □

The next result says that a portion of the $\lambda$-path of a given SVM is partitioned by intervals corresponding to some of the vertices and edges of the polytope $P$. Also, we can see that the $\lambda$ value for an edge classifier is a harmonic mean of the bounds on the $\lambda$ interval of the related vertices.

**Theorem 6.** *(i) For a classifier* $(\mathbf{w}, b)$ *which is a vertex* $v := (\mathbf{w}, b, \boldsymbol{\xi})$ *on the polytope* $P$*, the range* $[\lambda_l, \lambda_u]$ *of* $\lambda$ *values, for which* $v$ *is optimal, is an interval in* $\mathbb{R}$*. In fact, this range is always finite since the gradient of the objective is never parallel to the generators of the normal cone.*

*(ii) For a classifier on an edge or a facet of* $P$*, the feasible* $\lambda$ *value is a singleton, i.e.,* $\lambda_l = \lambda_u$*. Specifically, for an edge point classifier,* $e_{v_1,v_2}$*, lying between two on-the-path vertices* $v_1$ *and* $v_2$ *such that* $\lambda_u(v_1) < \lambda_l(v_2)$*, we have*

$$\lambda(e_{v_1,v_2}) = \frac{\lambda_u(v_1)\lambda_l(v_2)}{\beta\lambda_l(v_2) + (1-\beta)\lambda_u(v_1)}. \tag{7}$$

*(iii) A portion of the regularization path corresponding to vertex-edge boundary of* $P$ *can be decomposed into intervals corresponding to vertices on the path and the edges between them. This is so because, for an edge point classifier,* $e_{v_1,v_2}$*, as described above, we have*

$$\lambda(e_{v_1,v_2}) \in (\lambda_u(v_1), \lambda_l(v_2)). \tag{8}$$

*Proof.* (i) For optimality we require that the gradient of the objective function, $\nabla f(x^*)$, of (1) to lie in the normal cone to the polytope at the optimum $x^*$. The normal cone at the vertex $v$ is a full dimensional polyhedral forward cone [**?**].

$$\nabla f(\mathbf{w}, b, \boldsymbol{\xi}) = \begin{pmatrix} 2\lambda w_1 \\ \vdots \\ 2\lambda w_m \\ 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \tag{9}$$

$$\text{and } \nabla g_i^1(\mathbf{w}, b, \boldsymbol{\xi}) = \begin{pmatrix} y_i x_{i,1} \\ \vdots \\ y_i x_{i,m} \\ y_i \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \quad \forall i \text{ of type } A1 \tag{10}$$

$$\nabla g_i^2(\mathbf{w}, b, \boldsymbol{\xi}) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \quad \forall i \text{ of type } A2 \tag{11}$$

where A1 and A2 are the coefficient matrices for the constraints of the SVM QP as defined in (**??**) and (**??**).
For an optimal classifier $(\mathbf{w}, b, \boldsymbol{\xi})$, it is required that the gradient of the objective function at this classifier can be expressed as a convex combination of the generators of the normal cone at this point, i.e.,

$$\nabla f(\mathbf{w}, b, \boldsymbol{\xi}) = \sum_{i \in I^*(\mathbf{w}, b, \boldsymbol{\xi})} \left[ \alpha_i \nabla g_i^1(\mathbf{w}, b, \boldsymbol{\xi}) + \tilde{\alpha}_i \nabla g_i^2(\mathbf{w}, b, \boldsymbol{\xi}) \right]. \tag{12}$$

where $0 \leq \alpha_i, \tilde{\alpha}_i \leq 1$ for all $i \in \{1, \ldots, n\}$. By equating the components, it can be easily verified that $\tilde{\alpha}_i = 1 - \alpha_i$ for all $i \in \{1, \ldots, n\}$.
Using (9), (10) and (11), it can be noted that the regularization parameter $\lambda$ impacts only the $\mathbf{w}$ component of the classifier $(\mathbf{w}, b, \boldsymbol{\xi})$ for optimality. And hence only some suitable scaling(s) of $\mathbf{w}$ via $\lambda$ lie in the normal cone generated above. The generators for the normal cone at the vertex $v$ span $\mathbb{R}^{m+n+1}$. This results in a finite interval of $\lambda$ for which the vertex is optimal for (1).

It is important here to note that $\nabla f(v)$ is not parallel to any generator of the normal cone, otherwise the there would be an infinite range of $\lambda$ for $v$ to be optimal, which is not generally the case. We prove this claim as below:
Suppose $\nabla f(v)$ is parallel to some generator $\nabla g_i^1(v)$ of the normal cone at vertex $v$, which implies,

$$\alpha_i > 0 \text{ and } \alpha_j = 0 \quad \forall j \in \{1, \ldots, n\} \setminus \{i\}. \tag{13}$$

If the above is true, then the KKT condition (**??**) is violated and hence $v$ cannot be optimal for any $\lambda$ value.
Also, looking at the components of the gradients at (9), (10) and (11), we can see that for no convex combination, $\nabla f(v)$ is parallel to the generators $\nabla g_i^1(v)$ or $\nabla g_i^2(v)$.

(ii) For a point on the edge $e_{v_1,v_2}$, which is the set of convex combinations of two vertices, the normal cone is a part of the orthogonal hyperplane to this edge. Hence there is a unique $\lambda$ for which $\nabla f(e_{v_1,v_2})$ intersects this hyperplane. Similarly, for classifiers on the facets of dimension $k < (m+n+1)$, the generators of the normal cone span $\mathbb{R}^{m+n+1-k}$ and only a unique suitable scaling of $\nabla f(x)$ lies in the $\mathbb{R}^{m+n+1}$ subspace.

(iii) The $\lambda$ value for an edge classifier on the path is bounded by the upper bound of one vertex and lower bound of the other. This is evident from the continuity of the $\lambda$ path between two $\lambda$-feasible vertices on the polytope $P$. Consider two neighbouring vertices, $v_1 = (\mathbf{w}^1, b^1, \boldsymbol{\xi}^1)$ and $v_2 = (\mathbf{w}^2, b^2, \boldsymbol{\xi}^2)$ lying adjacent on the $\lambda$-path with $\lambda$ intervals as $(\lambda_l(v_1), \lambda_u(v_1))$ and $(\lambda_l(v_2), \lambda_u(v_2))$ respectively, where $\lambda_u(v_1) < \lambda_l(v_2)$. Suppose $e_{v_1,v_2} = (\mathbf{w}^{e_{v_1,v_2}}, b^{e_{v_1,v_2}}, \boldsymbol{\xi}^{e_{v_1,v_2}})$ is a point on the edge between $v_1$ and $v_2$. Then $e_{v_1,v_2}$ can be expressed as:

$$e_{v_1,v_2} = \beta v_1 + (1-\beta)v_2 \quad \text{for some } \beta \in (0,1). \tag{14}$$

If $e_{v_1,v_2}$ is optimal for some $\lambda(e)$, then $(\mathbf{w}^e, b^e, \boldsymbol{\xi}^e, \lambda(e))$ should satisfy the KKT system for the SVM QP. Thus,

$$2\lambda(e_{v_1,v_2})\mathbf{w}^{e_{v_1,v_2}} = \sum_{i=1}^{n} \alpha_i(e_{v_1,v_2}) y_i \mathbf{x}_i \tag{15}$$

Now, since $v_1$ and $v_2$ are optimal for $\lambda_u(v_1)$ and $\lambda_l(v_2)$, they too satisfy the KKT system and hence we have the following relationships:

$$2\lambda_u(v_1)\mathbf{w}^1 = \textstyle\sum_{i=1}^{n} \alpha_i(v_1) y_i \mathbf{x}_i \tag{16}$$
$$2\lambda_l(v_2)\mathbf{w}^2 = \textstyle\sum_{i=1}^{n} \alpha_i(v_2) y_i \mathbf{x}_i \tag{17}$$

Substituting for $\mathbf{w}^1$ and $\mathbf{w}^2$ in (15) using the above two relations, we have

$$\sum_{i=1}^{n} \alpha_i(e_{v_1,v_2}) y_i \mathbf{x}_i = \lambda(e_{v_1,v_2}) \sum_{i=1}^{n} \left[ \frac{\beta\alpha_i(v_1)}{\lambda_u(v_1)} + \frac{(1-\beta)\alpha_i(v_2)}{\lambda_u(v_2)} \right] y_i \mathbf{x}_i \tag{18}$$

Comparing the components of the sum, we arrive at following relation:

$$\alpha_i(e_{v_1,v_2}) = \lambda(e_{v_1,v_2}) \left[ \frac{\beta\alpha_i(v_1)}{\lambda_u(v_1)} + \frac{(1-\beta)\alpha_i(v_2)}{\lambda_u(v_2)} \right] \quad \forall i = 1, 2, \ldots, n \quad (19)$$

Consider a point $\mathbf{x}_i$, $i \in \{1, \ldots, n\}$ such that $\alpha_i(v_1) = \alpha_i(v_2) = 1$. This implies $\alpha_i(e_{v_1,v_2}) = 1$ using the KKT system and the fact that $e_{v_1,v_2}$ is a convex combination of $v_1$ and $v_2$. Solving for $\lambda(e_{v_1,v_2})$ in (19) for such an $i$, we get

$$\lambda(e_{v_1,v_2}) = \frac{\lambda_u(v_1)\lambda_l(v_2)}{\beta\lambda_l(v_2) + (1-\beta)\lambda_u(v_1)} \quad (20)$$

Since, $\lambda_u(v_1) < \lambda_l(v_2)$, we get

$$\lambda_u(v_1) < [\beta\lambda_l(v_2) + (1-\beta)\lambda_u(v_1)] < \lambda_l(v_2) \quad \forall\beta \in (0,1), \quad (21)$$

and hence, $\lambda_u(v_1) < \lambda(e_{v_1,v_2}) < \lambda_l(v_2)$.

□

To trace the regularization path, we solve the SVM QP for $\lambda = 0$ which is a linear program. This gives us a classifier corresponding to a vertex in $P$, say $v_0$. The range of $\lambda$ for $v_0$ can be obtained via the solutions to the linear programs: (5) and (6). We know that $\lambda$ traces a continuous path along the boundary of $P$, so the next vertex on the path will be a neighbour of $v_0$, found using the procedure NEIGHBOUR($v$). Many of the neighbouring vertices are not optimal classifiers for any value of $\lambda$ and hence, the LPs (5) and (6) become infeasible at such neighbouring vertices. We will have one such neighbour for which there exists an interval of $\lambda$ and hence it becomes the next vertex on the $\lambda$-path. Then we search amongst the neighbours of the current vertex, to find the next vertex on the path. This procedure continues iteratively till all the neighbours of the current vertex become infeasible for the path. Such a vertex corresponds to the last but one vertex on the path.

Yet, there can be instances, as we will show in our example, where the path does not retain continuity along the vertex-edge boundary of the polytope. This happens when none of the neighbours of the current vertex are optimal for any value of $\lambda$. This forces us to search exhaustively for next generation neighbours which are optimal for some value of the parameter $\lambda$.

## 4 An Illustrative Example

The purpose of this example is to illustrate two aspects of the regularization path: a contiguous portion of the path composed of vertices and edges of the polytope, and another portion on facets of two or more dimensions. Interestingly, this portion is sandwiched between intervals generated by some vertices and edges. We tabulate the $\lambda$ intervals for the vertices of $P$ on the $\lambda$-path for a binary classification SVM model. We consider a training set with 50 points drawn from a bivariate normal distribution. The two classes have means $(0, 0)$ and $(1, 0)$ and

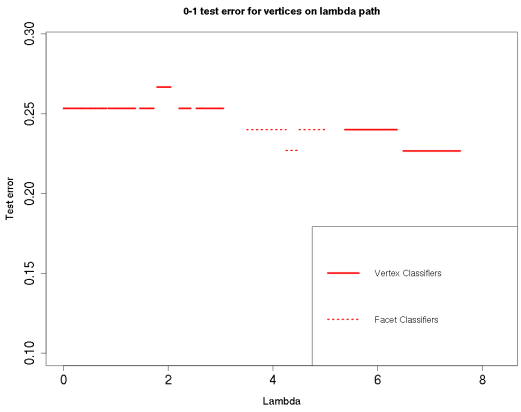


**Fig. 1.** Test error (averaged over 5 test data sets) for the $\lambda$-path for a bivariate normal data with 50 points

**Table 1.** $(\lambda_l, \lambda_u)$ for vertices on $\lambda$-path for a bivariate normal data with 50 points

| Vertex Index | $\lambda_l(v)$ | $\lambda_u(v)$ |
|---|---|---|
| 7274 ($v_0$) | 0 | 0.002712 |
| 7325 | 0.002776 | 0.059187 |
| 7327 | 0.059275 | 0.339041 |
| 7326 | 0.339110 | 0.522892 |
| 7328 | 0.523507 | 0.602838 |
| 7426 | 0.610561 | 0.745904 |
| 7420 | 0.749820 | 0.818734 |
| 7421 | 0.851777 | 1.191843 |
| 7424 | 1.198099 | 1.370998 |
| 7425 | 1.456195 | 1.723649 |
| 7352 | 1.785567 | 2.047029 |
| 9038 | 2.206312 | 2.428423 |
| 9158 | 2.535443 | 3.053205 |
| - | 3.5 | 3.5 |
| - | 4.25 | 4.25 |
| 10470 | 5.372196 | 6.368380 |
| 10471 | 6.486113 | 7.576469 |

same covariance matrix $(0.5, 0; 0, 0.5)$. A list of 15002 valid vertices with an index was generated using `rcdd` package [5] in R programming language.

The vertices are arranged so that the lower and upper bounds on the $\lambda$ intervals are in an increasing order. It was observed that each vertex on this path is a neighbour of the previous vertex on the list (except the first vertex, $v_0$). The portion of the path that occurs between vertices 9158 and 10470 corresponds to classifiers on the facets, since none of the first generation neighbours of vertex 9158 are optimal for any value of $\lambda$. The next optimal vertex is 10470 which may be obtained as a fifth generation neighbour to the vertex 9158.

Using Theorem 6, we note that only 15 of these 15002 vertices are on the $\lambda$-path. The set of first 13 vertices correspond to the first portion of the path, followed by a portion on the facets of two or higher dimensions. The last two vertices and the edge involving them correspond to the next segment of the path. As mentioned above, there are no more optimal vertices on the path.

We have plotted the test error of these classifiers in Figure 1, computed on 5 test data sets of 15 points with the same distribution as the training data set. From Figure 1, we can also see that the classifiers corresponding to the facets are dominated by the vertex classifiers in terms of test error. Hence, it is sufficient to consider the vertices corresponding to low values of parameter $\lambda$. The test error is lower for the last but one vertex on the path for the given test sets.

## 5 Discussion

The polyhedral structure of the feasible space of the standard SVM optimization model allows us to trace the $\lambda$-path on a subset of vertices of the base polytope. It was observed that for initial values, the $\lambda$-path comprises of vertices and related edges. We have examples where the path has classifiers that are on facets of the SVM polytope, $P$. We have restricted ourselves to the subset of vertices that dominate the whole polytope of feasible classifiers on 0-1 loss. The vertices and their neighbours can be identified via suitable active constraint sets. We noticed in our limited computational exercise, that the tracing of the $\lambda$-path has encountered numerical instabilities; such problems are also reported by [7].

Some aspects that naturally need attention include the need to come up with a scheme to pick that neighbour of a vertex which generates the adjacent interval on the $\lambda$-path, if such an interval exists. Perhaps, a more broader and important aspect is to be able to restrict ourselves to a suitable subset of vertices, which may or may not be on the $\lambda$-path, but have a promising test error.

A leave-one-out scheme [3] can be employed for testing the design of the classifier. Such leave-one-out training sets can be viewed as suitable perturbations of a given training set, and corresponding robust classification problems can be formalized. We can ensure the stability of the SVM algorithm by establishing such equivalence with a robust optimization formulation [13].

Besides the test error, some other measures such as bias, variance and the margin [4] of the classifier can be used for design. An analysis of error decomposition of the learning algorithms such as decision trees, $k$-NNs, etc. is done by [4], where the main consideration is for the algorithms that are consistent in the use of the same loss function for training as well as testing. This analysis was further taken up by [12] to the case of SVMs, where training makes use of hinge loss and testing is based on $0-1$ loss.

Some statistical properties about the risk of the classifier [14], [11] can be explored to improve the efficiency of our algorithms. These results may help us pick 'good' vertex classifiers, for example, via the $\lambda$ intervals corresponding to good generalizations guarantees.

## References


1. Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.
2. Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin. Training and testing low-degree polynomial data mappings via linear SVM. *The Journal of Machine Learning Research*, 11:1471–1490, 2010.
3. Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine learning*, 46(1):131–159, 2002.
4. Pedro Domingos. A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning. Stanford CA Morgan Kaufmann*, pages 231–238, 2000.

5. Charles J. Geyer, Glen D. Meeden, and incorporates code from cddlib written by Komei Fukuda. *rcdd: Computational Geometry*, 2015. R package version 1.1-9.
6. T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2001.
7. Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. *The Journal of Machine Learning Research*, 5:1391–1415, 2004.
8. Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of Convex Analysis*. Grundlehren Text Editions. Springer Berlin Heidelberg, 2004.
9. Pratik Jawanpuria, Manik Varma, and Saketha Nath. On p-norm path following in multiple kernel learning for non-linear feature selection. In *Proceedings of the 31st International Conference on Machine Learning*, pages 118–126, 2014.
10. R.Tyrrell Rockafellar. *Convex Analysis*. Princeton landmarks in mathematics and physics. Princeton University Press, 1997.
11. Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.
12. Giorgio Valentini and Thomas G Dietterich. Bias-variance analysis of support vector machines for the development of svm-based ensemble methods. *The Journal of Machine Learning Research*, 5:725–775, 2004.
13. Huan Xu, Constantine Caramanis, and Shie Mannor. Robustness and regularization of support vector machines. *The Journal of Machine Learning Research*, 10:1485–1510, 2009.
14. Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, pages 56–85, 2004.