Electronic Journal of Statistics
Vol. 15 (2021) 6311–6343
ISSN: 1935-7524
https://doi.org/10.1214/21-EJS1902

Geometric insights into support vector machine behavior using the KKT conditions*

Iain Carmichael

e-mail: idc9@uw.edu

J. S. Marron

e-mail: marron@unc.edu

Abstract: The *support vector machine* (SVM) is a powerful and widely used classification algorithm. This paper uses the Karush-Kuhn-Tucker conditions to provide rigorous mathematical proof for new insights into the behavior of SVM. These insights provide unexpected relationships between SVM and two other linear classifiers: the *mean difference* and the *maximal data piling direction*. For example, we show that in many cases SVM can be viewed as a cropped version of these classifiers. By carefully exploring these connections we show how SVM tuning behavior is affected by data characteristics including: balanced vs. unbalanced classes, low vs. high dimension, separable vs. non-separable data. These results provide further insights into tuning SVM via cross-validation by explaining observed pathological behavior and motivating improved cross-validation methodology.

MSC2020 subject classifications: Primary 62H99.
Keywords and phrases: High-dimensional classification, data piling, cross-validation.

Received October 2020.

1. Introduction

The *support vector machine* (SVM) is a popular and well studied classification algorithm (for an overview see Schölkopf, Smola 2002; Shawe-Taylor, Cristianini 2004; Steinwart, Christmann 2008; Mohri et al. 2012; Murphy 2012). Classical classification algorithms, such as *logistic regression* and *linear discrimination analysis* (LDA) are motivated by fitting a statistical distribution to the data. Hard margin SVM on the other hand is motivated directly as an optimization problem based on the idea that a good classifier should maximize the margin between two classes of separable data. Soft margin SVM balances two competing objectives; maximize the margin while penalizing points on the wrong side of the margin.

Interpretability, explainability, and more broadly understanding why a model makes its decisions are active areas of research in machine learning (Guidotti

*This research was supported in part by the National Science Foundation under Grant No. 1633074.

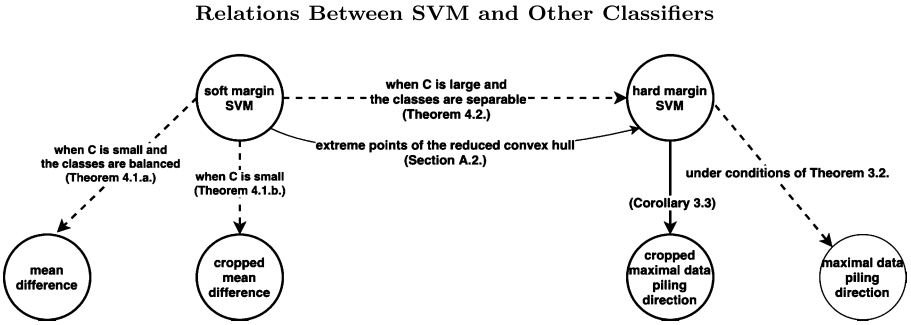


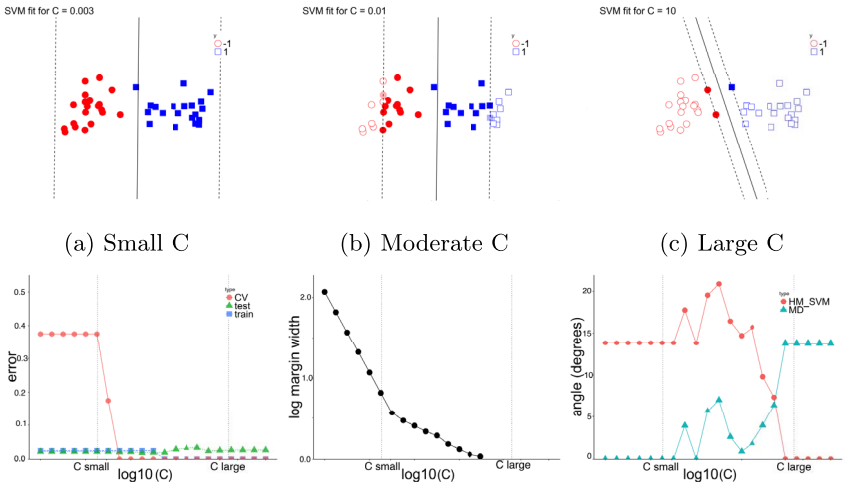
FIG 1. *SVM reduces to another classifier under the condition stated in the arrow. Solid line means the relation always holds. Dashed line means the relation may or may not hold depending on the data. For example, SVM reduces to the mean difference when the classes are balanced and C is sufficiently small ($C \leq C_{small}$) which is shown in Theorem 4.1.*

et al., 2018; Doshi-Velez, Kim, 2017). There is a large body of research providing theoretical guarantees and computational advances for studying SVM (Vapnik, 2013; Steinwart, Christmann, 2008). Several papers have shed some light on SVM by placing it in a probabilistic framework (Sollich, 2002; Polson et al., 2011; Franc et al., 2011). Here we take a different approach based on optimization and geometry, to understand the inner workings of SVM.

The main setting of this paper is the two class classification problem. We focus on linear classifiers, but the results extend to corresponding kernel classifiers. We consider a wide range of data analytic regimes including: high vs. low dimension, balanced vs. unbalanced class sizes and separable vs. non-separable data. Our results also extend to the multi-class case.

Using the *KKT conditions*, this paper demonstrates novel insights into how SVM's behavior is related to a given dataset and furthermore how soft-margin SVM's behavior is affected by the tuning parameter. We discover a number of connections between SVM and two other classifiers: the *mean difference* (MD) and *maximal data piling classifier* (MDP). These connections are summarized in Figure 1. In particular, when soft-margin SVM tuning parameter C is small, soft margin SVM behaves like a possibly *cropped* (see Section 2 below) MD classifier (Theorem 4.2). When the data are high dimensional, hard SVM (and soft margin with large C) behaves like a cropped MDP classifier (Theorem 3.2, Corollary 3.3, Theorem 4.2). The connection between SVM and the MD further implies connections between SVM, after a data transformation, and a variety of other classifiers such as *naive Bayes* (NB) (see Section 2.1). The connection between SVM and MDP provides novel insights into the geometry of the MDP classifier (Sections 3.1, A.1).

These insights explain several observed and surprising SVM behaviors, which motivated this paper (Section 1.1). These insights have immediate application to data analysis e.g. by improving SVM cross-validation methodology (Section 6). Online supplementary material including code to reproduce the figures and



(a) Small C (b) Moderate C (c) Large C

(d) Error rate (e) Margin (f) SVM vs. MD, HM-SVM

FIG 2. *(Balanced classes) The top rows show the SVM fit for three values of C. The bottom row shows diagnostics which are described in the text for a range of values of C. Figure 2d shows that the cross-validation error curve can be very different from the training and test error. Figure 2f shows that for small enough values of C, the SVM and MD directions are the same.*

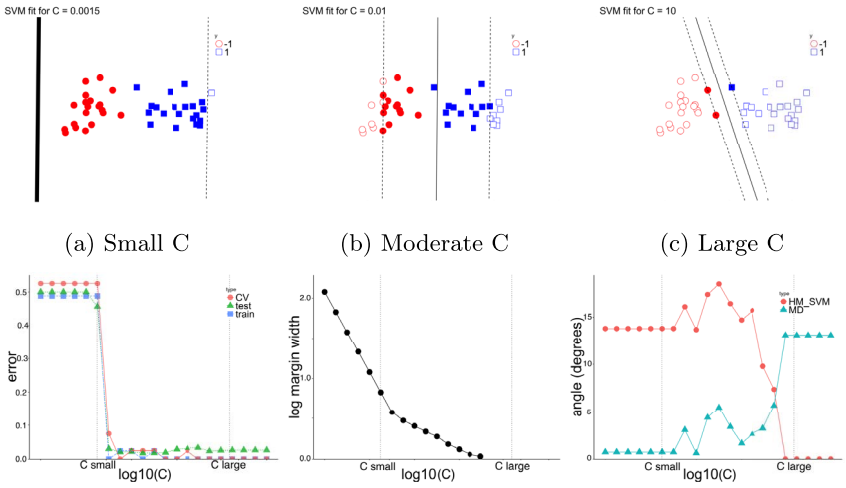
simulations can be found at: https://github.com/idc9/svm_geometry.

1.1. Motivating example

This section uses a simple, two dimensional example to demonstrate a number of instances of pathological or surprising SVM behavior, which the rest of the paper explains and builds on.

Figures 2 and 3 show the result of fitting soft-margin SVM for a range of tuning parameters. The data in both figures are generated from a two dimensional Gaussian with identity covariance such that the distance between the class means is 4. In Figure 2 the classes are balanced (20 points in each class). The data points in Figure 3 are the same points as the first figure, but one additional point is added to the positive class (blue squares) so the classes are unbalanced. In both cases the classes are linearly separable.

The top row of panels show the data along with the SVM separating hyperplane (solid line) for three different values of C. The *marginal hyperplanes* are shown as dashed lines and the filled in symbols are *support vectors*. The bottom three panels show various functions of C. The bottom left panel shows three error curves: training, cross-validation (5-folds), and test set error. The bottom middle panel shows the margin width. Finally, the bottom right panel shows the angle between the soft margin SVM direction and both the hard margin SVM



(a) Small C (b) Moderate C (c) Large C

(d) Error rate (e) Margin (f) SVM vs. MD, HM-SVM

FIG 3. *(Unbalanced classes). The panels are the same as in Figure 2, but the data now have one additional point added. When C is small, the top left panel shows SVM classifies every point to the larger class (the separating hyperplane is pushed past the smaller class). For this unbalanced example the cross-validation, train and test error all behave similarly, unlike the balanced case (compare Figure 3d to 2d). When C is small, the angle between SVM and the MD is small but not exactly zero (compare Figure 3f to 2f).*

direction and the mean difference direction. The vertical dashed lines indicate the values of C_{small} and C_{large} which are discussed below. See references above or Appendices C, D for definitions of the margin and support vectors.

Important features of these plots include:

1. For balanced classes (Figure 2), the training, cross-validation and test set error is low for most values of C, then suddenly shoots up to around 50% error for a small enough values of C (see Figure 2d). For balanced classes (Figure 2), this tuning error explosion for small C only happens for cross validation, not the tuning or test sets (see Figure 2d). This pathological behavior is concerning for a number of reasons. It demonstrates an example when performance with cross-validation may not reflect test set performance. Moreover, it is not clear why this behavior is happening.
2. Figure 2f shows that the SVM decision boundary can be parallel to the mean difference decision boundary when the data are balanced. This behavior is surprising because the SVM optimization problem is not immediately connected to the means of the two classes. Similarly, Figure 3f demonstrates an example when the SVM and MD decision boundaries are almost parallel for unbalanced classes.
3. Both Figures 2f and 3f show that soft margin SVM becomes exactly equivalent to hard margin SVM for some finite value of C when the data are

separable.

Theorem 4.1 gives a complete answer to why and when the first two of these behaviors occur. For the first example, if the data are unbalanced then the intercept term will always go off to infinity for small enough values of the tuning parameter. While SVM finds a good direction, its performance is betrayed by its intercept. For the second example, when C is smaller than a threshold value C_{small} (Definition 4.3), the SVM direction will be equivalent to the MD direction when the data are balanced. Similarly, when the data are unbalanced and $C < C_{\text{small}}$ the SVM direction is close to the MD direction. In this latter case, Equations (8), (9) show the SVM direction must satisfy constraints that make it a cropped mean difference direction.

This threshold C_{small} (Definition 4.3) governing when SVM behaves like the MD depends on diameter of the training data. Similarly, a threshold C_{large} (Definition 4.4) governing when soft margin SVM becomes hard margin SVM depends on the gap between the two training classes These two thresholding values are shown as dotted vertical lines in the bottom three panels of Figures 2 and 3.

Careful study of these behaviors, including the given formulas for the two thresholds, shows ways in which soft margin SVM's behavior can change depending on characteristics of the data including: balanced vs. unbalanced classes, whether $d \geq n - 1$, the two class diameter, whether the classes are separable and the gap between the two classes when they are separable. These results then lead to new insights into SVM tuning (Section 6).

1.2. Related literature

Hastie et al. (2004) show how to efficiently compute the entire SVM tuning path. While a consequence of their technical results shows that for small enough C, SVM behaves like the MD, they don't make the explicit connection to the MD classifier. For balanced classes they prove SVM is equivalent to the MD. For unbalanced we give a stronger, more specific characterization as a cropped MD (see Theorem 4.1 and Lemma 4.1). Additionally, they did not find the important, general threshold values C_{small} or C_{large} which depend on the diameter (gap) of the data which have useful consequences for cross-validation.

Connections between SVM and other classifiers have been studied before, for example, Jaggi (2014) studies connections between SVM and logistic regression with an L1 penalty.

The nu-SVM literature gives another perspective on SVM optimization (Schölkopf et al., 2000; Crisp, Burges, 2000; Bennett, Bredensteiner, 2000; Chen et al., 2005; Mavroforakis, Theodoridis, 2006; Barbero et al., 2015). The nu-SVM literature is focused on computation and there is not much overlap with our results.

SVM robustness properties have been previously studied (Schölkopf et al. 2000; Steinwart, Christmann 2008), however, the cropped MD characterization of SVM for small C appears to be new.

We find the gap and diameter (Definitions 4.2, 4.1) of the dataset are important quantities for SVM tuning. These quantities show up in other places in the SVM literature, for example, their ratio is an important quantity in statistical learning theory (Vapnik, 1999).

Some previous papers have suggested modifying SVM's intercept (Crisp, Burges, 2000). We suggest a particular modification (Section B) which addresses the *margin bounce* phenomena (Section 5.3).

SVM tuning has been extensively studied (Steinwart, Christmann, 2008, Chapter 11) with focus on: cheaply computing the full tuning path (Hastie et al., 2004), tuning kernel parameters (Sun et al., 2010), optimizing alternative metrics which attempt to better approximate the test set error (Chapelle, Vapnik, 2000; Ayat et al., 2005), providing default values for tuning parameters (Mattera, Haykin, 1999; Cherkassky, Ma, 2004), and empirical (Duan et al., 2003; Duarte, Wainer, 2017) as well as theoretical (Lin et al., 2002) study of tuning parameter selection. Our tuning results provide different kinds of insights whose applications are discussed in more detail in Section 6 and B.

2. Setup and notation

A linear classifier is defined via the *normal vector* to its discriminating hyperplane and an *intercept* (or *offset*). A key idea in this paper is to compare *directions* of linear classifiers. Comparing the direction between two classifiers means comparing their normal vector directions; we say two directions are equivalent if one is a scalar multiple of the other. Note that two classifiers may have the same direction, but lead to different classification algorithms (i.e. the intercepts may differ).

Suppose we have n labeled data points $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and index sets I_+, I_- such that $y_i = 1$ if $i \in I_+$, $y_i = -1$ if $i \in I_-$ and $\mathbf{x}_i \in \mathbb{R}^d$. Let $n_+ = |I_+|$ and $n_- = |I_-|$ be the class sizes. We consider linear classifiers whose decision function is given by

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b,$$

where $\mathbf{w} \in \mathbb{R}^d$ is the normal vector and $b \in \mathbb{R}$ is the intercept (classification rule $\text{sign}(f(x))$).

Given two vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^d$ we consider their *directions to be equivalent* if there exists $a \in \mathbb{R}, a \neq 0$ such that $a\mathbf{w} = \mathbf{v}$ (and we will write $\mathbf{w} \propto \mathbf{v}$).

In this paper we consider the following linear classifiers: hard margin SVM, soft margin SVM (which we refer to as SVM), mean difference (also called *nearest centroid*), and the maximal data piling direction.

For a linear classifier where $\mathbf{w} = \sum_{i=1}^n \alpha_i x_i$ for some weights α we say the linear classifier is *cropped* if some of the $\alpha_i = 0$ for some values of i. This is similar to the notion of a *trimmed mean* (Stigler, 1973).

Often linear classification algorithms can be extended to a wide range of nonlinear classification algorithms using the *kernel trick* (Schölkopf, Smola, 2002). We focus on the linear case, but our mathematical results extend to the kernel case.

2.1. Mean difference and convex classifiers

The *mean difference* (MD) classifier (sometimes referred to as *nearest centroid* classifiers) selects the hyperplane that lies half way between the two class means (Anderson, 1962; Tibshirani et al., 2002). The MD classifier is of course a special case of LDA in the case of spherical covariance matrices (Friedman et al., 2001). In particular the vector $\mathbf{w}_{md}$ is given by the difference of the class means

$$\begin{aligned}\mathbf{w}_{md} &:= \frac{1}{n_+}\sum_{i\in I_+}\mathbf{x}_i - \frac{1}{n_-}\sum_{i\in I_-}\mathbf{x}_i \\ &:= \bar{\mathbf{x}}_+ - \bar{\mathbf{x}}_-.\end{aligned} \tag{1}$$

We say a linear classifier is a *convex classifier* if its normal vector, $\mathbf{w}$, is given as the difference of points lying in the convex hulls of the two classes (i.e. $\mathbf{w} = \mathbf{c}_+ - \mathbf{c}_-$ where $\mathbf{c}_\pm \in \text{conv}(\{\mathbf{x}_i | i \in I_\pm\})$). We define *convex directions*, Cvx, to be the set of directions such a classifier can take.

Definition 2.1. *Let Cvx denote the set of all vectors associated with the directions that go between the convex hulls of the two classes i.e.*

$$Cvx := \{a\,(\mathbf{c}_+ - \mathbf{c}_-)\,|a \in \mathbb{R}, a \neq 0, \text{ and } \mathbf{c}_j \in conv(\{\mathbf{x}_i\}_{i\in I_j}), j = \pm\}.$$

The set Cvx may be all of $\mathbb{R}^d$ if, for example, the two convex hulls intersect. When the data are linearly separable Cvx is a strict subset of $\mathbb{R}^d$. This set of directions will play an important role in later sections.

2.2. Data transformation

It is common to transform the data before fitting a linear classifier (e.g. mean center variables then scale by their standard deviation). A number of classifiers can be viewed as either: apply a data transformation then fit a more simple classifier (such as MD) or as a distinct classifier. These classifiers include: *naive Bayes*, *linear discriminant analysis*, *nearest shrunken centroid*, *regularized discriminant analysis*, and more (Friedman et al., 2001).

For example, when $d < n - 1$ the linear discriminant direction is given by

$$\mathbf{w}_{lda} := \widehat{\Sigma}_{pool}^{-1}(\bar{\mathbf{x}}_+ - \bar{\mathbf{x}}_-), \tag{2}$$

letting X_- and X_+ be the data matrix for the respective classes and the *pooled sample covariance* is $\widehat{\Sigma}_{pool} := \frac{1}{n-2}\big[(X_+ - \overline{X}_+)^T(X_+ - \overline{X}_+) + (X_- - \overline{X}_-)^T(X_- - \overline{X}_-)\big]$. Note the inevitability of $\widehat{\Sigma}_{pool}$ plays an important role in the next section. LDA is equivalent to transforming the data by the pooled sample covariance matrix (i.e. multiplied each data point by $\widehat{\Sigma}_{pool}^{-1/2}$) then computing the MD classifier.

The technical results of this paper connect SVM to MD (and various other convex classifiers), however, they apply more generally. If the analyst first transforms the data before fitting SVM, as is common in practice, then our results

connect SVM to the more general classifier. For example, naive Bayes is equivalent to first transforming the data by a certain diagonal covariance matrix; in this case, our results connect SVM to naive Bayes.

2.3. Maximal data piling direction

For linear classifiers one frequently projects the data onto the one dimensional subspace spanned by the normal vector. *Data piling*, first discussed by Marron et al. (2007), is when multiple points have the same projection on the line spanned by the normal vector. For example, all points on SVM's margin have the same image under the projection map. Ahn, Marron (2010) showed that when $d \geq n-1$ there are directions such that each class is projected to a single point i.e. there is *complete data piling*.

Definition 2.2. *A vector* $\mathbf{w} \in \mathbb{R}^d$ *gives complete data piling for two classes of data if there exist* $a, b \in \mathbb{R}$*, with* $a \neq 0$ *such that*

$$\mathbf{w}^T \mathbf{x}_i = a y_i + b \text{ for each } i = 1, \ldots, n,$$

where b *is the midpoint of the projected classes and* a *is half the distance between the projected classes.*

The *maximal data piling* (MDP) direction, as its name suggests, searches around all directions of complete data piling and finds the one that maximizes the distance between the two projected class images (Ahn et al., 2012; Lee et al., 2013; Ahn, Marron, 2010). The MDP direction takes an analytical form,

$$\mathbf{w}_{mdp} = \widehat{\Sigma}^{-}(\bar{\mathbf{x}}_+ - \bar{\mathbf{x}}_-), \tag{3}$$

where A^- is the Moore-Penrose inverse of a matrix A and $\widehat{\Sigma} := \frac{1}{n-1}(X - \bar{X})^T(X - \bar{X})$ is the *global sample covariance* matrix.

The MDP direction has an interesting relationship to LDA. Recall the formula for LDA show in Equation (2) above. Ahn, Marron (2010) showed that in low dimensional settings LDA and the MDP formula are the same (though in low dimensional settings MDP does not give complete data piling); when $d < n-1$ the above two equations are equivalent.

Another view of this relation comes from the optimization perspective. LDA attempts to find the direction that maximizes the ratio of the projected "between-class variance to the within-class variance," (Bishop, 2006). This problem is well defined only in low dimensions; in high dimensions when $d \geq n-1$ there exist directions of complete data piling where the within class projected variance is zero. In the high dimensional setting MDP searches around these directions of zero within class variance to find the one that maximizes the distance between the two classes (i.e. the between-class variance).

2.4. Support vector machine

Hard margin support vector machine is only defined when the data are linearly separable; it seeks to find the direction that maximizes the margin separating the two classes. It is defined as the solution to the following optimization problem,

$$\begin{aligned} &\underset{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}}{\text{minimize}} && \frac{1}{2}||\mathbf{w}||^2 \\ &\text{subject to} && y_i(\mathbf{x}_i^T\mathbf{w}+b) \geq 1, \text{ for } i=1,\ldots,n. \end{aligned} \tag{4}$$

The marginal hyperplanes are given by $\{\mathbf{x}|\mathbf{w}^T\mathbf{x} = \pm 1\}$ and no training points lie strictly between the marginal hyperplanes for hard-margin SVM. The margin width ρ is the orthogonal distance from the marginal hyperplanes to the separating hyperplane.

When the data are not separable Problem (4) can be modified to give soft margin SVM by adding a tuning parameter C and slack variables ξ_i which allow points to be on the wrong side of the marginal hyperplanes,

$$\begin{aligned} &\underset{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}}{\text{minimize}} && \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i \\ &\text{subject to} && y_i(\mathbf{x}_i^T\mathbf{w}+b) \geq 1-\xi_i, \text{ for } i=1,\ldots,n \\ &&& \xi_i \geq 0, \text{ for } i=1,\ldots,n. \end{aligned} \tag{5}$$

For soft-margin SVM, the marginal hyperplanes/width are defined in the same way as above (see the dashed lines in Figures 2 and 3). In both cases the direction is a linear combination[1] of the training data points

$$\mathbf{w}_{svm} = \sum_{i\in I_+} \alpha_i \mathbf{x}_i - \sum_{i\in I_-} \alpha_i \mathbf{x}_i.$$

Points which receive non-zero weights are called *support vectors*. It can be checked that points lying strictly outside of the space between the marginal hyperplanes receive zero weight. For hard-margin SVM all support vectors lie directly on a marginal hyperplane. For soft-margin SVM, support vectors can lie either directly on a marginal hyperplane or strictly within the marginal hyperplanes; we denote the former by *margin vectors* and the latter by *slack vectors*. See Mohri et al. (2012) or sections C.2 and D for a more detailed discussion.

3. Hard margin SVM in high dimensions

This section provides novel insights into the geometry of complete data piling which are then used to characterize the relationship between hard margin SVM and MDP in high dimensions. The results are stated below then proved in Appendix C.

[1]It turns out this linear combination always gives a direction that points between the convex hull of the two classes (see Definition 2.1).

For this section we assume $d \geq n-1$ and the data are in general position and separable, which implies the data are linearly independent if $d \geq n$ and affine independent if $d = n-1$.

3.1. Complete data piling geometry

Define the set P of *complete data piling directions* using ideas from Definition 2.2.

Definition 3.1. *Let P denote the vectors associated with directions that give complete data piling i.e.*

$$P := \{\mathbf{v} \in \mathbb{R}^d | \exists a, b \in \mathbb{R}, a \neq 0 \text{ s.t. } \mathbf{v}^T \mathbf{x}_i = a \cdot y_i + b \text{ for each } i = 1, \ldots, n\}.$$

Note the set of complete data piling directions can be empty, however, if the data are in general position then $P \neq \emptyset$ when $d \geq n-1$. In this case, Ahn, Marron (2010) point out there are infinitely many of such directions in the n dimensional subspace generated by the data that give complete data piling; in fact there is a great circle of directions in this subspace (if we parameterize directions by points on the unit sphere).

Theorem 3.1 shows there is a single complete data piling direction that is also within the ($n-1$ dimensional) affine hull of the data. The remaining directions in P are linear combinations of this unique direction in the affine hull and any vector normal to that hull.

Theorem 3.1. *The set of complete data piling directions, P, intersects the affine hull of the data in a single direction which is the maximal data piling direction.*

Theorem 3.1 is proved in Appendix C.1.

3.2. Hard margin SVM and complete data piling

A simple corollary of Theorem 3.1 is:

Corollary 3.1. *The intersection of the convex directions, Cvx, and the complete data piling directions, P, is either empty or a single direction i.e.*

$$Cvx \cap P = \emptyset \text{ or } Cvx \cap P = \{a\mathbf{v} | a \in \mathbb{R}\}.$$

If a convex classifier gives complete data piling then it has to also be the MDP; furthermore, there can be at most one convex classifier which gives complete data piling.

The core results for hard margin SVM are summarized in the following theorem. Note that this theorem also characterizes when SVM has complete data piling.

Theorem 3.2. *The hard margin SVM and MDP directions are equivalent if and only if there is a non-empty intersection between the convex directions, C, and the complete data piling directions, P. In this case, the intersection is a single direction which is the hard margin SVM direction and the MDP direction i.e.*

$$\mathbf{w}_{hm-svm} \propto \mathbf{w}_{mdp} \iff P \cap C \neq \emptyset \iff \mathbf{w}_{hm-svm} \propto \mathbf{w}_{mdp} = C \cap P.$$

We use the equality sign to indicate $C \cap P$ is a single direction. Theorem 3.2 is a consequence of Corollary 3.1, Lemma C.1, Lemma C.2 and the KKT conditions which are all provided in C. Appendix E gives an alternate characterization of the event $P \cap C \neq \emptyset$ through a linear program. As a corollary of this theorem we can characterize when MD/MDP or SVM/MD are equivalent.

Corollary 3.2. *The hard margin SVM and MD directions are equivalent if and only if all three of hard margin SVM, MD and MDP are equivalent i.e.*

$$\mathbf{w}_{hm-svm} \propto \mathbf{w}_{md} \iff \mathbf{w}_{md} \propto \mathbf{w}_{mdp}.$$

Another corollary of this theorem is that hard margin SVM is always the MDP of the support vectors.

Corollary 3.3. *Let V be the set of support vectors for hard margin SVM, then $\mathbf{w}_{hm-svm}$ is the MDP of V.*

This corollary says that we can interpret hard margin SVM as a cropped MDP (i.e. it ignores points which are far away from the separating hyperplane).

4. Soft margin SVM small and large C regimes

This section characterizes the behavior of SVM for the small and large regimes of the cost parameter C. We make no assumptions about the dimension of the data d. The main results for the small and large C regimes are provided in this section while the KKT conditions and proofs are provided in Appendix D.

We first make two geometric definitions that play an important role in characterizing SVM's tuning behavior. The two class *diameter* measures the spread of the data.

Definition 4.1. *Let the two class diameter be*

$$D := \max_{\mathbf{x}_+ \in I_+, \mathbf{x}_- \in I_-} ||\mathbf{x}_+ - \mathbf{x}_-||.$$

Definition 4.2. *Let the two class gap G be the minimum distance between points in the convex hulls of the two classes i.e.*

$$G := \min_{\mathbf{c}_j \in conv(\{\mathbf{x}_i\}_{i \in I_j})} ||\mathbf{c}_+ - \mathbf{c}_-||.$$

Using the above geometric quantities we define two threshold values of C which determine when the SVM enters its different behavior regimes.

Definition 4.3. *For two classes of data let*

$$C_{small} := \frac{2}{\max(n_+, n_-)D^2}, \tag{6}$$

where D is the diameter of the training data.

Definition 4.4. *If the two data classes are linearly separable let*

$$C_{large} := \frac{2}{G^2}, \tag{7}$$

where G is the gap between the classes.

As illustrated in Figures 2 and 3, the main result for the small C regime is given by Theorem 4.1 and Corollary 4.1. We call the support vectors lying strictly within the margin *slack vectors* (see Section 2.4 or Definition D.2).

Theorem 4.1. *When every point in the smaller (negative) class is a slack vector,*

a. If the classes are balanced then the SVM direction becomes the mean difference direction i.e. $\mathbf{w}_{svm} \propto \mathbf{w}_{md}$.
b. If the classes are unbalanced then the SVM direction satisfies the constraints in Equations (8), (9) making it a cropped mean difference.

$$\mathbf{w}_{svm} = \sum_{i \in M_+} \alpha_i \mathbf{x}_i + C \sum_{i \in L_+} \mathbf{x}_i - C \sum_{i \in I_-} \mathbf{x}_i, \tag{8}$$

subject to

$$\sum_{i \in M_+} \alpha_i = C(|L_+| - n_-). \tag{9}$$

Furthermore, $C < C_{small}$ is a sufficient condition such that every point in the smaller class is a slack vector.

Theorem 4.1 characterizes a kind of cropped mean difference. The mean difference direction points between the mean of the first class and the mean of the second class. Recall $\mathbf{w}_{svm}$ always goes between points in the convex hulls of the two classes. Equation (8) says that in the small C regime $\mathbf{w}_{svm}$ points between the mean of the smaller (negative) class (the third term) and a point that is close to the mean in the larger (positive) class. The cropping happens by ignoring non-support vectors. While points on the margin do not necessarily receive equal weight, Equation (9) bounds the amount of weight put on points on margin points. Note Equations (8), (9) are stronger than the simple constraint that $\sum_{i \in I_+} \alpha_i = n_- C$ (Lemma 2 from Hastie et al. (2004)) since all of the slack vectors in the positive class receive the same weight.

This theorem suggests that the mean difference can be used to warm start the computation of the SVM tuning path for small values of C. A similar idea is discussed in Hastie et al. (2004). Note that our theorem suggests that the mean difference may be a good approximation of the initial SVM value for small C

that is cheaper to compute than the Quadratic program discussed in Section 3 of Hastie et al. (2004).

Lemma 4.1 strengthens Lemma 4.1 in the case $n_+ \gg d$ i.e. there can't be too many margin vectors in Equation (8).

Lemma 4.1. *If the data are in general position the larger class can have at most* $n_- + d - 1$ *support vectors.*

As C continues to shrink past C_{small} the margin width continues to grow. Eventually the separating hyperplane will be pushed past the smaller class and every training point will be classified to the larger class (see Figure 3d).

Corollary 4.1. *If the classes are unbalanced and* $C < \frac{1}{2}C_{small}$ *then every training point is classified to the larger (positive) class.*

If the data are separable then in the large C regime soft margin SVM becomes equivalent to hard margin SVM for sufficiently large C.

Theorem 4.2. *If the training data are separable then when* $C > C_{large}$*, soft margin SVM is equivalent to the hard margin SVM solution i.e.* $\mathbf{w}_{svm} = \mathbf{w}_{hm-svm}$.

Note that C_{small} and C_{large} are lower and upper bounds—their respective limiting behavior may happen for C larger that C_{small} and C smaller than C_{large}. In practice, these threshold values are a reasonable approximation. Furthermore, the $\frac{1}{D^2}$ scale is important for small values of C (this can be seen in the proofs of Corollary D.1 and Lemma D.3)

5. Discussion of SVM regimes

For sufficiently small values of C, SVM is related to the mean difference. When the data are separable, for sufficiently large values of C soft margin SVM is equivalent to hard margin SVM. We note this discussion applies more broadly than just binary, linear SVM. For example, when a kernel is used SVM is related to the kernel mean difference classifier.

5.1. Small C regime and the mean difference

For sufficiently small C (when every point in the smaller class is a slack vector) Theorem 4.1 shows how soft margin SVM is related to the mean difference.

If the data are unbalanced then the SVM direction becomes a cropped mean difference direction as characterized by Equations (8), (9). The direction points from the mean of the smaller class to a cropped mean of a subset of points in the larger class. The cropped mean of the larger class gives equal weight to slack vectors, puts smaller weight on margin vectors and ignores points that are outside the margin (non-support vectors). Furthermore, the number of margin vectors is bounded by the dimension when the data are in general position (Lemma 4.1).

In the small C regime, if the data are balanced then the SVM direction becomes exactly the mean difference direction. Note Lemma 1 from Hastie et al. (2004) proves this result for balanced classes, proves a weaker version in the unbalanced case, does not give the threshold C_{small}, and does not discuss the connection between SVM and the MD classifier.

The lower bound C_{small} is important because it shows SVM's MD like behavior applies for every dataset set. Furthermore, it shows that the value of C where the MD like behavior begins depends on the data diameter and class sizes $\left(\text{i.e. is proportional to } \frac{1}{\max(n_+,n_-)D^2}\right)$. This dependence on the data diameter has important consequences for cross-validation which are discussed in Section 6.

Note the cropped MD interpretation is often valid for a wide range of C (i.e. values of C larger than C_{small}). In particular, as C shrinks, more vectors become slack vectors receiving equal weight (see proofs and results in Section 4). As C shrinks to C_{small}, the angle between SVM and the cropped MD defined in Theorem 4.1 approaches zero. This can be seen, for example, in Figure 3f.

Finally, note that the relation between SVM and the MD also relates SVM to a larger set of classifiers by taking data transformation into account (see Section 2.2). It is common to apply a transformation to the data before fitting SVM (e.g. mean centering then scaling by some covariance matrix estimate). In this case, the small C regime of SVM will be a (cropped) version of the transformed MD classifier. This insight connects SVM to, for example, the naive Bayes classifier. A similar connection can be made between SVM and LDA when the data are sphered by the inverse covariance matrix. Similarly, our results also connect kernel SVM to the kernel (cropped) MD classifier.

SVM's MD behavior discussed in this section raises the question of how much performance gain SVM achieves over (robust, transformed) mean difference classifiers. This is discussed more in Section A.3.

5.2. Class imbalance and the MD regime

Theorem 4.1 gives some insights into SVM when the classes are imbalanced. When SVM is in the MD regime as discussed above (i.e. $C \leq C_{\text{small}}$), every point in the smaller (negative) class has to be a support vector receiving equal weight. In some scenarios the MD or a cropped MD may perform very well. However, this result says in the small C regime, SVM cannot crop the smaller class (it can still crop the smaller class when $C > C_{\text{small}}$). This insight can explain some scenarios where SVM performs well for small values of C, but then its performance suddenly degrades for even smaller values of C (i.e. an outlier is forced into the smaller class's slack vectors).

Lemma 4.1 says that (under weak conditions) the larger (positive) class can have at most $n_- + d + 1$ support vectors (n_- = size of the smaller class). In the case $n_+ \gg n_-, d$ then SVM can only use a small number of data points from the larger class to estimate the SVM direction (this is true for all values of C). This means SVM is forced to do a lot of cropping for the larger (positive) class

which may be a good thing in some scenarios (i.e. if the larger class has many outliers).

5.3. Small C regime and margin bounce

As C shrinks, the margin (distance between the marginal hyperplanes) increases. When the classes are unbalanced, the marginal hyperplane of the larger class has to stay within the convex hull of the larger class causing the separating hyperplane to move off to infinity. For small enough values of C ($\leq \frac{1}{2}C_{\text{small}}$), this means the separating hyperplane is pushed past the smaller class and every point is classified to the larger class (Corollary 4.1). We call this behavior *margin bounce* (see Figure 3a for an example). In other words, for small values of C, SVM picks a reasonable direction, but a bad intercept.

When the classes are exactly balanced, the margin bounce may or may not happen (we have seen data examples of both). It would be an interesting follow up question to determine conditions for when the margin bounce happens for balanced classes.

This insight has a few consequences.

1. For Figure 3d (unbalanced classes) it explains why the three tuning error curves are large for small values of C.
2. For Figure 2d (balanced classes) it explains why only the cross-validation error curve is bad for small values of C, but the tuning and test set error curves are fine (i.e. the cross-validation training sets are typically unbalanced).
3. For small values of C SVM picks a bad intercept, but a fine direction. We exploit this fact in Section B to develop an improved intercept for SVM
4. The value of C when the margin starts exploding depends on the diameter of the two classes. This has important implications for cross-validation which are discussed in Section 6.

5.4. Large C regime and the hard-margin SVM

If the data are separable, Theorem 4.2 says that for sufficiently large values of C, soft margin SVM will be equivalent to hard margin SVM. Note that in high-dimensions (i.e. $d > n$) the data are always separable. If the original dataset is non-separable, but a kernel is used the transformed dataset may be separable (for example, if the implicit kernel dimension is larger than n).

Furthermore, the value of C above which soft-margin SVM becomes equivalent to hard margin SVM depends on the gap between the two classes (see Definition 4.2). This can have important consequences for cross-validation as discussed in Section 6.

5.5. Hard-margin SVM and the (cropped) maximal data piling direction

In high dimensions, (i.e. $d \geq n-1$) Theorem 3.2 gives geometric conditions for when hard margin SVM gives complete data piling i.e. when the SVM direction is equivalent to the MDP direction. Hard margin SVM always has some data piling; support vectors in the same class project to the same point. In this case SVM is the MDP direction of the support vectors. In this sense, hard margin SVM can be viewed as a cropped MDP direction where points away from the margin are ignored.

Complete data piling is a strict constraint and the SVM normal vector can usually wiggle away from the MDP direction to find a larger margin. This raises the question: is complete data piling with hard margin SVM a probability zero event when the data are generated by an absolutely continuous distribution? We suspect the answer is no: it occurs with positive, but typically small probability. For example consider three points in $\mathbb{R}^2$.

Often data piling may not be desirable e.g. the normal vector may be sensitive to small scale noise artifacts Marron et al. (2007). Additionally, the projected data have a degenerate distribution since multiple data points lie on top of each other. However there are cases, such as an autocorrelated noise distribution, when the maximal data piling direction performs well, Miao (2015).

Corollary 3.3 (SVM is the MDP of the support vectors) also gives an alternative characterization of hard margin SVM. Hard margin SVM searches over every subset of the data points which have a nonempty set of complete data piling directions, computes the MDP of each such subset, and selects the direction giving the largest separation. This characterization is mathematically interesting because it says we can *a priori* restrict the hard margin SVM optimization problem, Equation (4), to search over a finite set of directions (i.e. the complete data piling directions of the subsets of the data). Furthermore, in some cases, the MDP (Equation (3)) can be cheaply computed or approximated. For example, the analyst may use a low rank approximation to $\widehat{\Sigma}^{-}$ and/or select a judicious subset of data points. In these scenarios, it may make sense to approximate hard margin SVM with the MDP.

5.6. Multiple classes

Often multi-class classification problems are reduced to a number of binary class problems e.g. using *one vs. one* (OVO) or *one vs. all* (OVA) schemes (Friedman et al., 2001). Our results apply to each of these binary classification problems. For example, in a multi-class problem, even if the classes are roughly balanced, the OVA scheme may produce unbalanced classes where the behavior discussed in Section 5.2 becomes applicable.

6. Insights into tuning SVM via cross-validation

Tuning SVM using cross-validation means attempting to estimate the tuning curve of the test set (the green line marked with triangles in Figures 2d, 3d) using the tuning curve from cross-validation (the red line marked with circles). It is known that the optimal hyper-parameter settings for the full training set (of size n) may differ from the optimal settings for the cross-validation sets (of size $\left(1-\frac{1}{k}\right)n$); for example, the smaller dataset often favors larger values of C (more regularization) Steinwart, Christmann (2008).

The results of this paper give a number of insights into how features of the data cause the cross-validation tuning curve to differ from the test set tuning curve. In particular, we have shown that the tuning curve is sensitive to: (1) balanced vs. unbalanced classes (2) the two class diameter D (3) whether or not the classes are separable, (4) whether or not $d \geq n-1$ (5) the gap between the two classes G.

Each of these characteristics can change between the full training set and the cross-validation training sets. When the characteristics change, so can SVM's behavior for small and large values of C. Therefore SVM may behave differently for the cross-validation folds than for the full training data.

One dramatic example of this change in behavior can be seen in Figure 2d as discussed in Sections 5.3, and 1.1. In this case, the full dataset is balanced, but the cross-validation folds are typically unbalanced.

Another example of tuning behavior differences between the training and cross-validation data can be seen by looking carefully at Figure 3d. In this figure we can see the cross-validation error rate shoots up for larger values of C than the train/test error rates. The error increases dramatically for small values of C because of the margin bounce phenomena discussed in Section 5.3. The value of C_{small} that guarantees this behavior is a function of the two class diameter D (see Definition 4.3). Since there are fewer points in the cross-validation training set, the diameter is smaller meaning the value of C_{small} is larger causing the margin to explode for larger values of C.

Different data domains in terms of $n \ll d, n \sim d$, and $n \gg d$ can make the above characteristics more or less sensitive to change induced by subsampling. For example, if $n \gg d$ then subsampling is least likely to change whether $d \geq n-1$ or significantly modify the diameter D. With a kernel, however, even if the original $n \gg d$ then it may no longer be true that $n \gg d_{\text{implicit}}$ where d_{implicit} is the dimension of the implicit kernel space. An interesting, possible exception to this was given by Rahimi, Recht (2008) where d_{implicit} may be small.

When n is larger than d, but not by much, then subsampling is likely to change whether or not $d \geq n-1$ and whether or not the data are separable. In this case the full training data may not be separable, but the cross-validation sets may be. This means large values of C will cause soft margin SVM to become hard margin SVM for cross-validation, but never for the full training data. This could result in the SVM direction being very different between cross-validation and training.

When $d \geq n-1$ soft margin SVM will become hard margin SVM for $C \geq$

C_{large} which depends on the gap G between the two classes. Subsampling the data will cause this gap to increase meaning C_{large} decreases. In this case the hard margin behavior will occur for smaller values of C in the cross-validation sets than for the full training set.

It is desirable to perform cross-validation in a way that is least likely to change some of the above characteristics between the full and the cross-validation training data set. For example,

- If the full training data are balanced one should ensure the cross-validation training classes are also balanced.
- Cross-validation with a large number of folds (e.g. leave one out CV) is least likely to modify the above characteristics of the data.
- When $n > d$ it could be judicious to ensure $n_{cv} > d$ for each cross-validation set.
- Chapelle, Vapnik (2000) (Section 4) suggests re-scaling the data using the covariance matrix. The analyst may modify this idea by additionally rescaling each cross-validation training set such that the diameter is (approximately) the same as the diameter of the full training set.
- Previous papers have proposed default values for C based on the given dataset Mattera, Haykin (1999); Cherkassky, Ma (2004). Our results suggest other default values in the interval $[C_{\text{small}}, C_{\text{large}}]$ (when the latter exists) may be reasonable. Furthermore, default values which lie in the middle of this range may be preferable. For example, the analyst may try a simple MD classifier (producing similar results to a small C), one moderate and one large value of C for SVM.

Section B demonstrates how these insights can be applied to develop an better SVM intercept for cross-validation.

Appendix A: Additional discussion

A.1. Geometry of complete data piling

Theorems 3.1 and 3.2 give further insight into the geometry of complete data piling directions. In this section we consider directions to be points on the unit sphere; the equivalence class of a single direction is represented by two antipodal points.

When $d \geq n$ there are an infinite number of directions P that give complete data piling. If we restrict ourselves to the n dimensional subspace generated by the data there are still an infinite number of directions that give complete data piling Ahn, Marron (2010); within this subspace P forms a great circle of directions. Theorem 3.1 says that if we further restrict ourselves to the $n-1$ dimensional affine hull of the data there is only a single direction of complete data piling and this direction is the maximal data piling direction. The aforementioned great circle of directions intersects the subspace parallel to the affine hull of the data at two points (i.e. a single direction).

Note Equation (3) shows $\mathbf{w}_{mdp}$ is a linear combination of the data and Theorem 3.1 shows furthermore that $\mathbf{w}_{mdp}$ an affine direction. Finally, Theorem 3.2 also characterizes the stronger condition when the MDP is a convex classifier (see Section 2.1) i.e. when the MDP direction points between the convex hulls of the two classes ($\mathbf{w}_{mdp} \in C$).

A.2. nu-SVM and the reduced convex hull

A number of papers look at an alternative formulation of the SVM optimization problem (so called nu-SVM). These papers give an interesting, geometric perspective that characterizes soft margin SVM in terms of hard margin SVM (see citations in Section 1.2).

Recall the convex hull of a set of points is given by $H(\{\mathbf{x}_i\}_{i=1}^n) := \{\sum_{i=1}^m \lambda_i \mathbf{x}_i | \sum_{i=1}^n \lambda_i = 1, \lambda_i \geq 0\}$. Suppose we decrease the upper bound on the coefficients such that $\lambda_i \leq c$ for some $c \geq 0$. Define the *reduced convex hull* (RCH) as

$$R_c(\{\mathbf{x}_i\}_{i=1}^n) := \left\{ \sum_{i=1}^n \lambda_i \mathbf{x}_i | \sum_{i=1}^n \lambda_i = 1, \lambda_i \leq c \right\}$$

Note $R_c \subseteq H$, $R_c = H \iff c = 1$ and $c = \frac{1}{n} \iff R_c = \{\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\}$ (i.e. a single point). Also note that, R_c is not necessarily a dilation of H e.g. see Figure 5 from Bennett, Bredensteiner (2000) for an example. Furthermore, define E_c to be the set of *extreme points* of R_c (the RCH of a finite set of points is a polytope and the extreme points are the vertices of this polytope).

Similarly to Definition 2.1 of the convex directions for two classes, we define the set of *reduced convex directions*, RC_c

Definition A.1. *Let* $0 \leq c \leq \min\left(\frac{1}{n_+}, \frac{1}{n_-}\right)$ *and let* RC_c *denote the set of all vectors associated with the directions that go between the* c *reduced convex hulls convex hulls of the two classes i.e.*

$$RC_c = \{a\,(\mathbf{c}_+ - \mathbf{c}_-)\,| a \in \mathbb{R}, a \neq 0, \text{ and } \mathbf{c}_j \in R_c(\{\mathbf{x}_i\}_{i \in I_j}), j = \pm\}.$$

Similarly, let ERC_c denote the set of extreme points of RC_c (where the points are marked by their respective class labels). Note that even if the convex hulls of the two classes intersect, there (usually[2]) exists a $c' \geq 0$ such that the c' reduced convex hulls of the two classes do not intersect.

The nu-SVM literature shows that for every C, there exists a $c \geq 0$ that soft margin SVM direction with tuning parameter C is equivalent to the hard margin SVM direction of the extreme points of the c-reduce convex hull of the data (ERC_c) which are a subset of the convex hull of the original data.

We point this geometric insight out because it gives similar geometric insights into SVM as our paper. Furthermore, the RCH formulation connects soft margin

[2]If, for example, the class means are identical the RCH formulation may breakdown.

SVM to the maximal data piling direction; in particular, soft margin SVM is the MDP of the extreme points of the RCH.

A.3. Relations between SVM and other classifiers

We have shown SVM can be exactly or approximately equivalent to the mean difference or maximal data piling direction (or possibly cropped versions of these two classifiers). When the data are balanced and C is sufficiently small, SVM becomes exactly the mean difference. When the data are unbalanced, SVM becomes a cropped version of the mean difference. Hard margin SVM is always the maximal data piling direction of the support vectors meaning it can be viewed as a cropped MDP. We gave conditions for when hard margin SVM is exactly the MDP of the full dataset.

These results are mathematically interesting i.e. they give conditions when a quadratic optimization problem reduces (exactly or approximately) to a problem which has a closed form solution with a simple geometric interpretation. By carefully studying how this behavior depends on the tuning parameter we give a number of insights into tuning SVM (see Section 6).

Furthermore, these insights can be directly relevant to the data analyst. For example, the analyst may learn something about the data when they encounter scenarios in which SVM is either exactly or approximately equivalent to one of these simple classifiers. In scientific applications using SVM, the data analyst may want to know more about why cross-validation selects a given tuning parameter.

Our results help both practitioners and researchers transfer intuition from the MD and MDP classifiers to SVM and vice versa. The mean difference classifier is widely used (especially if one takes the data transformation perspective from Section 2.2) and a lot is known about when it works well and doesn't (e.g. if the two classes are homoskedastic point clouds). While the MDP is an active topic of research, as discussed in Miao (2015), we understand some cases when the MDP works well and does not.

Finally, the results in this paper raise the question: how much performance gain does SVM achieve over more simple classifiers? For example, for a particular application it could be the case that the mean difference plus some combination of simple data transformation, robust mean estimation, and/or kernels would achieve a very similar test set error rate as SVM. This question is important to practitioners because more simple models are often favored for reasons of interpretability, computation, robustness, etc.

Appendix B: Improved SVM intercept for cross-validation

As discussed in Section 5.3, SVM's intercept can be problematic for small values of C; for small values of C the margin bounce causes every point to be classified to the larger of the two classes. This fact alone may not be concerning, however,

as Theorem 4.1 and Definition 4.3 show, SVM can behave differently, as a function of C, for cross-validation and on the full data set. The subsampled data sets for cross-validation will have a smaller diameter, D, meaning the threshold C_{small} is larger for these datasets than for the full dataset. In particular, the margin explosion happens a larger value of C during cross-validation than it does for the full dataset. This will cause the cross-validation test set error to be large for values of C where the test set error may in fact be small.

We can fix this issue by modifying the SVM intercept. This is significant in that it only involves a minor modification of existing SVM software and does not require new algorithms to be developed. Note that previous papers have suggesting modifying SVM's intercept Crisp, Burges (2000). Suppose we fit SVM to a dataset and it returns normal vector and intercept $\mathbf{w}_{svm}$ and b_{svm} respectively. Furthermore, define the *SVM centroids* by

$$\mathbf{m}_{svm,+} = \frac{1}{A} \sum_{i \in I_+} \alpha_i \mathbf{x_i},$$

where the α_i are the support vectors weights and A is the total weight (Equation (20)). Note this is a convex combination of points in the positive class (hence the name SVM centroid). We define $\mathbf{m}_{svm,-}$ similarly for the negative class.

Next define an new intercept by

$$b_{centroid} := \frac{1}{2} \mathbf{w}_{svm}^T (\mathbf{m}_{svm,+} + \mathbf{m}_{svm,-}) \tag{10}$$

Note $b_{centroid}$ is the value such that SVM's separating hyperplane sits halfway between $\mathbf{m}_{svm,+}$ and $\mathbf{m}_{svm,-}$. Furthermore, note this quantity can be computed when a kernel is used.

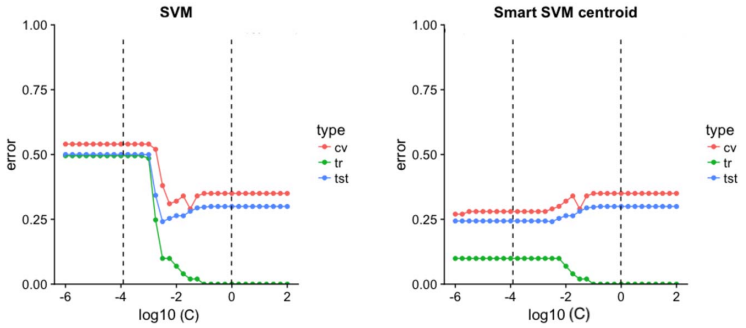
The SVM intercept is only a problem when C is small and one class is entirely support vectors (i.e. $\alpha_i > 0 \forall i \in I_+$ or $\forall i \in I_-$). Finally, we define a new intercept as follows

$$b = \begin{cases} b_{centroid}, & \text{if one class is entirely support vectors} \\ b_{svm}, & \text{otherwise} \end{cases} \tag{11}$$

Note that when the optimal value of C is large, the margin explosion discussed in this section is not an issue and b defined above will give the same result as the original b_{svm}.

The intercepts $b_{centroid}$ and b defined above are not the only options. One could, for example, replace the SVM centroids with the class means (i.e. replace $\mathbf{m}_{svm,-}$ with $\mathbf{x}_+$). Alternatively, one could use cross-validation to select b separately from $\mathbf{w}$. We focus on $b_{centroid}$ because it is simple can be interpreted as viewing SVM as a nearest centroid (as discussed in Section 2.1).

Below we demonstrate an example where b defined above improves SVM's test set performance. In this example, there are $n_+ = 51$ and $n_- = 50$ points in each class living in $d = 100$ dimensions. The two classes are generated from Gaussians with identity covariance and means which differ only in the first



(a) Regular SVM intercept (b) Intercept using SVM centroids

FIG 4. *Tuning error curves for standard SVM intercept vs. improved SVM intercept.*

coordinate; the mean of the positive class is the first standard basis vector and the mean of the negative class is negative the first standard basis vector. Note that MD is the Bayes rule in this example. We tune SVM using using 5-fold cross-validation to select the optimal value of C the compute the resulting test set error for an independent test set of 2000 points.

Figure 4 shows the error tuning curves (as in Figure 2d) for the two choices of SVM intercepts for a single draw of the data. The x-axis is the tuning parameter and the y-axis is the resulting SVM error for training, testing, and 5-fold cross-validation test set error. In the left panel we see each error curve jumps up to around 50% for small values of C for the regular SVM intercept. Furthermore, this error explosion happens for a smaller value of C for the test set error than for the cross-validation error (i.e. the blue test curve is to the left of the red cross validation curve). In the right panel, with the SVM centroid intercept, the error rate does not explode; moreover, the test error curve behaves similarly to the cross-validation curve. The curves on the right and left panels are identical for $C > 10^{-2}$. For this data set, 5-fold cross-validation gives a test set error of 28.1% for the regular SVM intercept, but 24.35% for the SVM centroid intercept.

Over 200 repetitions of this simulation, regular SVM has an mean test set error of 25.95% (MD gives 23.95%). If we replace the regular SVM intercept, b_{svm} with b defined above we get an average test set error of 24.80%; this intercept gives an average improvement of 1.15% for this dataset (this difference is statistically significant using a paired t-test which gives a p-value of 2×10^{-16}).

When the classes are unbalanced other error metrics are used (e.g. F-score, AUC, Choen's Kappa, etc). If AUC is used i.e. the intercept is tuned independently of the direction, issues with the intercept discussed in this section will not occur. However, when other metrics are used the improved intercepts will likely be more effective.

The intercept b defined above will not improve SVM's performance in all scenarios, but is not likely to harm the performance. The intercept b, however, is simple to implement and can give a better test set error.

Appendix C: Proofs for hard-margin SVM

C.1. Proofs for Section 3.1

Proof. **of Theorem 3.1** We first prove the existence and uniqueness of complete data piling directions P in the affine hull of the data. We then show that this unique, affine data piling direction is in fact the direction of maximal data piling.

Recall we assume that $d \geq n-1$ and the data are in general position. Let the set of affine directions A be given as follows

$$A = \{\mathbf{a}_1 - \mathbf{a}_2 | \mathbf{a}_j \in \text{aff}(\{\mathbf{x}_i\}_1^n), j = 1, 2\}.$$

Note that A is the $n-1$ dimensional subspace parallel to the affine space $\text{aff}(\{\mathbf{x}_i\}_1^n)$ generated by the data i.e. A contains the origin.

We first show that without loss of generality $d = n-1$. Note that both A and P are invariant to a fixed translation of the data. Therefore, we may translate the data so that $0 \in \text{aff}(\{\mathbf{x}_i\}_1^n)$ (e.g. translate by the mean of the data). The data now span an $n-1$ dimensional subspace since the affine hull of the data now contains the origin. Furthermore, $\text{span}(\{\mathbf{x}\}_1^n) = \text{aff}(\{\mathbf{x}\}_1^n) = A$. Thus without loss of generality we may consider the data to in fact be $n-1$ dimensional (i.e. $d = n-1$).

We are now looking for a vector $\mathbf{v} \in A$ that gives complete data piling. Note by the above discussion and assumption we have $A = \mathbb{R}^d$. This means we are looking for $\mathbf{v} \in \mathbb{R}^d$ and $a, b \in \mathbb{R}$ with $a \neq 0$ satisfying the following n linear equations

$$\mathbf{x}_i^T \mathbf{v} = a y_i + b \text{ for } i = 1, \ldots, n.$$

Since the magnitude of $\mathbf{v}$ is arbitrary we fix $a = 1$ without loss of generality. We now have

$$\mathbf{x}_i^T \mathbf{v} = y_i + b \text{ for } i = 1, \ldots, n$$

which can be written in matrix form as

$$X\mathbf{v} + b\mathbf{1}_n = \mathbf{y} \tag{12}$$

where $X \in \mathbb{R}^{n \times d}$ is the data matrix whose rows are the data vectors $\mathbf{x}_i$ and $\mathbf{y} \in \mathbb{R}^n$ is the vector of class labels. This is a system of n equations in $\mathbf{R}^{d+1}$ which can be seen by appending 1 onto the end of each $\mathbf{x}_i$ i.e. $\tilde{\mathbf{x}}_i = (\mathbf{x}_i, 1) \in \mathbb{R}^{d+1}$ and letting $\mathbf{w} = (\mathbf{v}, b)$. Then Equation (12) becomes

$$\widetilde{X}\mathbf{w} = \mathbf{y} \tag{13}$$

where $\widetilde{X} \in \mathbb{R}^{n \times d+1}$ is the appended data matrix.

Recall that we assumed $d = n-1$ so Equation (13) is a system of n equations in $\mathbb{R}^n$. Further recall that the data are in general position meaning that the n data points are affine independent in the $n-1$ dimensional subspace of the data. Affine independence is equivalent to linear independence of $\{(\mathbf{x}_i, 1)\}_1^n$. Therefore

the matrix $\widetilde{X} \in \mathbb{R}^{n \times n}$ has full rank and Equation (13) always has a solution, $\mathbf{v}^*$, and this solution is unique.

Existence of a solution to Equation (13) shows that $P \cap A \neq \emptyset$. Uniqueness of the solution to Equation (13) shows that this intersection $P \cap A$ can have only one direction of which $\mathbf{v}^*$ is a representative element.

We now show that $\mathbf{v}^*$ is in fact the maximal data piling direction. We no longer assume that $d = n - 1$.

We first construct an orthonormal basis $\{\mathbf{t}_i\}_1^d$ of $\mathbb{R}^d$ as follows. Let the first $n-1$ basis vectors $\mathbf{t}_1, \ldots, \mathbf{t}_{n-1}$ span A. Let $\mathbf{t}_n$ be orthogonal to A but in the span of the data $\{\mathbf{x}_i\}_1^n$ (recall the data span an n dimensional space while the affine hull of the data is $n-1$ dimensional). Let the remaining $d - n + 1$ basis vectors be orthogonal to A and the span of the data.

We show that the vector $\mathbf{t}_n$ projects every data point onto a single point i.e. $\mathbf{x}_i^T \mathbf{t}_n = c$ for each $i = 1, \ldots n$ and some $c \in \mathbb{R}$. Suppose we translate $\text{aff}(\{\mathbf{x}_i\}_1^n)$ along $\mathbf{t}_n$ until the origin lies in the affine hull of the translated data. In particular, the data now span an $n-1$ dimensional subspace that is orthogonal to $\mathbf{t}_n$ (where as before they spanned an n dimensional subspace). We now have that for some $c \in \mathbb{R}$

$$\mathbf{t}_n^T(\mathbf{x}_i + c\mathbf{t}_n) = 0 \text{ for each } i = 1, \ldots, n$$
$$\mathbf{t}_n^T \mathbf{x}_i = c \text{ for each } i = 1, \ldots, n$$

since $\mathbf{t}_n$ is unit norm.

Let $\mathbf{v} \in \mathbb{R}^d$ be a representative vector of the direction in the affine hull of the data that gives complete data piling (given above). Suppose $\mathbf{v}$ has unit norm and is oriented such that

$$\mathbf{v}^T \mathbf{x}_i = a y_i + b$$

for some $a, b \in \mathbb{R}$ with $a > 0$ (note fixing $a > 0$ eliminates the antipodal symmetry of data piling vectors).

We now show that $\mathbf{v}$ is in fact the maximal data piling direction. Let $\mathbf{w} \in \mathbb{R}^d$ be another vector with unit norm that gives complete data piling (i.e. $\mathbf{w} \in P$). In particular, there exists $a_v, a_w, b_v, b_w \in \mathbb{R}$ with $a_v, a_w > 0$ such that

$$\mathbf{v}^T \mathbf{x}_i = a_v y_i + b_v \text{ for each } i = 1, \ldots, n.$$
$$\mathbf{w}^T \mathbf{x}_i = a_w y_i + b_w \text{ for each } i = 1, \ldots, n.$$

Assume for the sake of contradiction that $\mathbf{w}$ projects the data possibly further apart than $\mathbf{v}$ does. In particular assume that $a_w \geq a_v$.

Since $\{\mathbf{t}_i\}_1^d$ is a basis we can write

$$\mathbf{w} = \sum_{i=1}^{d} \alpha_i \mathbf{t}_i.$$

Next compute the dot products with the data. For any $j = 1, \ldots, n$,

$$\mathbf{w}^T \mathbf{x}_j = \left(\sum_{i=1}^{n-1} \alpha_i \mathbf{t}_i \right)^T \mathbf{x}_j + \alpha_n \mathbf{t}_n^T \mathbf{x}_j + \sum_{i=n+1}^{d} \alpha_i \mathbf{t}_i^T \mathbf{x}_j.$$

Recall the basis vectors $\mathbf{t}_{n+1}, \ldots, \mathbf{t}_d$ are orthogonal to the data points so the third term in the sum is zero. Furthermore, the dot product of $\mathbf{t}_n$ with each data point is a constant. Thus we now have

$$\mathbf{w}^T \mathbf{x}_j = \left(\sum_{i=1}^{n-1} \alpha_i \mathbf{t}_i \right)^T \mathbf{x}_j + \alpha_n c, \text{ for all } j = 1, \ldots, n.$$

Thus we can see the vector

$$\mathbf{w}' = \sum_{i=1}^{n-1} \alpha_i \mathbf{t}_i$$

also gives complete data piling. However this vector lies in A since it is a linear combination of the first $n-1$ basis vectors. We have shown that there is only one direction in A with complete data piling thus $\sum_{i=1}^{n-1} \alpha_i \mathbf{t}_i \propto \mathbf{v}$. In particular, for some $\alpha > 0$

$$\sum_{i=1}^{n-1} \alpha_i \mathbf{t}_i = \alpha \mathbf{v}.$$

So we now have

$$\mathbf{w}' = \alpha \mathbf{v} + \alpha_n \mathbf{t}_n.$$

Recall $||\mathbf{v}|| = ||\mathbf{w}|| = 1$ and $\mathbf{t}_n$ is orthogonal to $\mathbf{v}$ by construction. Therefore $\alpha^2 + \alpha_n^2 = 1$. In particular if $\alpha_n > 0$ then $\alpha < 1$.

Let $\mathbf{x}_+$ and $\mathbf{x}_-$ be any point from the positive and negative class respectively. By construction we have

$$\mathbf{v}^T(\mathbf{x}_+ - \mathbf{x}_-) = a_v.$$
$$\mathbf{w}^T(\mathbf{x}_+ - \mathbf{x}_-) = a_w.$$

However expanding this last line we get

$$\mathbf{w}^T(\mathbf{x}_+ - \mathbf{x}_-) = (\alpha \mathbf{v} + \alpha_n \mathbf{t}_n)^T(\mathbf{x}_+ - \mathbf{x}_-)$$
$$\mathbf{w}^T(\mathbf{x}_+ - \mathbf{x}_-) = \alpha \mathbf{v}^T(\mathbf{x}_+ - \mathbf{x}_-) + \alpha_n \mathbf{t}_n^T(\mathbf{x}_+ - \mathbf{x}_-).$$

But $\mathbf{t}_n^T \mathbf{x}_+ = \mathbf{t}_n^T \mathbf{x}_- = c$ so the last term is zero. Thus we now have

$$\mathbf{w}^T(\mathbf{x}_+ - \mathbf{x}_-) = \alpha a_v.$$

Thus

$$\alpha a_v = a_w.$$

However unless $\mathbf{w} = \mathbf{v}$ (so $\alpha_n = 0$) we have $0 < \alpha < 1$. Therefore $a_w < a_v$ contradicting the assumption that $a_w \geq a_v$. Therefore $\mathbf{v}$ is the maximal data piling direction. □

C.2. Proofs for Section 3.2

Derivation and discussion of the KKT conditions can be found in Mohri et al. (2012). From the Lagrangian of Problem (4) we can derive the KKT conditions

$$\mathbf{w}_{hm-svm} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \tag{14}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \tag{15}$$

$$\alpha_i = 0 \text{ or } y_i(\mathbf{w}^T \mathbf{x}_i + b) = 1, \tag{16}$$

with $\alpha_i \geq 0$ for each $i = 1, \ldots, n$.

Condition (15) says that the sum of the weights in both classes has to be equal. Combining this with (14) we find that the hard margin SVM direction is given by

$$\mathbf{w}_{hm-svm} \propto \sum_{i \in I_+} \frac{\alpha_i}{A} \mathbf{x}_i - \sum_{i \in I_-} \frac{\alpha_i}{A} \mathbf{x}_i, \tag{17}$$

where $\sum_{i \in I_+} \alpha_i = \sum_{i \in I_-} \alpha_i := A$. Thus $\mathbf{w}_{hm-svm} \in C$ i.e. the hard margin SVM direction is always a convex direction. As discussed in Bennett, Bredensteiner (2000); Pham (2010) hard margin SVM is equivalent to finding the nearest points in the convex hulls of the two classes.

The last KKT condition (16) says that a point $\mathbf{x}_i$ either lies on one of the marginal hyperplanes $\{\mathbf{x} | \mathbf{w}_{hm-svm}^T \mathbf{x} = \pm 1\}$ or receives zero weight. In the former case when $\alpha_i \neq 0$, $\mathbf{x}_i$ is called a *support vector*. The margin width, ρ, is given by the magnitude of the normal vector

$$\rho^2 = \frac{1}{||\mathbf{w}_{hm-svm}||_2^2} = \frac{1}{\sum_{i=1}^{n} \alpha_i} := \frac{1}{||\alpha||_1}. \tag{18}$$

The following lemma about SVM and MDP is a consequence of the fact that complete data piling directions satisfy the SVM KKT conditions.

Lemma C.1. *If hard margin SVM has complete data piling then the SVM direction is equivalent to the MDP direction i.e.*

$$\mathbf{w}_{hm-svm} \in P \implies \mathbf{w}_{hm-svm} \propto \mathbf{w}_{mdp}.$$

Lemma C.2. *If $P \cap Cvx \neq \emptyset$ then $\mathbf{w}_{svm} \in P \cap Cvx$.*

Proof. Let $\mathbf{v} \in P \cap \text{Cvx}$. We show $\mathbf{v}$ satisfies the KKT conditions. The lemma then follows since the KKT conditions necessary and sufficient for hard margin SVM (the constraints are qualified, see Chapter 4 of Mohri et al. 2012).

Since $\mathbf{v} \in \text{Cvx}$ we have that $\mathbf{v} \propto \mathbf{c}_+ - \mathbf{c}_-$ where $\mathbf{c}_j \in \text{conv}(\{\mathbf{x}_i\}_{i \in I_j})$. For some constant $a > 0$

$$\mathbf{v} = a \left(\sum_{i \in I_+} \lambda_i \mathbf{x}_i - \sum_{i \in I_-} \lambda_i \mathbf{x}_i \right),$$

where

$$\sum_{i\in I_+}\lambda_i = \sum_{i\in I_+}\lambda_i = 1 \text{ and } \lambda_i \geq 0.$$

Since $\mathbf{v} \in P$ we can select $b, \mathbf{v}$ such that

$$y_i(\mathbf{x}_i^T\mathbf{v} + b) = 1 \ \forall i.$$

But these three equations are the KKT conditions with $\alpha_i = a\lambda_i$. □

Appendix D: Proofs for soft-margin SVM

The KKT conditions for soft margin SVM are (see Mohri et al. 2012 for derivations)

$$\mathbf{w}_{svm} = \sum_{i\in I_+}\alpha_i\mathbf{x}_i - \sum_{i\in I_-}\alpha_i\mathbf{x}_i, \tag{19}$$

$$\sum_{i\in I_+}\alpha_i = \sum_{i\in I_-}\alpha_i := A, \tag{20}$$

$$\alpha_i + \mu_i = C \text{ for } i = 1,\ldots,n, \tag{21}$$

$$\alpha_i = 0 \text{ or } y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1 - \xi_i \text{ for } i = 1,\ldots,n, \tag{22}$$

$$\xi_i = 0 \text{ or } \mu_i = 0 \text{ for each } i, \tag{23}$$

For soft margin SVM we define the marginal hyper planes to be $\{\mathbf{x}|\mathbf{x}^T\mathbf{w}_{svm} = \pm 1\}$ and the margin width (or just margin), ρ the distance from the separating hyperplane to the marginal hyperplanes. By construction $\rho = \frac{1}{||\mathbf{w}_{svm}||}$. A points is a support vector if and only if it is contained within the marginal hyperplanes.

As with hard margin SVM, the soft margin direction is always a convex direction. Again points $\mathbf{x}_i$ such that $\alpha_i \neq 0$ are called support vectors. We further separate support vectors into two types.

Definition D.1. *Margin vectors are support vectors* $\mathbf{x}_i$ *such* $\alpha_i \neq 0$ *and* $\xi_i = 0$.

Definition D.2. *Slack vectors are support vectors* $\mathbf{x}_i$ *such* $\alpha_i \neq 0$ *and* $\xi_i > 0$.

Margin vectors are support vectors lying on one of the two marginal hyperplanes. Slack vectors are support vectors lying strictly on the inside of the marginal hyperplanes. Call the set of margin vectors in each class M_j and the set of slack vectors L_j for $j = \pm$.

The KKT conditions imply

- all support vectors receive weight upper bounded by C ($\mathbf{x}_i \in M_j \implies 0 < \alpha_i \leq C$)
- slack vectors receive weight exactly C ($\mathbf{x}_i \in L_j \implies \alpha_i = C$)

Furthermore, the following constraint balances the weights between the two classes

$$C|L_+| + \sum_{i \in M_+} \alpha_i = C|L_-| + \sum_{i \in M_-} \alpha_i. \tag{24}$$

We assume that the positive class is the larger of the two classes i.e. $n_+ \geq n_-$. Unbalanced classes means $n_+ > n_-$.

D.1. Small C regime

As $C \to 0$ the margin width increases to infinity ($\rho \to \infty$). As the margin width grows as many points as possible become slack vectors and all slack vectors get the same weight $\alpha_i = C$. Hence if the classes are balanced the SVM direction will be equivalent to the mean difference. If the classes are unbalanced then there will be some margin vectors which receive weight $\alpha_i \leq C$. The number of margin vectors is bounded by the class sizes and the dimension.

Note the diameter, D, does not change if we consider the convex hull of the two classes (proof of Lemma D.1 is a straightforward exercise).

Lemma D.1.

$$\max_{\mathbf{c}_j \in conv(\{\mathbf{x}_i\}_{i \in I_j})} ||\mathbf{c}_+ - \mathbf{c}_-|| = \max_{\mathbf{x}_j \in I_+} ||\mathbf{x}_+ - \mathbf{x}_-|| =: D.$$

As $C \to 0$ the magnitude of $\mathbf{w}_{svm}$ goes to zero. In particular, the KKT conditions give the following bound.

Lemma D.2. *For a given C the magnitude of the SVM solution is*

$$||\mathbf{w}_{svm}|| \leq n_+ C \cdot D.$$

Proof. From the KKT conditions we have

$$\mathbf{w}_{svm} = \sum_{i \in I_+} \alpha_i \mathbf{x}_i - \sum_{i \in I_-} \alpha_i \mathbf{x}_i$$

and

$$\sum_{i \in I_+} \alpha_i = \sum_{i \in I_-} \alpha_i =: A.$$

Computing the magnitude of $\mathbf{w}_{svm}$

$$||\mathbf{w}_{svm}|| = A \left|\left| \sum_{i \in I_+} \frac{\alpha_i}{A} \mathbf{x}_i - \sum_{i \in I_-} \frac{\alpha_i}{A} \mathbf{x}_i \right|\right|.$$

Since the two terms are convex combinations we get

$$||\mathbf{w}_{svm}|| \leq A \sup_{\mathbf{c}_j \in \text{conv}(\{\mathbf{x}_i\}_{i \in I_j})} ||\mathbf{c}_+ - \mathbf{c}_-||.$$

applying Lemma D.1

$$||\mathbf{w}_{svm}|| = A \max_{\mathbf{x}_j \in I_+} ||\mathbf{x}_+ - \mathbf{x}_-||$$
$$||\mathbf{w}_{svm}|| = AD.$$

Since $0 \leq \alpha_i \leq C$ we get $A \leq n_+ C$ thus proving the bound. □

Since the magnitude of $\mathbf{w}_{svm}$ determines the margin width, using the previous lemma we get the following corollary.

Corollary D.1. *The margin ρ goes to infinity as C goes to zero. In particular*

$$\rho = \frac{1}{||\mathbf{w}_{svm}||} \geq \frac{1}{n_+ CD}.$$

Since the margin width increases, for small enough C the smaller class becomes all slack variables.

Lemma D.3. *If $C < C_{small}$ then all points in the smaller class become slack vectors ($\xi_i > 0$ for all $i \in I_-$).*

Proof. By Corollary D.1 the margin width goes to infinity as $C \to 0$ since

$$\rho \geq \frac{1}{n_+ CD}.$$

Recall the margin width, ρ, is the distance from the separating hyperplane to the marginal hyperplanes. Note that if $\rho > \frac{1}{2}D$ then at least one class must be complete slack. Thus if $C < \frac{2}{n_+ D^2}$ at least one class must be complete slack i.e. $\xi_i > 0$ for all $i \in I_j$ for $j = +$ and/or $j = -$. If the classes are balanced then either class can become complete slack (or both classes).

If the classes are unbalanced i.e. $n_- < n_+$ then the smaller class becomes complete slack. To see this, assume for the sake of contradiction that the larger class becomes complete slack i.e. $\xi_i \neq 0$ for each $i \in I_+$. Then the KKT conditions imply $\alpha_i = C$ for each $i \in I_+$. KKT condition (20) says

$$\sum_{i \in I_+} \alpha_i = \sum_{i \in I_-} \alpha_i$$
$$n_+ C = \sum_{i \in I_-} \alpha_i.$$

But $\alpha_i \leq C$ and $n_- < n_+$ by assumption therefore this constraint cannot be satisfied. □

If the classes are balanced then the margin swallows both classes and the SVM direction becomes the mean difference direction.

Lemma D.4. *If the classes are balanced and $C < C_{small}$ the SVM direction is equivalent to the mean difference direction i.e. $\mathbf{w}_{svm} \propto \mathbf{w}_{md}$.*

Proof. When $C < C_{\text{small}}$ one of the classes (without loss of generality the negative class) becomes slack i.e. $\xi_i > 0$ for each $i \in I_-$ thus $\alpha_i = C$ for each $i \in I_-$. The KKT conditions then require

$$\sum_{i \in I_+} \alpha_i = \sum_{i \in I_-} \alpha_i = n_- C.$$

Since $\alpha_i \leq C$ and $|I_+| = n_-$ this constraint can only be satisfied if $\alpha_i = C$ for each $i \in I_+$. We now have

$$\mathbf{w}_{svm} = \sum_{i \in I_+} C\mathbf{x}_i - \sum_{i \in I_-} C\mathbf{x}_i$$
$$\mathbf{w}_{svm} = C\frac{n}{2}(\bar{\mathbf{x}}_+ - \bar{\mathbf{x}}_-) \propto \mathbf{w}_{md}.$$

□

Lemma D.5. *If the classes are unbalanced and $C < C_{small}$ the SVM solution satisfies the the constraints in Equations (8), (9).*

Proof. Recall for $C < C_{\text{small}}$ we have $\xi_i > 0$ for $i \in I_-$. From the KKT conditions $\xi_i > 0 \implies \mu_i = 0 \implies \alpha_i = 0$ meaning $\alpha_i = C$ for each $i \in I_-$. The weight balance constraint (24) from the KKT conditions becomes

$$C|L_+| + \sum_{i \in M_+} \alpha_i = C|L_-| + \sum_{i \in M_-} \alpha_i,$$

which then implies the conditions on $\mathbf{w}_{svm}$. □

Corollary D.2. *When $C < C_{small}$ the larger (positive) class can have at most n_- slack vectors. If the larger class has more than n_- support vectors then at least one of them must be a margin vector.*

D.2. Large C regime

Lemma D.6. *If there is at least one slack vector then for a given C*

$$||\mathbf{w}_{svm}|| \geq CG,$$

or equivalently

$$\rho \leq \frac{1}{CG},$$

where G is the class gap.

Proof. From the KKT conditions

$$||\mathbf{w}_{svm}|| = || \sum_{i \in I_+} \alpha_i \mathbf{x}_i - \sum_{i \in I_-} \alpha_i \mathbf{x}_i ||,$$
$$||\mathbf{w}_{svm}|| = A|| \sum_{i \in I_+} \frac{\alpha_i}{A} \mathbf{x}_i - \sum_{i \in I_-} \frac{\alpha_i}{A} \mathbf{x}_i ||,$$

where $A = \sum_{i \in I_+} \alpha_i = \sum_{i \in I_-} \alpha_i$. Since the two sums are convex combinations, using the definition of G we get

$$||\mathbf{w}_{svm}|| \geq AG.$$

Since there is at least one slack vector there is at least one i such that $\alpha_i = C$ thus $A \geq C$ and the result follows. □

Appendix E: Convex and complete data piling directions

Theorem 3.2 gives a geometric characterization when the set of convex directions intersects the set of complete data piling directions. We can also characterize this event through a linear program.

An alternative way of deciding if $C \cap P = \emptyset$ and computing the intersection if it exists is through the following linear program (proof of Theorem E.1 is a straightforward exercise in linear programming).

Theorem E.1. *$C \cap P \neq \emptyset$ if and only if there is a solution to the following linear program*

$$\begin{aligned} \underset{\alpha \in \mathbb{R}^{n+}, \beta \in \mathbb{R}^{n-}, \mathbf{v} \in \mathbb{R}^d, b \in \mathbb{R}}{\textit{minimize}} \quad & 1 \\ \textit{subject to} \quad & X\mathbf{v} + \mathbf{1}_n b = \mathbf{y} \\ & \sum_{i \in I_+} \alpha_i \mathbf{x}_i - \sum_{i \in I_-} \beta_i \mathbf{x}_i = \mathbf{v} \\ & \sum_{i \in I_+} \alpha_i = 1 \\ & \sum_{i \in I_-} \beta_i = 1 \\ & \alpha_i, \beta_i \geq 0 \textit{ for } i = 1, \ldots, n. \end{aligned} \tag{25}$$

In the case a solution $\mathbf{v}$ *exists then* $\mathbf{v} \in C \cap P$.

The vector $\mathbf{1}_n \in \mathbb{R}^n$ is the vector of ones, X is the $\mathbb{R}^{n \times d}$ data matrix and $\mathbf{y} \in \mathbb{R}^n$ is the vector of class labels. The first constraint says $\mathbf{v}$ must be a complete data piling direction, $\mathbf{v} \in P$. The remaining constraints say $\mathbf{v}$ must be a convex direction, $\mathbf{v} \in C$.

Note that solving this linear program is at least as hard as solving the original SVM quadratic program therefore Theorem E.1 is not of immediate computational interest. This theorem, however, does give an alternate mathematical description $C \cap P \neq \emptyset$ which may be of theoretical interest.

References

Ahn Jeongyoun, Lee Myung Hee, Yoon Young Joo. Clustering high dimension, low sample size data using the maximal data piling distance // Statistica Sinica. 2012. 443–464. MR2954347

Ahn Jeongyoun, Marron J. S. The maximal data piling direction for discrimination // Biometrika. 2010. 97, 1. 254–259. MR2594434

Anderson Theodore Wilbur. An introduction to multivariate statistical analysis. 1962. MR0091588

Ayat Nedjem-Eddine, Cheriet Mohamed, Suen Ching Y. Automatic model selection for the optimization of SVM kernels // Pattern Recognition. 2005. 38, 10. 1733–1745.

Barbero Alvaro, Takeda Akiko, López Jorge. Geometric intuition and algorithms for Ev-SVM // The Journal of Machine Learning Research. 2015. 16, 1. 323–369. MR3335794

Bennett Kristin P, Bredensteiner Erin J. Duality and geometry in SVM classifiers // ICML. 2000. 57–64.

Bishop Christopher. Pattern Recognition and Machine Learning. 2006. MR2247587

Chapelle Olivier, Vapnik Vladimir. Model selection for support vector machines // Advances in neural information processing systems. 2000. 230–236.

Chen Pai-Hsuen, Lin Chih-Jen, Schölkopf Bernhard. A tutorial on ν-support vector machines // Applied Stochastic Models in Business and Industry. 2005. 21, 2. 111–136. MR2137545

Cherkassky Vladimir, Ma Yunqian. Practical selection of SVM parameters and noise estimation for SVM regression // Neural networks. 2004. 17, 1. 113–126.

Crisp David J, Burges Christopher JC. A geometric interpretation of v-SVM classifiers // Advances in neural information processing systems. 2000. 244–250.

Towards a rigorous science of interpretable machine learning. //. 2017.

Duan Kaibo, Keerthi S Sathiya, Poo Aun Neow. Evaluation of simple performance measures for tuning SVM hyperparameters // Neurocomputing. 2003. 51. 41–59.

Duarte Edson, Wainer Jacques. Empirical comparison of cross-validation and internal metrics for tuning SVM hyperparameters // Pattern Recognition Letters. 2017. 88. 6–11.

Franc Vojtech, Zien Alexander, Schölkopf Bernhard. Support vector machines as probabilistic models // Proceedings of the 28th International Conference on Machine Learning (ICML-11). 2011. 665–672.

Friedman Jerome, Hastie Trevor, Tibshirani Robert. The elements of statistical learning. 1. 2001. MR1851606

Guidotti Riccardo, Monreale Anna, Turini Franco, Pedreschi Dino, Giannotti Fosca. A Survey Of Methods For Explaining Black Box Models // arXiv preprint arXiv:1802.01933. 2018.

Hastie Trevor, Rosset Saharon, Tibshirani Robert, Zhu Ji. The entire regularization path for the support vector machine // Journal of Machine Learning Research. 2004. 5, Oct. 1391–1415. MR2248021

Jaggi Martin. An equivalence between the lasso and support vector machines. 2014. MR3380631

Lee Myung Hee, Ahn Jeongyoun, Jeon Yongho. HDLSS discrimination with adaptive data piling // Journal of Computational and Graphical Statistics.

2013. 22, 2. 433–451. MR3173723

Lin Yi, Wahba Grace, Zhang Hao, Lee Yoonkyung. Statistical properties and adaptive tuning of support vector machines // Machine Learning. 2002. 48, 1. 115–136. MR2703664

Marron James Stephen, Todd Michael J, Ahn Jeongyoun. Distance-weighted discrimination // Journal of the American Statistical Association. 2007. 102, 480. 1267–1271. MR2412548

Mattera Davide, Haykin Simon. Support vector machines for dynamic reconstruction of a chaotic system // Advances in kernel methods. 1999. 211–241.

Mavroforakis Michael E, Theodoridis Sergios. A geometric approach to support vector machine (SVM) classification // IEEE transactions on neural networks. 2006. 17, 3. 671–682.

Miao Di. CLASS-SENSITIVE PRINCIPAL COMPONENTS ANALYSIS. 2015. MR3358280

Mohri Mehryar, Rostamizadeh Afshin, Talwalkar Ameet. Foundations of machine learning. 2012. MR3057769

Murphy Kevin P. Machine learning: a probabilistic perspective. 2012.

Pham Tung. Some Problems in High Dimensional Data Analysis. 2010.

Polson Nicholas G, Scott Steven L, others . Data augmentation for support vector machines // Bayesian Analysis. 2011. 6, 1. 1–23. MR2781803

Rahimi Ali, Recht Benjamin. Random features for large-scale kernel machines // Advances in neural information processing systems. 2008. 1177–1184.

Schölkopf Bernhard, Smola Alex J, Williamson Robert C, Bartlett Peter L. New support vector algorithms // Neural computation. 2000. 12, 5. 1207–1245.

Schölkopf Bernhard, Smola Alexander J. Learning with kernels: support vector machines, regularization, optimization, and beyond. 2002.

Shawe-Taylor John, Cristianini Nello. Kernel methods for pattern analysis. 2004.

Sollich Peter. Bayesian methods for support vector machines: Evidence and predictive class probabilities // Machine learning. 2002. 46, 1-3. 21–52.

Steinwart Ingo, Christmann Andreas. Support vector machines. 2008. MR2450103

Stigler Stephen M. The asymptotic distribution of the trimmed mean // The Annals of Statistics. 1973. 472–477. MR0359134

Sun Jiancheng, Zheng Chongxun, Li Xiaohe, Zhou Yatong. Analysis of the distance between two classes for tuning SVM hyperparameters // IEEE transactions on neural networks. 2010. 21, 2. 305–318.

Tibshirani Robert, Hastie Trevor, Narasimhan Balasubramanian, Chu Gilbert. Diagnosis of multiple cancer types by shrunken centroids of gene expression // Proceedings of the National Academy of Sciences. 2002. 99, 10. 6567–6572.

Vapnik Vladimir. The nature of statistical learning theory. 2013. MR1367965

Vapnik Vladimir Naumovich. An overview of statistical learning theory // IEEE transactions on neural networks. 1999. 10, 5. 988–999. MR1641250