Journal of Machine Learning Research () Submitted 12/08; Published

Maximum Relative Margin and Data-Dependent Regularization

Pannagadatta K. Shivaswamy PKS2103@CS.COLUMBIA.EDU
Tony Jebara JEBARA@CS.COLUMBIA.EDU
Department of Computer Science
Columbia University
New York, NY 10027, USA

Editor:

Abstract

Leading classification methods such as support vector machines (SVMs) and their counterparts achieve strong generalization performance by maximizing the margin of separation between data classes. While the maximum margin approach has achieved promising performance, this article identifies its sensitivity to affine transformations of the data and to directions with large data spread. Maximum margin solutions may be misled by the spread of data and preferentially separate classes along large spread directions. This article corrects these weaknesses by measuring margin not in the absolute sense but rather only relative to the spread of data in any projection direction. Maximum relative margin corresponds to a data-dependent regularization on the classification function while maximum absolute margin corresponds to an ℓ_2 norm constraint on the classification function. Interestingly, the proposed improvements only require simple extensions to existing maximum margin formulations and preserve the computational efficiency of SVMs. Through the maximization of relative margin, surprising performance gains are achieved on real-world problems such as digit, text classification and on several other benchmark datasets. In addition, risk bounds are derived for the new formulation based on Rademacher averages.

Keywords: Support Vector Machines, Kernel Methods, Large Margin, Rademacher Complexity, MNIST.

1. Introduction

In classification problems, the aim is to learn a classifier that generalizes well on future data from a limited number of training examples. Support vector machines (SVMs) and maximum margin classifiers (Vapnik, 1995; Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004) have been a particularly successful approach both in theory and in practice. Given a labeled training set, these return a predictor that accurately labels previously unseen test examples. For simple binary classification in Euclidean spaces, this predictor is a function $f : \mathbb{R}^m \to \{\pm 1\}$ estimated from observed training data $(\mathbf{x}_i, y_i)_{i=1}^n$ consisting of inputs $\mathbf{x}_i \in \mathbb{R}^m$ and outputs $y_i \in \{\pm 1\}$. A linear function[1] $f(\mathbf{x}) := \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$ where $\mathbf{w} \in \mathbb{R}^m, b \in \mathbb{R}$ serves as the decision rule throughout this article. The parameters of the hyperplane $(\mathbf{w}, b)$ are estimated by maximizing the margin (e.g., the distance between the

1. In this article the dot product $\mathbf{w}^\top \mathbf{x}$ is used with the understanding that it can be replaced with a generalized inner product or by using a kernel for generic objects.

hyperplanes defined by $\mathbf{w}^\top \mathbf{x} + b = 1$ and $\mathbf{w}^\top \mathbf{x} + b = -1$) while minimizing a weighted upper bound on the misclassification rate on training data (via so-called slack variables). In practice, the margin is maximized by minimizing $\frac{1}{2}\mathbf{w}^\top \mathbf{w}$ plus an upper bound on the misclassification rate.

While maximum margin classification works well in practice, its solution can easily be perturbed by an (invertible) affine or scaling transformation of the input space. For instance, by transforming all training and testing inputs by an invertible linear transformation, the SVM solution and its resulting classification performance can be significantly varied. This is worrisome since an adversary could directly exploit this shortcoming and transform the data to drive performance down. Moreover, this phenomenon is not limited to an explicit adversarial setting, it can naturally occur in many real world classification problems, especially in high dimensions. This article will explore such shortcomings in maximum margin solutions (or equivalently, SVMs in the context of this article) which exclusively measure margin by the points near the classification boundary regardless of how spread the remaining data is away from the separating hyperplane. An alternative approach will be followed based on controlling the spread while maximizing the margin. This helps overcome this bias and produces a formulation that is affine invariant. The key is to recover a large margin solution while normalizing the margin by the spread of the data. Thus, margin is measured in a *relative* sense rather than in the absolute sense. In addition, theoretical results using Rademacher averages support this intuition. The resulting classifier will be referred to as the relative margin machine (RMM) and was first introduced by Shivaswamy and Jebara (2009a) with this longer article serving to provide more details, more thorough empirical evaluation and more theoretical support. In fact, the RMM approach has also been successfully extended to structured prediction problems as well (Shivaswamy and Jebara, 2009b).

Traditionally, controlling spread has been an important theme in classification problems. For instance, classical linear discriminant analysis (LDA) (Duda et al., 2000) finds projections of the data so that the inter-class separation is large while within-class scatter is small. However, the spread (or scatter in this context) is estimated by LDA using only simple first and the second order statistics of the data. While this is appropriate if class-conditional densities are Gaussian, second-order statistics are inappropriate for many real-world datasets and thus, the classification performance of LDA is typically weaker than that of SVMs. The estimation of spread should not make second-order assumptions about the data and should be tied to the margin criterion (Vapnik, 1995). A similar line of reasoning has been proposed to perform feature selection. Weston et al. (2000) showed that second order tests and filtering methods on features perform poorly compared to wrapper methods on SVMs which more reliably remove features that have low discriminative value. In this prior work, a feature's contribution to margin is compared to its effect on the radius of the data by computing bounding hyper-spheres rather than simple Gaussian statistics. Unfortunately, there, only axis-aligned feature selection was considered. Similarly, ellipsoidal kernel machines (Shivaswamy and Jebara, 2007) were proposed to normalize data in feature space by estimating bounding hyper-ellipsoids while avoiding inappropriate second-order assumptions. Similarly, the radius-margin bound has been used as a criterion to tune the hyper-parameters of the SVM (Keerthi, 2002). Another criterion based jointly on ideas from the SVM method as well as Linear Discriminant Analysis has been studied in

Zhang et al. (2005). This technique involves first solving the SVM and then solving an LDA problem based on the support vectors that were obtained. While these previous methods showed performance improvements, they relied on multiple-step locally optimal algorithms for interleaving spread information with margin estimation.

To overcome the limitations of local non-convex optimization schemes, the formulations derived here will remain convex, will be efficiently solvable and will admit helpful generalization bounds. A similar method to the RMM was described by Haffner (2001) yet that approach started from a different overall motivation. In contrast, this article starts with a novel intuition, produces a novel algorithm and provides novel empirical and theoretical support. Another interesting contact point is the second order perceptron framework (Cesa-Bianchi et al., 2005) which parallels some of the intuitions underlying the RMM. In an on-line setting, the second order perceptron maintains both a decision rule and a covariance matrix to whiten the data. The mistake bounds it inherits were shown to be better than those of the classical perceptron algorithm. Alternatively, one may consider distributions over classifier solutions which provide a different estimate than the maximum margin setting and have also shown empirical improvements over SVMs (Jaakkola et al., 1999; Herbrich et al., 2001). In recent papers, Dredze et al. (2008); Crammer et al. (2009a) consider a distribution on the perceptron hyperplane. These distribution assumptions permit update rules that resemble a whitening of the data, thus alleviating adversarial affine transformations and producing changes to the basic maximum margin formulation that are similar in spirit to those the RMM provides. In addition, recently, a new batch algorithm called the Gaussian margin machine (GMM) (Crammer et al., 2009b) has been proposed. The GMM maintains a Gaussian distribution over weight vectors for binary classification and seeks the least informative distribution that correctly classifies training data. While the GMM is well motivated from a PAC-Bayesian perspective, the optimization problem itself is expensive involving a log-determinant optimization.

Another alternative route for improving SVM performance includes the use of additional examples. For instance, test samples may be available in semi-supervised or transductive formulations of the SVM (Joachims, 1999; Belkin et al., 2005). Alternatively, additional data that does not belong to any of the classification classes of interest may be available as in the so-called Universum approach (Weston et al., 2006; Sinz et al., 2008). In principle, these methods also change the way margin is measured and the way regularization is applied to the learning problem. While additional data can be helpful in overcoming limitations for many classifiers, this article will be interested in only the simple binary classification setting. The argument is that, without any additional assumptions beyond the simple classification problem, maximizing margin in the absolute sense may be suboptimal and that maximizing relative margin is a promising alternative.

The organization of this article is as follows. Two dimensional visualization examples are given in Section 2 to show a key shortcoming of the max-margin solution. In addition, motivation is provided by considering affine transformations and the (lack of) invariance maximum margin methods exhibit to them. The relative margin formulation is detailed in Section 3 and several variants and implementations are proposed. Generalization bounds for the various function classes are studied in Section 4. Experimental results are provided in Section 5. Finally, conclusions are presented in Section 6. Some proofs and otherwise standard results are provided in the Appendix.

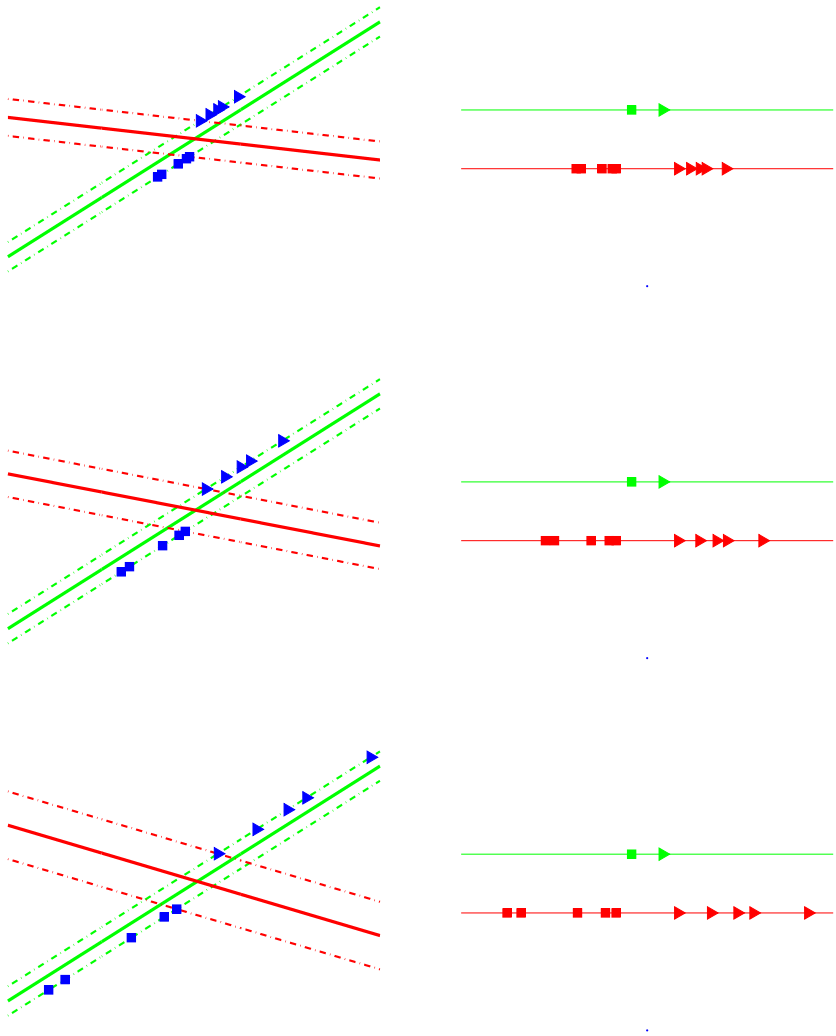

Figure 1: Left: As the data is scaled, the maximum margin SVM solution (red or dark shade) deviates from the maximum relative margin solution (green or light shade). Three different scaling scenarios are shown. Right: The projections of the examples (that is $\mathbf{w}^\top \mathbf{x} + b$) on the real line for the SVM solution (red or dark shade) and the proposed classifier (green or light shade) under each scaling scenario. These projections have been drawn on separated axes for clarity. The absolute margins for the maximum margin solution (red) are 1.24, 1.51 and 2.08 from top to bottom. For the maximum relative margin solution (green) the absolute margin is merely 0.71. However, the relative margin (the ratio of absolute margin to the spread of the projections) is 41%, 28%, and 21% for the maximum margin solution (red) and 100% for the relative margin solution (green). The scale of all axes is kept locked to permit direct visual comparison.

Notation Throughout this article, boldface letters indicate vectors/matrices. For two vectors $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^m$, $\mathbf{u} \leq \mathbf{v}$ indicates that $u_i \leq v_i$ for all i from 1 to m. $\mathbf{1}$, $\mathbf{0}$ and $\mathbf{I}$ denote the vectors of all ones, all zeros and the identity matrix respectively; $\mathbf{0}$ also denotes a matrix of all zeros in some contexts. The dimensionality of vectors and matrices should be clear from the context.

2. Motivation

This section begins with intuitive motivation for why maximizing the absolute margin can be suboptimal. In Section 2.1 a simple two dimensional example illustrates how relative margin can be a useful alternative. In Section 2.3 further support is shown for relative margin maximization by starting from an affine invariance perspective.

2.1 Intuitive motivation with a two dimensional example

Consider the simple two dimensional dataset in Figure 1 where the goal is to separate the two classes of points: triangles and squares. The figure depicts three scaled versions of the two dimensional problem to illustrate potential problems with the large margin solution.

In the topmost plot in the left column of Figure 1, two possible linear decision boundaries separating the classes are shown. The red (or dark shade) solution is the SVM estimate while the green (or light shade) solution is the proposed maximum relative margin alternative. Clearly, the SVM solution achieves the largest margin possible while separating both classes yet is this necessarily the best solution?

Next, consider the same set of points after a scaling transformation in the second and the third row of Figure 1. Note that all these three problems correspond to the same discrimination problem up to a scaling factor. With progressive scaling, the SVM increasingly deviates from the maximum relative margin solution (green), clearly indicating that the SVM decision boundary is sensitive to affine transformations of the data. Essentially, the SVM produces a family of different solutions as a result of the scaling. This sensitivity to scaling and affine transformations is worrisome. If the SVM solution and its generalization accuracy vary with scaling, an adversary may exploit such scaling to ensure that the SVM performs poorly. Meanwhile, an algorithm producing the maximum relative margin (green) decision boundary could remain resilient to adversarial scaling.

In the previous example, a direction with a small spread in the data produced a good and affine-invariant discriminator which maximized relative margin. Unlike the maximum margin solution, this solution accounts for the spread of the data in various directions. This permits it to recover a solution which has a large margin relative to the spread in that direction. Such a solution would otherwise be overlooked by a maximum margin criterion. A small margin in a correspondingly smaller spread of the data might be better than a large absolute margin with correspondingly larger data spread. This particular weakness in large margin estimation has only received limited attention in previous work.

It is helpful to consider the generative model for the above motivating example. Therein, each class was generated from a one dimensional line distribution with the two classes on two parallel lines. In this case, the maximum relative margin (green) decision boundary should obtain zero test error even if it is estimated from a finite number of samples. However, for finite training data, the SVM solution will make errors and will do so increasingly as the

data is scaled further. While it is possible to anticipate these problems and choose kernels or nonlinear mappings to correct for them in advance, this is not necessarily practical. The right mapping or kernel is never provided in advance in realistic settings. Instead, one has to estimate kernels and nonlinear mappings, a difficult endeavor which can often exacerbate the learning problem. Similarly, simple data preprocessing (affine whitening to make the dataset zero-mean and unit-covariance or scaling to place the data into a zero-one box) can also fail, possibly because of estimation problems in recovering the correct transformation (this will be shown in real-world experiments).

The above arguments show that large margin on its own is not enough; it is also necessary to control the spread of the data after projection. Therefore, maximum margin should be traded-off or balanced with the goal of simultaneously minimizing the spread of the projected data, for instance, by bounding the spread $|\mathbf{w}^\top \mathbf{x} + b|$. This will allow the linear classifier to recover large margin solutions not in the absolute sense but rather *relative to* the spread of the data in that projection direction.

In the case of a kernel such as the RBF kernel, the points are first mapped to a space so that all the input examples are unit vectors (i.e., $\langle \phi(\mathbf{x}), \phi(\mathbf{x}) \rangle = 1$). Note that the intuitive motivation proposed here still applies in such cases. No matter how they are mapped initially, a large margin solution still projects these points to the real line where the margin of separation is maximized. However, the spread of the projection can still vary significantly among the different projection directions. Given the above motivation, it is important to achieve a large margin relative to the spread of the projections even in such situations. Furthermore, experiments will support this intuition with dramatic improvements on many real problems and with a variety of kernels (including radial basis function and polynomial kernels).

2.2 Probabilistic motivation

In this subsection, we provide an informal motivation for why maximizing relative margin may be helpful. Suppose $(\mathbf{x}_i, y_i)_{i=1}^n$ are drawn independently and identically (*iid*) from a distribution $\mathcal{D}$. A classifier $\mathbf{w} \in \mathbb{R}^m$ is sought which will produce low error on future unseen examples according to the decision rule $\hat{y} = \text{sign}(y\mathbf{w}^\top \mathbf{x})$. An alternative criterion is that the classifier should produce a large value of η according to the following expression:

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}} \left[y\mathbf{w}^\top \mathbf{x} \geq 0 \right] \geq \eta,$$

where, $\mathbf{w} \in \mathbb{R}^m$ is the classifier. One way to ensure the above constraint is by requiring that the following inequality hold:

$$\mathbf{E}_{\mathcal{D}}[y\mathbf{w}^\top \mathbf{x}] \geq \sqrt{\frac{\eta}{1-\eta}} \sqrt{\mathbf{V}_{\mathcal{D}}[y\mathbf{w}^\top \mathbf{x}]}. \tag{1}$$

A proof of the above claim for a general distribution can be found in Shivaswamy et al. (2006). In fact, Gaussian margin machines (Crammer et al., 2009b) start with a similar motivation but assume a Gaussian distribution on the classifier.

According to (1), achieving a low probability of error requires the projections to have a large mean and a small variance. The mean and variance for the true distribution $\mathcal{D}$ may be

unavailable, however, the empirical counterparts of these quantities are available and known to be concentrated. The above inequality is used as a loose motivation. Instead of precisely finding low variance and high mean projections, this paper implements this intuition by trading off between large margin and small projections of the data while correctly classifying most of the examples with a hinge loss.

2.3 Motivation from an affine invariance perspective

Another motivation for maximum relative margin can be made by reformulating the classification problem altogether. Instead of learning a classifier from data, consider learning an affine transformation on data such that an a priori *fixed* classifier performs well. The data will be mapped by an affine transformation such that it is separated with large margin while it also produces a small radius. Recall that maximum margin classification and SVMs are motivated by generalization bounds based on Vapnik-Chervonenkis complexity arguments. These generalization bounds depend on the ratio of the margin to the radius of the data (Vapnik, 1995). Similarly, Rademacher generalization bounds (Shawe-Taylor and Cristianini, 2004) also consider the ratio of the trace of the kernel matrix to the margin. Here the radius of the data refers to an R such that $||\mathbf{x}|| \leq R$ for all $\mathbf{x}$ drawn from a distribution.

Instead of learning a classification rule, the optimization problem considered in this section will recover an affine transformation which achieves a large margin from a *fixed* decision rule while also achieving small radius. Assume the classification hyperplane is given a priori via the decision boundary $\mathbf{w}_0^\top \mathbf{x} + b_0 = 0$ with the two supporting margin hyperplanes $\mathbf{w}_0^\top \mathbf{x} + b_0 = \pm\rho$. Here, $\mathbf{w}_0 \in \mathbb{R}^m$ can be an arbitrary unit vector and b_0 is an arbitrary scalar. Consider the problem of mapping all the training points (by an affine transformation $\mathbf{x} \rightarrow \mathbf{A}\mathbf{x}+\mathbf{b}, \mathbf{A} \in \mathbb{R}^{m\times m}, \mathbf{b} \in \mathbb{R}^m$) so that the mapped points (i.e., $\mathbf{A}\mathbf{x}_i+\mathbf{b}$) satisfy the classification constraints $\mathbf{w}_0^\top \mathbf{x}+b_0 = \pm\rho$ while producing small radius, $\sqrt{R}$. The choice of $\mathbf{w}_0$ and b_0 is arbitrary since the affine transformation can completely compensate for it. For brevity, we denote by $\tilde{\mathbf{A}} = [\mathbf{A}\ \mathbf{b}]$ and $\tilde{\mathbf{x}} = [\mathbf{x}^\top\ 1]^\top$. With this notation, the affine transformation learning problem is formalized by the following optimization:

$$\begin{aligned} \min_{\tilde{\mathbf{A}},R,\rho} \ & -\rho + ER & (2)\\ & y_i(\mathbf{w}_0^\top \tilde{\mathbf{A}}\tilde{\mathbf{x}}_i + b_0) \geq \rho, & \forall 1 \leq i \leq n \\ & \frac{1}{2}(\tilde{\mathbf{A}}\tilde{\mathbf{x}}_i)^\top(\tilde{\mathbf{A}}\tilde{\mathbf{x}}_i) \leq R & \forall 1 \leq i \leq n. \end{aligned}$$

The parameter E trades off between the radius of the affine transformed data and the margin[2] that will be obtained. The following Lemma shows that this affine transformation learning problem is basically equivalent to learning a large margin solution with a small spread.

Lemma 1 *The solution* $\tilde{\mathbf{A}}^*$ *to* (2) *is rank one.*

2. For brevity, the so-called slack variables have been intentionally omitted since the proof holds in any case.

Proof Consider the Lagrangian of the above problem with Lagrange multipliers $\boldsymbol{\alpha}, \boldsymbol{\lambda}, \geq \mathbf{0}$:

$$\mathcal{L}(\tilde{\mathbf{A}}, \rho, R, \boldsymbol{\alpha}, \boldsymbol{\lambda}) = -\rho + ER - \sum_{i=1}^{n} \alpha_i (y_i (\mathbf{w}_0^\top \tilde{\mathbf{A}} \tilde{\mathbf{x}}_i + b_0) - \rho) + \sum_{i=1}^{n} \lambda_i (\frac{1}{2} (\tilde{\mathbf{A}} \tilde{\mathbf{x}}_i)^\top (\tilde{\mathbf{A}} \tilde{\mathbf{x}}_i) - R).$$

Differentiating the above Lagrangian with respect to $\mathbf{A}$ gives the following expression:

$$\frac{\partial \mathcal{L}(\tilde{\mathbf{A}}, \rho, R, \alpha, \lambda)}{\partial \tilde{\mathbf{A}}} = -\sum_{i=1}^{n} \alpha_i y_i \mathbf{w}_0 \tilde{\mathbf{x}}_i^\top + \tilde{\mathbf{A}} \sum_{i=1}^{n} \lambda_i \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^\top. \tag{3}$$

From (3), at optimum,

$$\tilde{\mathbf{A}}^* \sum_{i=1}^{n} \lambda_i \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^\top = -\sum_{i=1}^{n} \alpha_i y_i \mathbf{w}_0 \tilde{\mathbf{x}}_i^\top.$$

It is therefore clear that $\tilde{\mathbf{A}}^*$ can always be chosen to have rank one since the right hand side of the expression is just an outer product of two vectors. ■

Lemma 1 gives further intuition on why one should limit the spread of the recovered classifier. Learning a transformation matrix $\tilde{\mathbf{A}}$ so as to maximize the margin while minimizing the radius given an a priori hyperplane $(\mathbf{w}_0, b_0)$ is no different from learning a classification hyperplane $(\mathbf{w}, b)$ with a large margin as well as a small spread. This is because the rank of the affine transformation $\tilde{\mathbf{A}}^*$ is one; thus, $\tilde{\mathbf{A}}^*$ merely maps all the points $\tilde{\mathbf{x}}_i$ onto a line achieving a certain margin ρ but also limiting the output or spread. This means that finding an affine transformation which achieves a large margin and small radius is equivalent to finding a $\mathbf{w}$ and b with a large margin and with projections constrained to remain close to the origin. Thus, the affine transformation learning problem complements the intuitive arguments in Section 2.1 and also suggests that the learning algorithm should bound the spread of the data.

3. From absolute margin to relative margin

This section will provide an upgrade path from the maximum margin classifier (or SVM) to a maximum relative margin formulation. Given independent identically distributed samples $(\mathbf{x}_i, y_i)_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^m$ and $y_i \in \{\pm 1\}$ are drawn from $\Pr(\mathbf{x}, y)$, the support vector machine primal formulation is as follows:

$$\begin{aligned} &\min_{\mathbf{w}, b, \xi} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i \\ &\text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \qquad\qquad \forall 1 \leq i \leq n. \end{aligned} \tag{4}$$

The above is an easily solvable quadratic program (QP) and maximizes the margin by minimizing $\|\mathbf{w}\|^2$. Since real data is seldom separable, slack variables (ξ_i) are used to relax

the hard classification constraints. Thus, the above formulation maximizes the margin while minimizing an upper bound on the number of classification errors. The trade-off between the two quantities is controlled by the parameter C. Equivalently, the following dual of the formulation (4) can be solved:

$$\begin{aligned} \max_{\alpha} & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ \text{s.t.} & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C \qquad \forall 1 \leq i \leq n. \end{aligned} \tag{5}$$

Lemma 2 *The formulation in* (5) *is invariant to a rotation of the inputs.*

Proof Replace each $\mathbf{x}_i$ with $\mathbf{A}\mathbf{x}_i$ where $\mathbf{A}$ is a rotation matrix such that $\mathbf{A} \in \mathbb{R}^{m \times m}$ and $\mathbf{A}^\top \mathbf{A} = \mathbf{I}$. It is clear that the dual remains the same. ■

However, the dual is not the same if $\mathbf{A}$ is more general than a rotation matrix, for instance, if it is an arbitrary affine transformation.

The above classification framework can also handle non-linear classification readily by making use of Mercer kernels. A kernel function $k : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ replaces the dot products $\mathbf{x}_i^\top \mathbf{x}_j$ in (5). The kernel function k is such that $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$, where $\phi : \mathbb{R}^m \to \mathcal{H}$ is a mapping to a Hilbert space. Thus, solving the SVM dual formulation (5) with a kernel function can give a non-linear solution in the input space. In the rest of this article, $\mathbf{K} \in \mathbb{R}^{n \times n}$ denotes the Gram matrix whose individual entries are given by $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. When applying Lemma 2 on a kernel defined feature space, the affine transformation is on $\phi(\mathbf{x}_i)$ and not on $\mathbf{x}_i$.

3.1 The whitened SVM

One way of limiting sensitivity to affine transformations while recovering a large margin solution is to whiten the data with the covariance matrix prior to estimating the SVM solution. This may also reduce the bias towards regions of large data spread as discussed in Section 2. Denote by

$$\boldsymbol{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top - \frac{1}{n^2} \sum_{i=1}^{n} \mathbf{x}_i \sum_{j=1}^{n} \mathbf{x}_j^\top, \quad \text{and} \quad \boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i,$$

the sample covariance and sample mean, respectively. Now, consider the following formulation called $\boldsymbol{\Sigma}$-SVM:

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} & \frac{1-D}{2} \|\mathbf{w}\|^2 + \frac{D}{2} \|\boldsymbol{\Sigma}^{\frac{1}{2}} \mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i \\ \text{s.t.} & \; y_i (\mathbf{w}^\top (\mathbf{x}_i - \boldsymbol{\mu}) + b) \geq 1 - \xi_i, \; \xi_i \geq 0 \qquad \forall 1 \leq i \leq n \end{aligned} \tag{6}$$

where $0 \leq D \leq 1$ is an additional parameter that trades off between the two regularization terms. When $D = 0$, (6) gives back the usual SVM primal (although on translated data).

When $D = 1$, the regularization is entirely on the transformed space. The dual of (6) can be shown to be:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \alpha_i y_i (\mathbf{x}_i - \boldsymbol{\mu})^\top ((1-D)\mathbf{I} + D\boldsymbol{\Sigma})^{-1} \sum_{j=1}^{n} \alpha_j y_j (\mathbf{x}_j - \boldsymbol{\mu}) \tag{7}$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0$$
$$0 \leq \alpha_i \leq C \qquad \forall 1 \leq i \leq n.$$

It is easy to see that the above formulation (7) is translation invariant and tends to an affine invariant solution when D tends to one. However, there are some problems with this formulation. First, the whitening process only considers second order statistics of the input data which may be inappropriate for non-Gaussian datasets. Furthermore, there are computational difficulties associated with whitening. Consider the following term:

$$(\mathbf{x}_i - \boldsymbol{\mu})^\top ((1-D)\mathbf{I} + D\boldsymbol{\Sigma})^{-1} (\mathbf{x}_j - \boldsymbol{\mu}).$$

When $0 < D < 1$, it can be shown, by using the Woodbury matrix inversion formula, that the above term can be kernelized as

$$\hat{k}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{1-D} \left(k(\mathbf{x}_i, \mathbf{x}_j) - \frac{\mathbf{K}_i^\top \mathbf{1}}{n} - \frac{\mathbf{K}_j^\top \mathbf{1}}{n} + \frac{\mathbf{1}^\top \mathbf{K} \mathbf{1}}{n^2} \right)$$
$$- \frac{1}{1-D} \left(\left(\mathbf{K}_i - \frac{\mathbf{K}\mathbf{1}}{n} \right)^\top \left(\frac{\mathbf{I}}{n} - \frac{\mathbf{1}\mathbf{1}^\top}{n^2} \right) \left[\frac{1-D}{D} \mathbf{I} + \mathbf{K} \left(\frac{\mathbf{I}}{n} - \frac{\mathbf{1}\mathbf{1}^\top}{n^2} \right) \right]^{-1} \left(\mathbf{K}_j - \frac{\mathbf{K}\mathbf{1}}{n} \right) \right),$$

where $\mathbf{K}_i$ is the i^{th} column of $\mathbf{K}$. This implies that the $\boldsymbol{\Sigma}$-SVM can be solved merely by solving (5) after replacing the kernel with $\hat{k}(\mathbf{x}_i, \mathbf{x}_j)$ as defined above. Note that the above formula involves a matrix inversion of size n, making the kernel computation alone $\mathcal{O}(n^3)$. Even performing whitening as a preprocessing step in the feature space would involve this matrix inversion which is often computationally prohibitive.

3.2 Relative margin machines

While the above $\boldsymbol{\Sigma}$-SVM does address some of the issues of data spread, it made second order assumptions to recover $\boldsymbol{\Sigma}$ and involved a cumbersome matrix inversion. A more direct and efficient approach to control the spread is possible and will be proposed next.

The SVM will be modified such that the projections on the training examples remain bounded. A parameter will also be introduced that helps trade off between large margin and small spread of the projection of the data. This formulation will initially be solved by a quadratically constrained quadratic program (QCQP) in this section. The dual of this formulation will also be of interest and yield further geometric intuitions.

Consider the following formulation called the relative margin machine (RMM):

$$\begin{aligned} &\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i && (8)\\ &\text{s.t.} y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 && \forall 1 \leq i \leq n \\ &\quad \frac{1}{2}(\mathbf{w}^\top \mathbf{x}_i + b)^2 \leq \frac{B^2}{2} && \forall 1 \leq i \leq n. \end{aligned}$$

This formulation is similar to the SVM primal (4) except for the additional constraints $\frac{1}{2}(\mathbf{w}^\top \mathbf{x}_i + b)^2 \leq \frac{B^2}{2}$. The formulation has one extra parameter B in addition[3] to the SVM parameter C. When B is large enough, the above QCQP gives the same solution as the SVM. Also note that only settings of $B > 1$ are meaningful since a value of B less than one would prevent any training examples from clearing the margin, i.e., none of the examples could satisfy $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$ otherwise. Let $\mathbf{w}_C$ and b_C be the solutions obtained by solving the SVM (4) for a particular value of C. It is clear, then, that $B > \max_i |\mathbf{w}_C^\top \mathbf{x}_i + b_C|$, makes the constraint on the second line in the formulation (8) inactive for each i and the solution obtained is the same as the SVM estimate. This gives an upper threshold for the parameter B so that the RMM solution is not trivially identical to the SVM solution.

As B is decreased, the RMM solution increasingly differs from the SVM solution. Specifically, with a smaller B, the RMM still finds a large margin solution but with a smaller projection of the training examples. By trying different B values (within the aforementioned thresholds), different large relative margin solutions are explored. It is helpful to next consider the dual of the RMM problem.

The Lagrangian of (8) is given by:

$$\begin{aligned} \mathcal{L}(\mathbf{w}, b, \alpha, \lambda, \beta) =& \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 + \xi_i\right) - \sum_{i=1}^{n}\beta_i\xi_i \\ &+ \sum_{i=1}^{n}\lambda_i\left(\frac{1}{2}(\mathbf{w}^\top \mathbf{x}_i + b)^2 - \frac{1}{2}B^2\right), \end{aligned}$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda} \geq 0$ are the Lagrange multipliers corresponding to the constraints. Differentiating with respect to the primal variables and equating to zero produces:

$$\begin{aligned} &(\mathbf{I} + \sum_{i=1}^{n}\lambda_i \mathbf{x}_i \mathbf{x}_i^\top)\mathbf{w} - b\sum_{i=1}^{n}\lambda_i \mathbf{x}_i = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i, \\ &\frac{1}{\boldsymbol{\lambda}^\top \mathbf{1}}(\sum_{i=1}^{n}\alpha_i y_i - \sum_{i=1}^{n}\lambda_i \mathbf{w}^\top \mathbf{x}_i) = b, \\ &\qquad\qquad \alpha_i + \beta_i = C \qquad\qquad \forall 1 \leq i \leq n. \end{aligned}$$

Denoting by

$$\boldsymbol{\Sigma}_{\boldsymbol{\lambda}} = \sum_{i=1}^{n}\lambda_i \mathbf{x}_i \mathbf{x}_i^\top - \frac{1}{\boldsymbol{\lambda}^\top \mathbf{1}}\sum_{i=1}^{n}\lambda_i \mathbf{x}_i \sum_{j=1}^{n}\lambda_j \mathbf{x}_j^\top, \quad \text{and} \quad \boldsymbol{\mu}_{\boldsymbol{\lambda}} = \frac{1}{\boldsymbol{\lambda}^\top \mathbf{1}}\sum_{i=1}^{n}\lambda_i \mathbf{x}_i,$$

3. It is possible to relax the constraints $\frac{1}{2}(\mathbf{w}^\top \mathbf{x}_i + b)^2 \leq \frac{1}{2}B^2$ with slack variables at the expense of one additional parameter. However, this will not be investigated in this article.

the dual of (8) can be shown to be:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\lambda}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \alpha_i y_i (\mathbf{x}_i - \boldsymbol{\mu}_{\boldsymbol{\lambda}})^\top (\mathbf{I} + \boldsymbol{\Sigma}_{\boldsymbol{\lambda}})^{-1} \sum_{j=1}^{n} \alpha_j y_j (\mathbf{x}_j - \boldsymbol{\mu}_{\boldsymbol{\lambda}}) + \frac{1}{2} B^2 \sum_{i=1}^{n} \lambda_i \tag{9}$$
$$\text{s.t. } 0 \leq \alpha_i \leq C \quad \lambda_i \geq 0 \qquad \forall 1 \leq i \leq n.$$

Moreover, the optimal $\mathbf{w}$ can be shown to be:

$$\mathbf{w} = (\mathbf{I} + \boldsymbol{\Sigma}_{\boldsymbol{\lambda}})^{-1} \sum_{i=1}^{n} \alpha_i y_i (\mathbf{x}_i - \boldsymbol{\mu}_{\boldsymbol{\lambda}}). \tag{10}$$

Note that the above formulation is translation invariant since $\boldsymbol{\mu}_{\boldsymbol{\lambda}}$ is subtracted from each $\mathbf{x}_i$. $\boldsymbol{\Sigma}_{\boldsymbol{\lambda}}$ corresponds to a shape matrix (which is potentially low rank) determined by $\mathbf{x}_i$'s that have non-zero λ_i. From the Karush-Kuhn-Tucker (KKT) conditions of (8) it is clear that $\lambda_i(\frac{1}{2}(\mathbf{w}^\top \mathbf{x}_i + b)^2 - \frac{B^2}{2}) = 0$. Consequently $\lambda_i > 0$ implies $(\frac{1}{2}(\mathbf{w}^\top \mathbf{x}_i + b)^2 - \frac{B^2}{2}) = 0$. Notice the similarity in the two dual formulations in (7) and (9); both formulations look similar except for the choice of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ which transform the inputs. The RMM in (9) whitens data with the matrix $(\mathbf{I} + \boldsymbol{\Sigma}_{\boldsymbol{\lambda}})$ while simultaneously solving an SVM-like classification problem. While this is similar in spirit to the $\boldsymbol{\Sigma}$-SVM, the matrix $(\mathbf{I} + \boldsymbol{\Sigma}_{\boldsymbol{\lambda}})$ is being estimated directly to optimize the margin with a small data spread. The $\boldsymbol{\Sigma}$-SVM only whitens data as a preprocessing independently of the margin and the labels. The $\boldsymbol{\Sigma}$-SVM is equivalent to the RMM only in the rare situation when all $\lambda_i = t$ for some t which makes the $\boldsymbol{\mu}_{\boldsymbol{\lambda}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\lambda}}$ in the RMM and $\boldsymbol{\Sigma}$-SVM identical up to a scaling factor.

In practice, the above formulation will not be solved since it is computationally impractical. Solving (9) requires semi-definite programming (SDP) which prevents the method from scaling beyond a few hundred data points. Instead, an equivalent optimization will be used which gives the same solution but only requires quadratic programming. This is achieved by simply replacing the constraint $\frac{1}{2}(\mathbf{w}^\top \mathbf{x}_i + b)^2 \leq \frac{1}{2} B^2$ with the two equivalent linear constraints: $(\mathbf{w}^\top \mathbf{x}_i + b) \leq B$ and $-(\mathbf{w}^\top \mathbf{x}_i + b) \leq B$. With these linear constraints replacing the quadratic constraint, the problem is now merely a QP. In the primal, the QP has $4n$ constraints (including $\boldsymbol{\xi} \geq 0$) instead of the $2n$ constraints in the SVM. Thus, the RMM's quadratic program has the same order of complexity as the SVM. In the next section, an efficient implementation of the RMM problem is presented.

3.3 Fast implementation

Once the quadratic constraints have been replaced with linear constraints, the RMM is merely a quadratic program which admits many fast implementation schemes. It is now possible to adapt previous fast SVM algorithms in the literature to the RMM. In this section, the SVM^{light} (Joachims, 1998) approach will be adapted to the following RMM

optimization problem

$$\begin{aligned}
&\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i && && (11)\\
&\text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1-\xi_i, \quad \xi_i \geq 0 && \forall 1 \leq i \leq n\\
&\quad \mathbf{w}^\top \mathbf{x}_i + b \leq B && \forall 1 \leq i \leq n\\
&\quad -\mathbf{w}^\top \mathbf{x}_i - b \leq B && \forall 1 \leq i \leq n.
\end{aligned}$$

The dual of (11) can be shown to be the following:

$$\begin{aligned}
&\max_{\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\lambda}^*} -\frac{1}{2}\left(\boldsymbol{\alpha}\bullet\mathbf{y}-\boldsymbol{\lambda}+\boldsymbol{\lambda}^*\right)^\top \mathbf{K}\left(\boldsymbol{\alpha}\bullet\mathbf{y}-\boldsymbol{\lambda}+\boldsymbol{\lambda}^*\right)+\boldsymbol{\alpha}^\top\mathbf{1}-B\boldsymbol{\lambda}^\top\mathbf{1}-B\boldsymbol{\lambda}^{*\top}\mathbf{1} && (12)\\
&\text{s.t. } \boldsymbol{\alpha}^\top\mathbf{y}-\boldsymbol{\lambda}^\top\mathbf{1}+\boldsymbol{\lambda}^{*\top}\mathbf{1}=0\\
&\quad 0\leq\boldsymbol{\alpha}\leq C\mathbf{1}\\
&\quad \boldsymbol{\lambda},\boldsymbol{\lambda}^*\geq\mathbf{0},
\end{aligned}$$

where, the operator $\bullet$ denotes the element-wise product of two vectors.

The QP in (12) is solved in an iterative way. In each step, only a subset of the dual variables are optimized. For instance, in a particular iteration, take q, r and s ($\tilde{q}$, $\tilde{r}$ and $\tilde{s}$) to be indices of the free (fixed) variables in $\boldsymbol{\alpha}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^*$ respectively (ensuring that $q\cup\tilde{q}=\{1,2,\dots n\}$ and $q\cap\tilde{q}=\emptyset$ and proceeding similarly for the other two indices). The optimization over the free variables in that step can then be expressed as:

$$\begin{aligned}
\max_{\boldsymbol{\alpha}_q,\boldsymbol{\lambda}_r,\boldsymbol{\lambda}_s^*} & -\frac{1}{2}\begin{bmatrix}\boldsymbol{\alpha}_q\bullet\mathbf{y}_q\\ \boldsymbol{\lambda}_r\\ \boldsymbol{\lambda}_s^*\end{bmatrix}^\top\begin{bmatrix}\mathbf{K}_{qq} & -\mathbf{K}_{qr} & \mathbf{K}_{qs}\\ -\mathbf{K}_{rq} & \mathbf{K}_{rr} & -\mathbf{K}_{rs}\\ \mathbf{K}_{sq} & -\mathbf{K}_{sr} & \mathbf{K}_{ss}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_q\bullet\mathbf{y}_q\\ \boldsymbol{\lambda}_r\\ \boldsymbol{\lambda}_s^*\end{bmatrix} && (13)\\
& -\frac{1}{2}\begin{bmatrix}\boldsymbol{\alpha}_q\bullet\mathbf{y}_q\\ \boldsymbol{\lambda}_r\\ \boldsymbol{\lambda}_s^*\end{bmatrix}^\top\begin{bmatrix}\mathbf{K}_{q\tilde{q}} & -\mathbf{K}_{q\tilde{r}} & \mathbf{K}_{q\tilde{s}}\\ -\mathbf{K}_{r\tilde{q}} & \mathbf{K}_{r\tilde{r}} & -\mathbf{K}_{r\tilde{s}}\\ \mathbf{K}_{s\tilde{q}} & -\mathbf{K}_{s\tilde{r}} & \mathbf{K}_{s\tilde{s}}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_{\tilde{q}}\bullet\mathbf{y}_{\tilde{q}}\\ \boldsymbol{\lambda}_{\tilde{r}}\\ \boldsymbol{\lambda}_{\tilde{s}}^*\end{bmatrix}\\
& +\boldsymbol{\alpha}_q^\top\mathbf{1}-B\boldsymbol{\lambda}_r^\top\mathbf{1}-B\boldsymbol{\lambda}_s^{*\top}\mathbf{1}\\
\text{s.t. } & \boldsymbol{\alpha}_q^\top y_q-\boldsymbol{\lambda}_r^\top\mathbf{1}+\boldsymbol{\lambda}_s^{*\top}\mathbf{1}=-\boldsymbol{\alpha}_{\tilde{q}}^\top y_{\tilde{q}}+\boldsymbol{\lambda}_{\tilde{r}}^\top\mathbf{1}-\boldsymbol{\lambda}_{\tilde{s}}^{*\top}\mathbf{1},\\
& \mathbf{0}\leq\boldsymbol{\alpha}_q\leq C\mathbf{1},\\
& \boldsymbol{\lambda}_r,\ \boldsymbol{\lambda}_s^*\geq\mathbf{0}.
\end{aligned}$$

While the first term in the above objective is quadratic in the free variables (over which it is optimized), the second term is merely linear. Essentially, the above is a working-set scheme which iteratively solves the QP over subsets of variables until some termination criteria are achieved. The following enumerates the termination criteria that will be used in this article. If $\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\lambda}^*$ and b are the current solution (b is determined by the KKT conditions just as

with SVMs), then:

$$\forall i \text{ s.t. } 0 < \alpha_i < C : \quad b - \epsilon \leq y_i - (\sum_{j=1}^{n} (\alpha_j y_j - \lambda_j + \lambda_j^*) k(\mathbf{x}_i, \mathbf{x}_j)) \leq b + \epsilon$$

$$\forall i \text{ s.t. } \alpha_i = 0 : \quad y_i (\sum_{j=1}^{n} (\alpha_j y_j - \lambda_j + \lambda_j^*) k(\mathbf{x}_i, \mathbf{x}_j) + b) \geq 1 - \epsilon$$

$$\forall i \text{ s.t. } \alpha_i = C : \quad y_i (\sum_{j=1}^{n} (\alpha_j y_j - \lambda_j + \lambda_j^*) k(\mathbf{x}_i, \mathbf{x}_j) + b) \leq 1 + \epsilon$$

$$\forall i \text{ s.t. } \lambda_i > 0 : \quad B - \epsilon \leq (\sum_{j=1}^{n} (\alpha_j y_j - \lambda_j + \lambda_j^*) k(\mathbf{x}_i, \mathbf{x}_j) + b) \leq B + \epsilon$$

$$\forall i \text{ s.t. } \lambda_i = 0 : \quad (\sum_{j=1}^{n} (\alpha_j y_j - \lambda_j + \lambda_j^*) k(\mathbf{x}_i, \mathbf{x}_j) + b) \leq B - \epsilon$$

$$\forall i \text{ s.t. } \lambda_i^* > 0 : \quad B - \epsilon \leq -(\sum_{j=1}^{n} (\alpha_j y_j - \lambda_j + \lambda_j^*) k(\mathbf{x}_i, \mathbf{x}_j) + b) \leq B + \epsilon$$

$$\forall i \text{ s.t. } \lambda_i^* = 0 : \quad -(\sum_{j=1}^{n} (\alpha_j y_j - \lambda_j + \lambda_j^*) k(\mathbf{x}_i, \mathbf{x}_j) + b) \leq B - \epsilon.$$

In each step of the algorithm, a small sub-problem of the structure of (13) is solved. To select the free variables, these conditions are checked to find the worst violating variables both from the top of the violation list and from the bottom. The selected variables are optimized by solving (13) while keeping the other variables fixed. Since only a small QP is solved in each step, the cubic time scaling behavior is circumvented for improved efficiency. A few other book-keeping tricks have also been adapted from SVM^{light} to yield other minor improvements.

Denote by p the number of elements chosen in each step of the optimization (i.e., $p = |q| + |r| + |s|$). The QP in each step takes $\mathcal{O}(p^3)$ and updating the prediction values to compute the KKT violations takes $\mathcal{O}(nq)$ time. Sorting the output values to choose the most violated constraints takes $\mathcal{O}(n \log(n))$ time. Thus, the total time taken in each iteration of the algorithm is $\mathcal{O}(p^3 + n \log(n) + nq)$. Empirical running times are provided in Section 5 for a digit classification problem.

Many other fast SVM solvers could also be adapted to the RMM. Recent advances such as the cutting plane SVM algorithm (Joachims, 2006), Pegasos (Shalev-Shwartz et al., 2007) and so forth are also applicable and are deferred for future work.

3.4 Variants of the RMM

It is not always desirable to have a parameter in a formulation that would depend explicitly on the output from a previous computation as in (11). It is possible to overcome this issue via the following optimization problem:

$$\begin{aligned}
\min_{\mathbf{w},b,\xi,t\geq 1} \ & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i + Dt & (14)\\
\text{s.t.} & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1-\xi_i, \quad \xi_i \geq 0 & \forall 1\leq i\leq n,\\
& +(\mathbf{w}^\top\mathbf{x}_i + b) \leq t & \forall 1\leq i\leq n,\\
& -(\mathbf{w}^\top\mathbf{x}_i + b) \leq t & \forall 1\leq i\leq n.
\end{aligned}$$

Note that (14) has a parameter D instead of the parameter B in (11). The two optimization problems are equivalent in the sense that for every value of B in (11), it is possible to have a corresponding D such that both optimization problems give the same solution.

Further, in some situations, a hard constraint bounding the outputs as in (14) can be detrimental due to outliers. Thus, it might be required to have a relaxation on the bounding constraints as well. This motivates the following relaxed version of (14):

$$\begin{aligned}
\min_{\mathbf{w},b,\xi,t\geq 1} \ & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i + D(t+\frac{\nu}{n}\sum_{i=1}^{n}(\tau_i+\tau_i^*)) & (15)\\
\text{s.t.} & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1-\xi_i, \quad \xi_i \geq 0 & \forall 1\leq i\leq n,\\
& +(\mathbf{w}^\top\mathbf{x}_i + b) \leq t+\tau_i & \forall 1\leq i\leq n,\\
& -(\mathbf{w}^\top\mathbf{x}_i + b) \leq t+\tau_i^* & \forall 1\leq i\leq n.
\end{aligned}$$

In the above formulation, ν controls the fraction of outliers. It is not hard to derive the dual of the above to express it in kernelized form.

4. Risk bounds

This section provides generalization guarantees for the classifiers of interest (the SVM, $\mathbf{\Sigma}$-SVM and RMM) which all produce decision[4] boundaries of the form $\mathbf{w}^\top\mathbf{x} = 0$ from a limited number of examples. In the SVM, the decision boundary is found by minimizing a combination of $\mathbf{w}^\top\mathbf{w}$ and an upper bound on the number of errors. This minimization is equivalent to choosing a function $g(\mathbf{x}) = \mathbf{w}^\top\mathbf{x}$ from a set of linear functions with bounded 2-norm. Therefore, with a suitable choice of E, the SVM solution chooses the function $g(\cdot)$ from the set $\{\mathbf{x}\rightarrow\mathbf{w}^\top\mathbf{x}|\frac{1}{2}\mathbf{w}^\top\mathbf{w}\leq E\}$.

By measuring the complexity of the function class being explored, it is possible to derive generalization guarantees and risk bounds. A natural measure of how complex a function class is the Rademacher complexity which has been fruitful in the derivation of generalization bounds. For SVMs, such results can be found in Shawe-Taylor and Cristianini (2004). This section continues in the same spirit and defines the function classes and their corresponding Rademacher complexities for slightly modified versions of the RMM as well as the $\mathbf{\Sigma}$-SVM. Furthermore, these will be used to provide generalization guarantees for both classifiers. The style and content of this section closely follows that of Shawe-Taylor and Cristianini (2004).

4. The bias term is suppressed in this section for brevity.

The function classes for the RMM and $\mathbf{\Sigma}$-SVM will depend on the data. Thus, these both entail so-called data-dependent regularization which is not quite as straightforward as the function classes explored by SVMs. In particular, the data involved in defining data-dependent function classes will be treated differently and referred to as landmarks to distinguish them from the training data. Landmark data is used to define the function class while training data is used to select a specific function from the class. This distinction is important for the following theoretical derivations. However, in practical implementations, both the $\mathbf{\Sigma}$-SVM and the RMM may use the training data to both define the function class and to choose the best function within it. Thus, the distinction between landmark data and training data is merely a formality for deriving generalization bounds which require independent sets of samples for both stages. Ultimately, however, it will be possible to still provide generalization guarantees that are independent of the particular landmark samples. Details of this argument are provided in Section 4.6. For this section, however, it is assumed that, in parallel with the training data, a separate dataset of landmarks is provided to define the function class for the RMM and the $\mathbf{\Sigma}$-SVM.

4.1 Function class definitions

Consider the training data set $(\mathbf{x}_i, y_i)_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^m$ and $y_i \in \{\pm 1\}$ which are drawn independently and identically distributed (*iid*) from an unknown underlying distribution $\mathbf{P}[(\mathbf{x}, y)]$ denoted as $\mathcal{D}$. The features of the training examples above are denoted by the set $\mathbf{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$.

Given a choice of the parameter E in the SVM (where E plays the role of the regularization parameter), the set of linear functions the SVM considers is:

Definition 3 $\mathcal{F}_E := \{\mathbf{x} \to \mathbf{w}^\top \mathbf{x} | \frac{1}{2}\mathbf{w}^\top \mathbf{w} \leq E\}$.

The RMM maximizes the margin while also limiting the spread of projections on the training data. It effectively considers the following function class:

Definition 4 $\mathcal{H}^S_{E,D} := \{\mathbf{x} \to \mathbf{w}^\top \mathbf{x} | \frac{\bar{D}}{2}\mathbf{w}^\top \mathbf{w} + \frac{D}{2}(\mathbf{w}^\top \mathbf{x}_i)^2 \leq E \ \forall 1 \leq i \leq n\}$.

Above, take $\bar{D} := 1 - D$ and $0 < D < 1$ trades off between large margin and small spread on the projections[5]. Since the above function class depends on the training examples, standard Rademacher analysis, which is straightforward for the SVM, is no longer applicable. Instead, define another function class for the RMM using a distinct set of landmark examples.

A set $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_{n_v}\}$ drawn *iid* from the same distribution $\mathbf{P}[\mathbf{x}]$, denoted as $\mathcal{D}_x$, is used as the landmark examples. With these landmark examples, the modified RMM function class can be written as:

Definition 5 $\mathcal{H}^V_{E,D} := \{\mathbf{x} \to \mathbf{w}^\top \mathbf{x} | \frac{\bar{D}}{2}\mathbf{w}^\top \mathbf{w} + \frac{D}{2}(\mathbf{w}^\top \mathbf{v}_i)^2 \leq E \ \forall 1 \leq i \leq n_v\}$.

Finally, function classes that are relevant for the $\mathbf{\Sigma}$-SVM are considered. These limit the average projection rather than the maximum projection. The data-dependent function class is defined as below:

Definition 6 $\mathcal{G}^S_{E,D} := \{\mathbf{x} \to \mathbf{w}^\top \mathbf{x} | \frac{\bar{D}}{2}\mathbf{w}^\top \mathbf{w} + \frac{D}{2n}\sum_{i=1}^n (\mathbf{w}^\top \mathbf{x}_i)^2 \leq E\}$.

5. Zero and one are excluded from the range of D to avoid degenerate cases.

The corresponding landmark function class is defined using a different landmark set $\mathbf{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ again drawn *iid* from $\mathcal{D}_x$ as follows:

Definition 7 $\mathcal{G}^U_{B,D} := \{\mathbf{x} \to \mathbf{w}^\top \mathbf{x} | \frac{\bar{D}}{2}\mathbf{w}^\top \mathbf{w} + \frac{D}{2n}\sum_{i=1}^n (\mathbf{w}^\top \mathbf{u}_i)^2 \leq B\}$.

Note that the parameter E is fixed in $\mathcal{H}^V_{E,D}$ but n_v may be different from n. In the case of $\mathcal{G}^U_{B,D}$, the number of landmarks is the same (n) as the number of training examples but the parameter B is used instead of E. These distinctions are intentional and will be clarified in subsequent sections.

4.2 Rademacher complexity

In this section the Rademacher complexity of the aforementioned function classes are quantified by bounding the empirical Rademacher complexity. Rademacher complexity measures the richness of a class of real-valued functions with respect to a probability distribution (Bartlett and Mendelson, 2002; Shawe-Taylor and Cristianini, 2004; Bousquet et al., 2004).

Definition 8 *For a sample* $\mathbf{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ *generated by a distribution on* $\mathbf{x}$ *and a real-valued function class* $\mathcal{F}$ *with domain* $\mathbf{x}$*, the empirical Rademacher complexity*[6] *of* $\mathcal{F}$ *is*

$$\hat{R}(\mathcal{F}) := \mathbf{E}_\sigma \left[\sup_{f \in \mathcal{F}} \left| \frac{2}{n} \sum_{i=1}^n \sigma_i f(\mathbf{x}_i) \right| \right]$$

where $\sigma = \{\sigma_1, \ldots \sigma_n\}$ *are independent random variables that take values* $+1$ *or* -1 *with equal probability. Moreover, the Rademacher complexity of* $\mathcal{F}$ *is:* $R(\mathcal{F}) := \mathbf{E}_\mathbf{S}\left[\hat{R}(\mathcal{F})\right]$.

A stepping stone for quantifying the true Rademacher complexity is obtained by considering its empirical counterpart.

4.3 Empirical Rademacher complexity

In this subsection, upper bounds on the empirical Rademacher complexities are derived for the previously defined function classes. These bounds provide insights on the regularization properties of the function classes for the sample $\mathbf{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n\}$.

Theorem 9 $\hat{R}(\mathcal{F}_E) \leq T_0 := \frac{2\sqrt{2E}}{n}\sqrt{tr(\mathbf{K})}$, *where* $tr(\mathbf{K})$ *is the trace of the Gram matrix of the elements in* $\mathbf{S}$.

6. The dependence of the empirical Rademacher complexity on n and $\mathbf{S}$ is suppressed by writing $\hat{R}(\mathcal{F})$ for brevity.

Proof

$$\hat{R}(\mathcal{F}_E) = \mathbf{E}_\sigma\left[\sup_{f\in\mathcal{F}_E}\left|\frac{2}{n}\sum_{i=1}^{n}\sigma_i f(\mathbf{x}_i)\right|\right] = \frac{2}{n}\mathbf{E}_\sigma\left[\max_{||\mathbf{w}||\leq\sqrt{2E}}\left|\mathbf{w}^\top\sum_{i=1}^{n}\sigma_i\mathbf{x}_i\right|\right]$$
$$\leq \frac{2\sqrt{2E}}{n}\mathbf{E}_\sigma\left[\left|\left|\sum_{i=1}^{n}\sigma_i\mathbf{x}_i\right|\right|\right] = \frac{2\sqrt{2E}}{n}\mathbf{E}_\sigma\left[\left(\sum_{i=1}^{n}\sigma_i\mathbf{x}_i^\top\sum_{j=1}^{n}\sigma_j\mathbf{x}_j\right)^{\frac{1}{2}}\right]$$
$$\leq \frac{2\sqrt{2E}}{n}\left(\mathbf{E}_\sigma\left[\sum_{i,j=1}^{n}\sigma_i\sigma_j\mathbf{x}_i^\top\mathbf{x}_j\right]\right)^{\frac{1}{2}} = \frac{2\sqrt{2E}}{n}\sqrt{\mathrm{tr}(\mathbf{K})}.$$

The proof uses Jensen's inequality on the function $\sqrt{\cdot}$ and the fact that σ_i and σ_j are random variables taking values $+1$ or -1 with equal probability. Thus, when $i \neq j$, we have $\mathbf{E}_\sigma[\sigma_i\sigma_j\mathbf{x}_i^\top\mathbf{x}_i] = 0$ and, otherwise, $\mathbf{E}_\sigma[\sigma_i\sigma_i\mathbf{x}_i^\top\mathbf{x}_i] = \mathbf{E}_\sigma[\mathbf{x}_i^\top\mathbf{x}_i] = \mathbf{x}_i^\top\mathbf{x}_i$. The result follows from the linearity of the expectation operator. ■

Roughly speaking, by keeping E small, the classifier's ability to fit arbitrary labels is reduced. This is one way to motivate a maximum margin strategy. Note that $\sqrt{\mathrm{tr}(\mathbf{K})}$ is a coarse measure of the spread of the data. However, most SVM formulations do not directly optimize this term. This motivates us to next consider two new function classes.

Theorem 10 *$\hat{R}(\mathcal{H}^V_{E,D}) \leq T_2(\mathbf{V},\mathbf{S})$ where, for any training set $\mathcal{B}$ and landmark[7] set $\mathcal{A}$, $T_2(\mathcal{A},\mathcal{B}) := \min_{\lambda\geq 0}\frac{1}{|\mathcal{B}|}\sum_{\mathbf{x}\in\mathcal{B}}\mathbf{x}^\top\left(\bar{D}\mathbf{I}\sum_{\mathbf{u}\in\mathcal{A}}\lambda_\mathbf{u} + D\sum_{\mathbf{u}\in\mathcal{A}}\lambda_\mathbf{u}\mathbf{u}\mathbf{u}^\top\right)^{-1}\mathbf{x} + \frac{2E}{|\mathcal{B}|}\sum_{\mathbf{u}\in\mathcal{A}}\lambda_\mathbf{u}$.*

Proof Start with the definition of the empirical Rademacher complexity:

$$\hat{R}(\mathcal{H}^V_{E,D}) = \mathbf{E}_\sigma\left[\sup_{\mathbf{w}:\frac{1}{2}(\bar{D}\mathbf{w}^\top\mathbf{w}+D(\mathbf{w}^\top\mathbf{v}_i)^2)\leq E}\left|\frac{2}{n}\sum_{i=1}^{n}\sigma_i(\mathbf{w}^\top\mathbf{x}_i)\right|\right].$$

Consider the supremum inside the expectation. Depending on the sign of the term inside $|\cdot|$, the above corresponds to either a maximization or a minimization. Without loss of generality, consider the case of maximization. When a minimization is involved, the value of the objective still remains the same. The supremum is recovered by solving the following optimization problem:

$$\max_{\mathbf{w}}\ \mathbf{w}^\top\sum_{i=1}^{n}\sigma_i\mathbf{x}_i \quad \text{s.t.} \qquad \frac{1}{2}(\bar{D}\mathbf{w}^\top\mathbf{w} + D(\mathbf{w}^\top\mathbf{v}_i)^2) \leq E \qquad \forall 1\leq i\leq n_v. \tag{16}$$

Using Lagrange multipliers $\lambda_1 \geq 0, \ldots \lambda_{n_v} \geq 0$, the Lagrangian of (16) is: $\mathcal{L}(\mathbf{w},\lambda) = -\mathbf{w}^\top\sum_{i=1}^{n}\sigma_i\mathbf{x}_i + \sum_{i=1}^{n_v}\lambda_i\left(\frac{1}{2}\left(\bar{D}\mathbf{w}^\top\mathbf{w} + D(\mathbf{w}^\top\mathbf{v}_i)^2\right) - E\right)$. Differentiating this with respect

7. $T_2(\mathcal{A},\mathcal{B})$ has been defined on generic sets. When an already defined set, such as $\mathbf{V}$ (with a known number n_v of elements) is an argument to T_2, λ will be subscripted with i or j.

to the primal variable $\mathbf{w}$ and equating it to zero gives: $\mathbf{w} = \mathbf{\Sigma}_{\lambda,D}^{-1} \sum_{i=1}^{n} \sigma_i \mathbf{x}_i$, where $\mathbf{\Sigma}_{\lambda,D} := \bar{D} \sum_{i=1}^{n_v} \lambda_i \mathbf{I} + D \sum_{i=1}^{n_v} \lambda_i \mathbf{v}_i \mathbf{v}_i^\top$. Substituting this $\mathbf{w}$ in $\mathcal{L}(\mathbf{w}, \lambda)$ gives the dual of (16):

$$\min_{\lambda \geq 0} \ \frac{1}{2} \sum_{i=1}^{n} \sigma_i \mathbf{x}_i^\top \mathbf{\Sigma}_{\lambda,D}^{-1} \sum_{j=1}^{n} \sigma_j \mathbf{x}_j + E \sum_{i=1}^{n_v} \lambda_i.$$

This permits the following upper bound on the empirical Rademacher complexity since the primal and the dual objectives are equal at the optimum:

$$\begin{aligned}
\hat{R}(\mathcal{H}_{E,D}^{V}) &= \frac{2}{n} \mathbf{E}_\sigma \left[\min_{\lambda \geq 0} \ \frac{1}{2} \sum_{i=1}^{n} \sigma_i \mathbf{x}_i^\top \mathbf{\Sigma}_{\lambda,D}^{-1} \sum_{j=1}^{n} \sigma_j \mathbf{x}_j + E \sum_{i=1}^{n_v} \lambda_i \right] \\
&\leq \min_{\lambda \geq 0} \frac{2}{n} \mathbf{E}_\sigma \left[\frac{1}{2} \sum_{i=1}^{n} \sigma_i \mathbf{x}_i^\top \mathbf{\Sigma}_{\lambda,D}^{-1} \sum_{j=1}^{n} \sigma_j \mathbf{x}_j + E \sum_{i=1}^{n_v} \lambda_i \right] \\
&\leq \min_{\lambda \geq 0} \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i^\top \mathbf{\Sigma}_{\lambda,D}^{-1} \mathbf{x}_i + \frac{2}{n} E \sum_{i=1}^{n_v} \lambda_i = T_2(\mathbf{V}, \mathbf{S}).
\end{aligned}$$

On line one, the expectation is over the minimizers over λ; this is less than first taking the expectation and then minimizing over λ in line two. Then, simply recycle the arguments used in Theorem 9 to handle the expectation over σ. ■

Theorem 11 *$\hat{R}(\mathcal{G}_{B,D}^{U}) \leq T_1(\mathbf{U}, \mathbf{S})$, where, for any training set $\mathcal{B}$ and landmark set $\mathcal{A}$,* $T_1(\mathcal{A}, \mathcal{B}) := \frac{2\sqrt{2B}}{|\mathcal{B}|} \left(\sum_{\mathbf{x} \in \mathcal{B}} \mathbf{x}^\top \left(\bar{D}\mathbf{I} + \frac{D}{|\mathcal{A}|} \sum_{\mathbf{u} \in \mathcal{A}} \mathbf{u}\mathbf{u}^\top \right)^{-1} \mathbf{x} \right)^{\frac{1}{2}}$.

Proof The proof is similar to the one for Theorem 10. ■

Thus, the empirical Rademacher complexities of the function classes of interest are bounded using the functions T_0, $T_1(\mathbf{U}, \mathbf{S})$ and $T_2(\mathbf{V}, \mathbf{S})$. For both $\mathcal{F}_E$ and $\mathcal{G}_{E,D}^{U}$, the empirical Rademacher complexity is bounded by a closed-form expression. For $\mathcal{H}_{E,D}^{V}$, optimizing over the Lagrange multipliers (i.e. the λ's) can further reduce the upper bound on empirical Rademacher complexity. This can yield advantages over both $\mathcal{F}_E$ and $\mathcal{G}_{E,D}^{U}$ in many situations and the overall shape of $\mathbf{\Sigma}_{\lambda,D}$ plays a key role in determining the overall bound; this will be discussed in Section 4.7. Note that the upper bound $T_2(\mathbf{V}, \mathbf{S})$ is not a closed-form expression in general but can be evaluated in polynomial time using semi-definite programming by invoking Schur's complement lemma as shown by Boyd and Vandenberghe (2003).

4.4 From empirical to true Rademacher complexity

By definition 8, the empirical Rademacher complexity of a function class is dependent on the data sample, $\mathbf{S}$. In many cases, it is not possible to give exact expressions for the Rademacher complexity since the underlying distribution over the data is unknown.

However, it is possible to give probabilistic upper bounds on the Rademacher complexity. Since the Rademacher complexity is the expectation of its empirical estimate over the data, by a straightforward application of McDiarmid's inequality (Appendix A), it is possible to show the following:

Lemma 12 *Fix $\delta \in (0,1)$. With probability at least $1-\delta$ over draws of the samples $\mathbf{S}$ the following holds for any function class $\mathcal{F}$:*

$$R(\mathcal{F}) \leq \hat{R}(\mathcal{F}) + 2\sqrt{\frac{\ln(2/\delta)}{2n}} \tag{17}$$

and,

$$\hat{R}(\mathcal{F}) \leq R(\mathcal{F}) + 2\sqrt{\frac{\ln(2/\delta)}{2n}}. \tag{18}$$

At this point, the motivation for introducing the landmark sets $\mathbf{U}$ and $\mathbf{V}$ becomes clear. The inequalities (17) and (18) do not hold when the function class $\mathcal{F}$ is dependent on the set $\mathbf{S}$. Specifically, using the sample $\mathbf{S}$ instead of the landmarks breaks the required *iid* assumptions in the derivation of (17) and (18). Thus neither Lemma 12, nor any of the results in Section 4.5 are sound for the function classes $\mathcal{G}^S_{B,D}$ and $\mathcal{H}^S_{E,D}$.

4.5 Generalization bounds

This section presents generalization bounds for the three different function classes. The derivation largely follows the approach of Shawe-Taylor and Cristianini (2004) and, therefore, several details will be omitted in this article. Recall the theorem from Shawe-Taylor and Cristianini (2004) that leverages the empirical Rademacher complexity to provide a generalization bound.

Theorem 13 *Let $\mathcal{F}$ be a class of functions mapping Z to $[0,1]$; let $\{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ be drawn from the domain Z independently and identically distributed (*iid*) according to a probability distribution $\mathcal{D}$. Then, for any fixed $\delta \in (0,1)$, the following bound holds for any $f \in \mathcal{F}$ with probability at least $1-\delta$ over random draws of a set of samples of size n:*

$$\mathbf{E}_{\mathcal{D}}[f(\mathbf{z})] \leq \hat{\mathbf{E}}[f(\mathbf{z})] + \hat{R}(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2n}}. \tag{19}$$

Similarly, under the same conditions as above, with probability at least $1-\delta$,

$$\hat{\mathbf{E}}[f(\mathbf{z})] \leq \mathbf{E}_{\mathcal{D}}[f(\mathbf{z})] + \hat{R}(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2n}}. \tag{20}$$

Inequality (19) can be found in Shawe-Taylor and Cristianini (2004) and inequality (20) is obtained by a simple modification of the proof in Shawe-Taylor and Cristianini (2004). The following theorem, found in Shawe-Taylor and Cristianini (2004), gives a probabilistic upper bound on the future error rate based on the empirical error and the function class complexity.

Theorem 14 *Fix $\gamma > 0$. Let $\mathcal{F}$ be the class of functions from $\mathbb{R}^m \times \{\pm 1\} \to \mathbb{R}$ given by $f(\mathbf{x}, y) = -yg(\mathbf{x})$. Let $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be drawn* iid *from a probability distribution $\mathcal{D}$. Then, with probability at least $1 - \delta$ over the samples of size n, the following bound holds:*

$$\Pr_{\mathcal{D}}[y \neq \text{sign}(g(\mathbf{x}))] \leq \frac{1}{n\gamma} \sum_{i=1}^{n} \xi_i + \frac{2}{\gamma} \hat{R}(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2n}}, \tag{21}$$

where $\xi_i = \max(0, 1 - y_i g(\mathbf{x}_i))$ are the so-called slack variables.

The upper bounds that were derived in Section 4.2, namely: T_0, $T_1(\mathbf{U}, \mathbf{S})$ and $T_2(\mathbf{V}, \mathbf{S})$ can now be inserted into Equation 21 to give generalization bounds for each class of interest. However, a caveat remains since a separate set of landmark data was necessary to provide such generalization bounds. The next section provides steps to eliminate the landmark data set from the bound.

4.6 Stating bounds independently of landmarks

Note that the original function classes were defined using landmark examples. However, it is possible to eliminate these and state the generalization bounds independent of the landmark examples on function classes defined on the training data. Landmarks are eliminated from the generalization bounds in two steps. First, the empirical Rademacher complexities are shown to be concentrated and, second, the function classes defined using landmarks are shown to be supersets of the original function classes. One mild and standard assumption will be necessary, namely, that all samples from the distribution $\Pr([\mathbf{x}])$ have a norm bounded above by R with probability one.

4.6.1 Concentration of empirical Rademacher complexity

Recall the upper bound $T_1(\mathbf{U}, \mathbf{S})$ that was derived in Theorem 11. The following bounds show that these quantities are concentrated.

Theorem 15
i) With probability at least $1 - \delta$,

$$T_1(\mathbf{U}, \mathbf{S}) \leq \mathbf{E}_{\mathbf{U}}[T_1(\mathbf{U}, \mathbf{S})] + \mathcal{O}\left(\frac{1}{\sqrt{n}\sqrt{tr(\mathbf{K})}}\right).$$

ii) With probability at least $1 - \delta$,

$$T_2(\mathbf{V}, \mathbf{S}) \leq \mathbf{E}_{\mathbf{V}}[T_1(\mathbf{V}, \mathbf{S})] + \mathcal{O}\left(\frac{1}{\sqrt{n_v}\sqrt{tr(\mathbf{K})}}\right).$$

Proof McDiarmid's inequality from Appendix A can be applied to $T_1(\mathbf{U}, \mathbf{S})$ since it is possible to compute Lipschitz constants $c_1, c_2, \ldots, c_n$ that correspond to each input of the function. These Lipschitz constants all share the same value c which is derived in Appendix B. With this Lipschitz constant, McDiarmid's inequality (36) is directly applicable and

yields: $\Pr[T_1(\mathbf{U},\mathbf{S}) - \mathbf{E}_{\mathbf{U}}[T_1(\mathbf{U},\mathbf{S})] \geq \epsilon] \leq \exp\left(-2\epsilon^2/(nc^2)\right)$ Setting the upper bound on probability to δ, the following inequality holds with probability at least $1-\delta$:

$$T_1(\mathbf{U},\mathbf{S}) \leq \mathbf{E}_{\mathbf{U}}[T_1(\mathbf{U},\mathbf{S})] + \frac{2\sqrt{\ln(1/\delta)E}}{\bar{D}\sqrt{n}}\left(\sqrt{\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i} - \sqrt{\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i - \frac{DR^2\mu_{max}}{n\bar{D}+DR^2}}\right). \tag{22}$$

The second term above is:

$$\begin{aligned}
&\frac{2\sqrt{\ln(1/\delta)E}}{\bar{D}\sqrt{n}}\left(\sqrt{\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i} - \sqrt{\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i - \frac{DR^2\mu_{max}}{n\bar{D}+DR^2}}\right)\\
&= \frac{2\sqrt{\ln(1/\delta)E}}{\bar{D}\sqrt{n}}\frac{DR^2\mu_{max}/(n\bar{D}+DR^2)}{\sqrt{\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i} + \sqrt{\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i - \frac{DR^2\mu_{max}}{n\bar{D}+DR^2}}}\\
&\leq \frac{2\sqrt{\ln(1/\delta)E}}{\bar{D}\sqrt{n}}\frac{DR^2\mu_{max}/(n\bar{D}+DR^2)}{\sqrt{\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i}}\\
&\leq \frac{2\sqrt{\ln(1/\delta)E}}{\bar{D}\sqrt{n}}\frac{DR^4 n}{(n\bar{D}+DR^2)\sqrt{\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i}}\\
&\leq \frac{2\sqrt{\ln(1/\delta)E}}{\bar{D}\sqrt{n}}\frac{DR^4 n}{(n\bar{D})\sqrt{\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i}} = \mathcal{O}\left(\frac{1}{\sqrt{n}\sqrt{\mathrm{tr}(\mathbf{K})}}\right).
\end{aligned}$$

Here, $\mu_{max} \leq nR^2$ is the largest eigenvalue of the Gram matrix $\mathbf{K}$. The big oh notation refers to the asymptotic behavior in n. Note that $\mathrm{tr}(\mathbf{K})$ also grows with n. Thus, asymptotically, the above term is better than $\mathcal{O}(1/\sqrt{n})$ which is the behavior of (21). So, from (22), with probability at least $1-\delta$: $T_1(\mathbf{U},\mathbf{S}) \leq \mathbf{E}_{\mathbf{U}}[T_1(\mathbf{U},\mathbf{S})] + \mathcal{O}\left(1/\sqrt{n\ \mathrm{tr}(\mathbf{K})}\right)$.

The proof for the second claim is similar since $T_2(\mathbf{V},\mathbf{S})$ has the same Lipschitz constants (Appendix B). The only difference is in the number of elements in $\mathbf{V}$ which is reflected in the bound. ■

4.6.2 Function class inclusion

At this point, using Equation 21 and Theorem 15, it is possible to state bounds that hold for functions in $\mathcal{G}^U_{B,D}$ and $\mathcal{H}^U_{B,D}$ but that are independent of $\mathbf{U}$ and $\mathbf{V}$ otherwise. However, the aim is to state uniform convergence bounds for functions in $\mathcal{G}^S_{B,D}$ and $\mathcal{H}^S_{B,D}$. This is achieved by showing the latter two sets are subsets of the former two with high probability. It is not enough to show that each element of one set is inside the other. Since uniform bounds are required for the initial function classes, one has to prove set-inclusion results[8].

Theorem 16 *For $B = E + \epsilon$ where $\epsilon = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$, with probability at least $1-2\delta$ $\mathcal{G}^S_{E,D} \subseteq \mathcal{G}^U_{B,D}$.*

8. The function classes will also be treated as sets of parameters $\mathbf{w}$ without introducing additional notation.

Proof First, note that $\mathcal{G}^S_{E,D} \subseteq \mathcal{F}_{E/\bar{D}}$. Thus, $\mathcal{F}_{E/\bar{D}}$ is a bigger class of functions than $\mathcal{G}^S_{E,D}$. Moreover, $\mathcal{F}_{E/\bar{D}}$ is not dependent on data. Now, consider $\frac{\bar{D}}{2}\mathbf{w}^\top\mathbf{w} + \frac{D}{2}(\mathbf{w}^\top\mathbf{x})^2$ where $\mathbf{w} \in \mathcal{F}_{E/\bar{D}}$. For $||\mathbf{x}|| \leq R^2$, the Cauchy-Schwarz inequality yields $\sup_{\mathbf{w}\in\mathcal{F}_{E/\bar{D}}} \frac{\bar{D}}{2}\mathbf{w}^\top\mathbf{w} + \frac{D}{2}(\mathbf{w}^\top\mathbf{x})^2 \leq \kappa$ where $\kappa = E/2 + DER^2/(2\bar{D})$. Now, define the function $h^{\mathbf{w}} : \mathcal{R}^m \to [0,1]$, as : $h^{\mathbf{w}}(\mathbf{x}) = (\frac{\bar{D}}{2}\mathbf{w}^\top\mathbf{w} + \frac{D}{2}(\mathbf{w}^\top\mathbf{x})^2)/\kappa$. Since the sets $\mathbf{S}$ and $\mathbf{U}$ are drawn *iid* from the distribution $\mathcal{D}_x$, it is now possible to apply (19) and (20) for any $\mathbf{w} \in \mathcal{F}_{E/\bar{D}}$. Applying (20) to $h^{\mathbf{w}}(\cdot)$ on $\mathbf{S}$, $\forall \mathbf{w} \in \mathcal{F}_{E/\bar{D}}$, with probability at least $1-\delta$, the following inequality holds:

$$\mathbf{E}_{\mathcal{D}_x}[h^{\mathbf{w}}(\mathbf{x})] \leq \frac{1}{n}\sum_{i=1}^{n} h^{\mathbf{w}}(\mathbf{x}_i) + 2\sqrt{\frac{2E}{n\bar{D}}}\sqrt{\frac{1}{n}\mathrm{tr}(\mathbf{K})} + 3\sqrt{\frac{\ln(2/\delta)}{2n}}, \tag{23}$$

where the value of $\hat{R}(\mathcal{F}_{E/\bar{D}})$ has been obtained from Theorem 9. The expectation is over the draw of $\mathbf{S}$. Similarly, applying (19) to $h^{\mathbf{w}}(\cdot)$ on $\mathbf{U}$, with probability at least $1-\delta$, $\forall \mathbf{w} \in \mathcal{F}_{E/\bar{D}}$, the following inequality holds:

$$\frac{1}{n}\sum_{i=1}^{n} h^{\mathbf{w}}(\mathbf{u}_i) \leq \mathbf{E}_{\mathcal{D}_x}[h^{\mathbf{w}}(\mathbf{u})] + 2\sqrt{\frac{2E}{n\bar{D}}}\sqrt{\frac{1}{n}\mathrm{tr}(\mathbf{K}_u)} + 3\sqrt{\frac{\ln(2/\delta)}{2n}} \tag{24}$$

where $\mathbf{K}_u$ is the Gram matrix of the landmark examples in $\mathbf{U}$. Using the fact that expectations in (23) and (24) are the same, $\mathrm{tr}(\mathbf{K}_u) \leq nR^2$, and the union bound, the following inequality holds $\forall \mathbf{w} \in \mathcal{F}_{E/\bar{D}}$ with probability at least $1-2\delta$:

$$\frac{1}{n}\sum_{i=1}^{n} h^{\mathbf{w}}(\mathbf{u}_i) \leq \frac{1}{n}\sum_{i=1}^{n} h^{\mathbf{w}}(\mathbf{x}_i) + 4R\sqrt{\frac{2E}{n\bar{D}}} + 6\sqrt{\frac{\ln(2/\delta)}{2n}}.$$

Using the definition of $h^{\mathbf{w}}(\cdot)$, with probability at least $1-2\delta$, $\forall \mathbf{w} \in \mathcal{F}_{E/\bar{D}}$,

$$\frac{\bar{D}}{2}\mathbf{w}^\top\mathbf{w} + \frac{D}{2n}\sum_{i=1}^{n}(\mathbf{w}^\top\mathbf{u}_i)^2 \leq \frac{\bar{D}}{2}\mathbf{w}^\top\mathbf{w} + \frac{D}{2n}\sum_{i=1}^{n}(\mathbf{w}^\top\mathbf{x}_i)^2 + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right). \tag{25}$$

Now, suppose, $\frac{\bar{D}}{2}\mathbf{w}^\top\mathbf{w} + \frac{D}{2n}\sum_{i=1}^{n}(\mathbf{w}^\top\mathbf{x}_i)^2 \leq E$, which describes the function class $\mathcal{G}^S_{E,D}$. If B is chosen to be $E+\epsilon$ where $\epsilon = \mathcal{O}(\frac{1}{\sqrt{n}})$, then, $\forall \mathbf{w} \in \mathcal{F}_{E/\bar{D}}$, with probability at least $1-2\delta$, $\mathbf{w}^\top\mathbf{w} + \frac{D}{2n}\sum_{i=1}^{n}(\mathbf{w}^\top\mathbf{u}_i)^2 \leq B$. Since $\mathcal{F}_{E/\bar{D}}$ is a superset of $\mathcal{G}^S_{E,D}$, with probability at least $1-2\delta$, $\mathcal{G}^S_{E,D} \subseteq \mathcal{G}^U_{E,D}$. ■

Theorem 17 *For $n_v = \mathcal{O}(\sqrt{n})$, with probability at least $1-2\delta$, $\mathcal{H}^S_{E,D} \subseteq \mathcal{H}^V_{E,D}$.*

Proof First define the function, $g^{\mathbf{w}} : \mathbb{R}^m \to \mathbb{R}$, as $g^{\mathbf{w}}(\mathbf{v}) = \frac{\bar{D}}{2}\mathbf{w}^\top\mathbf{w} + \frac{D}{2}(\mathbf{w}^\top\mathbf{v})^2$. Define the indicator random variable $\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{v})>E]}$ which has a value 1 if $g^{\mathbf{w}}(\mathbf{v}) > E$ and a value 0 otherwise. By definition, $\forall \mathbf{w} \in \mathcal{H}^S_{E,D}$, $\forall \mathbf{x}_i \in \mathbf{S}$, $\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{x}_i)>E]} = 0$. Similarly, $\forall \mathbf{w} \in \mathcal{H}^V_{E,D}$, $\forall \mathbf{v}_i \in \mathbf{V}$, $\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{v}_i)>E]} = 0$. As before, consider a larger class of functions that is independent

of $\mathbf{S}$, namely, $\mathcal{F}_{E/\bar{D}}$. For an *iid* sample $\mathbf{S}$ from the distribution $\mathcal{D}_x$, applying (19) to the indicator random variables $\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{x})>E]}$ on the set $\mathbf{S}$, with probability at least $1-\delta$,

$$\mathbf{E}_{\mathcal{D}_x}[\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{x})>E]}] \leq \frac{1}{n}\sum_{i=1}^{n}\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{x}_i)>E]} + 2\sqrt{\frac{2E}{n\bar{D}}}\sqrt{\frac{1}{n}\mathrm{tr}(\mathbf{K})} + 3\sqrt{\frac{\ln(2/\delta)}{2n}}. \tag{26}$$

Similarly, applying (20) on the set $\mathbf{V}$, with probability at least $1-\delta$,

$$\frac{1}{n_v}\sum_{i=1}^{n_v}\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{v}_i)>E]} \leq \mathbf{E}_{\mathcal{D}_x}[\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{x})>E]}] + 2\sqrt{\frac{2E}{n\bar{D}}}\sqrt{\frac{1}{n_v}\mathrm{tr}(\mathbf{K}_v)} + 3\sqrt{\frac{\ln(2/\delta)}{2n_v}}. \tag{27}$$

Performing a union bound on (26) and (27), using the fact that $\mathrm{tr}(\mathbf{K}) \leq nR^2$ and $\mathrm{tr}(\mathbf{K}_v) \leq n_v R^2$ with probability at least $1-2\delta$, $\forall \mathbf{w} \in \mathcal{F}_{E/\bar{D}}$,

$$\frac{1}{n_v}\sum_{i=1}^{n_v}\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{v}_i)>E]} - \frac{1}{n}\sum_{i=1}^{n}\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{x}_i)>E]} \leq 4R\sqrt{\frac{2E}{n\bar{D}}} + 3\sqrt{\frac{\ln(2/\delta)}{2}}\left(\frac{1}{\sqrt{n}}+\frac{1}{\sqrt{n_v}}\right). \tag{28}$$

Equating the right hand side of the above inequality to $\frac{1}{n_v}$, the above inequality can be written more succinctly as:

$$\begin{aligned}&\mathbf{P}\left[\exists \mathbf{w} \in \mathcal{F}_{E/\bar{D}} \frac{1}{n_v}\sum_{i=1}^{n_v}\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{v}_i)>E]} - \frac{1}{n}\sum_{i=1}^{n}\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{x}_i)>E]}) \geq \frac{1}{n_v}\right] \\ \leq & 2\exp\left(-\frac{2}{9}\left(\frac{1}{n_v} - 4R\sqrt{\frac{2E}{n\bar{D}}}\right)^2 \Big/ \left(\frac{1}{\sqrt{n}}+\frac{1}{\sqrt{n_v}}\right)^2\right)\end{aligned}$$

The left hand side of the equation above is the probability that there exists a $\mathbf{w}$ such that the difference in the fraction of the number of examples that fall outside $\frac{\bar{D}}{2}\mathbf{w}^\top\mathbf{w} + \frac{D}{2}(\mathbf{w}^\top\mathbf{x})^2 \leq E$ over the random draw of the sets $\mathbf{S}$ and $\mathbf{V}$ is at least $\frac{1}{n_v}$. Thus, it gives an upper bound on the probability that $\mathcal{H}^S_{E,D}$ is contained in $\mathcal{H}^V_{E,D}$. This is because, if there is a $\mathbf{w} \in \mathcal{H}^S_{E,D}$ that is not in $\mathcal{H}^V_{E,D}$, for such a $\mathbf{w}$, $\frac{1}{n_v}\sum_{i=1}^{n_v}\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{v}_i)>E]} > \frac{1}{n_v}$ and $\frac{1}{n}\sum_{i=1}^{n}\mathbf{I}_{[g^{\mathbf{w}}(\mathbf{x}_i)>E]} = 0$. Thus, equating the right hand side of (28) to $\frac{1}{n_v}$ and solving for n_v, the result follows. Both an exact value and the asymptotic behavior of n_v are derived in Appendix C. ■

It is straightforward to write the generalization bounds of Section 4.5 only in terms of $\mathbf{S}$, completely eliminating the landmark set $\mathbf{U}$ from the results in this section. However, the resulting bounds now have additional factors which further loosen them. In spite of this, in principle, using a landmark set and compensating with McDiarmid's inequality can overcome the difficulties associated with a data-dependent hypothesis class and provide important generalization guarantees. In summary, the following overall bounds can now be provided for the function classes $\mathcal{F}_E$, $\mathcal{H}^S_{E,D}$ and $\mathcal{G}^S_{E,D}$. This result is obtained from a union bound of Theorem 14, Theorem 15, Theorem 16 and Theorem 17.

Theorem 18 *Fix $\gamma > 0$ and let $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be drawn* iid *from a probability distribution $\mathcal{D}$ where $\|x\| \leq R$.*

i) For any g from the function class $\mathcal{F}_E$, the following holds with probability at least $1-\delta$,

$$\Pr_{\mathcal{D}}[y \neq \operatorname{sign}(g(\mathbf{x}))] \leq \frac{1}{\gamma n}\sum_{i=1}^{n}\xi_i + 3\sqrt{\frac{\ln(2/\delta)}{2n}} + \frac{4\sqrt{2E}}{n\gamma}\sqrt{tr(\mathbf{K})}. \tag{29}$$

ii) For any g from the function class $\mathcal{H}^S_{E,D}$, the following inequality (a solution of a semi-definite program) holds for $n_v = \mathcal{O}(\sqrt{n})$ with probability at least $1-\delta$,

$$\begin{aligned}\Pr_{\mathcal{D}}[y \neq \operatorname{sign}(g(\mathbf{x}))] &\leq \frac{1}{n\gamma}\sum_{i=1}^{n}\xi_i + 3\sqrt{\frac{\ln(8/\delta)}{2n}} + \mathcal{O}\left(\frac{1}{\sqrt{n_v}\sqrt{tr(\mathbf{K})}}\right)\\ &+ \frac{2}{\gamma}\mathbf{E}_{\mathbf{V}}\left(\min_{\lambda\geq 0}\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^\top\left(\bar{D}\sum_{j=1}^{n_v}\lambda_j\mathbf{I} + D\sum_{j=1}^{n_v}\lambda_j\mathbf{v}_j\mathbf{v}_j^\top\right)^{-1}\mathbf{x}_i + \frac{2E}{n}\sum_{i=1}^{n_v}\lambda_i\right).\end{aligned} \tag{30}$$

iii) Similarly, for any g from the function class $\mathcal{G}^S_{E,D}$, the following bound holds for $B = E + \mathcal{O}(\frac{1}{\sqrt{n}})$ with probability at least $1-\delta$,

$$\begin{aligned}\Pr_{\mathcal{D}}[y \neq \operatorname{sign}(g(\mathbf{x}))] &\leq \frac{1}{n\gamma}\sum_{i=1}^{n}\xi_i + 3\sqrt{\frac{\ln(8/\delta)}{2n}} + \mathcal{O}\left(\frac{1}{\sqrt{n}\sqrt{tr(\mathbf{K})}}\right)\\ &+ \frac{4\sqrt{2B}}{n\gamma}\mathbf{E}_{\mathbf{U}}\left(\sum_{i=1}^{n}\mathbf{x}_i^\top\left(\bar{D}\mathbf{I} + \frac{D}{n}\sum_{j=1}^{n}\mathbf{u}_j\mathbf{u}_j^\top\right)^{-1}\mathbf{x}_i\right)^{\frac{1}{2}},\end{aligned} \tag{31}$$

where $\xi_i = \max(0, \gamma - y_i g(\mathbf{x}_i))$ are the so-called slack variables.

4.7 Discussion of the bounds

Clearly, all the three bounds in Theorem (18) have similar asymptotic behavior in n; so, how do they differ? We will consider the separable case such that the slack variables are zero and examine the data dependent terms in the three bounds above (which will be referred to as the SVM bound, RMM bound and $\mathbf{\Sigma}$-SVM bound respectively). For the SVM bound, the quantity of interest is $4\frac{\sqrt{2E}}{n\gamma}\sqrt{\operatorname{tr}(\mathbf{K})}$ and, for the $\mathbf{\Sigma}$-SVM bound, the quantity of interest is $\frac{4\sqrt{2E}}{n\hat{\gamma}}\sqrt{\left(\sum_{i=1}^{n}\mathbf{x}_i^\top\left(\bar{D}\mathbf{I} + \frac{D}{n}\sum_{j=1}^{n}\mathbf{u}_j\mathbf{u}_j^\top\right)^{-1}\mathbf{x}_i\right)}$. Similarly, for the RMM bound, the quantity of interest is:

$$\frac{2}{\hat{\gamma}}\left(\min_{\lambda\geq 0}\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^\top\left(\bar{D}\sum_{j=1}^{n_v}\lambda_j\mathbf{I} + D\sum_{j=1}^{n_v}\lambda_j\mathbf{v}_j\mathbf{v}_j^\top\right)^{-1}\mathbf{x}_i + \frac{2E}{n}\sum_{i=1}^{n_v}\lambda_i\right).$$

Here the expectations over $\mathbf{U}$ and $\mathbf{V}$ have been dropped for brevity; in fact, this is how these terms would have appeared without the concentration result (Theorem 15). Moreover, in the latter two cases, γ has been replaced by $\hat{\gamma}$ intentionally.

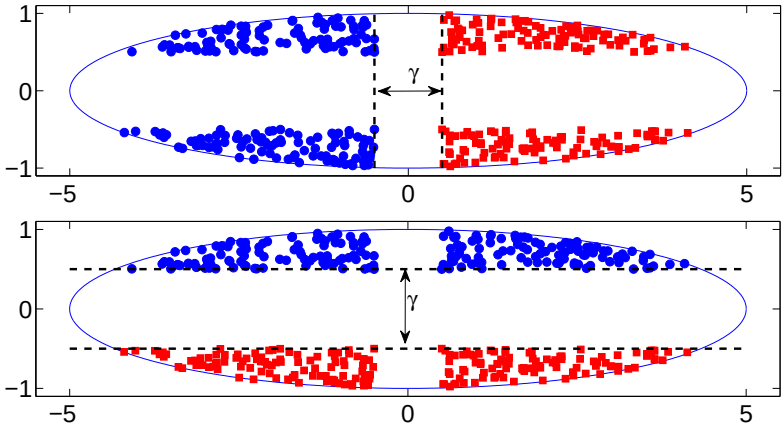

Figure 2: Two labelings of the same examples. Circles and squares denote the two classes (positive and negative). The top case is referred to as **toy example 1** and the bottom case is referred to as **toy example 2** in the sequel. The bound for the function class $\mathcal{F}_E$ does not distinguish between these two cases.

The differences between the three bounds will be illustrated with a toy example. In Figure 2, two different labelings of the same dataset are shown. The two different labelings of the data produce completely different classification boundaries. However, in both the cases, the absolute margin of separation γ remains the same. A similar synthetic setting was explored in the context of second order perceptron bounds (Cesa-Bianchi et al., 2005).

The margin γ corresponding to the function class $\mathcal{F}$ is found by solving the following optimization problem:

$$\max_{\gamma,\mathbf{w}} \gamma, \quad \text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i) \geq \gamma, \ \frac{1}{2}\mathbf{w}^\top \mathbf{w} \leq E.$$

This merely recovers the absolute margin γ which is shown in the figure. Similarly, for the function class $\mathcal{G}$, a margin $\hat{\gamma}$ is obtained by solving:

$$\max_{\gamma,\mathbf{w}} \ \gamma, \text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i) \geq \gamma, \ \frac{1}{2}\mathbf{w}^\top \left(\bar{D}\mathbf{I} + \frac{D}{n}\sum_{j=1}^{n} \mathbf{x}_j \mathbf{x}_j^\top \right) \mathbf{w} \leq E. \tag{32}$$

Through a change of variable, $\mathbf{u} = \mathbf{\Sigma}^{\frac{1}{2}}\mathbf{w}$ where $\mathbf{\Sigma} = \left(\bar{D}\mathbf{I} + \frac{D}{n}\sum_{j=1}^{n} \mathbf{x}_j \mathbf{x}_j^\top \right)$ it is easy to see that the above optimization problem is equivalent to

$$\max_{\gamma,\mathbf{u}} \ \gamma, \ \text{s.t. } y_i \mathbf{u}^\top \mathbf{\Sigma}^{-\frac{1}{2}} \mathbf{x}_i \geq \gamma, \ \frac{1}{2}\mathbf{u}^\top \mathbf{u} \leq E. \tag{33}$$

Thus, when a linear function is selected from the function class $\mathcal{G}^S_{D,E}$, the margin $\hat{\gamma}$ is estimated from a whitened version of the data. Similarly, for function class $\mathcal{H}^S_{E,D}$, the

	toy example 1	**toy example 2**
SVM bound	0.643	0.643
$\mathbf{\Sigma}$-SVM bound, D=0	0.643	0.643
$\mathbf{\Sigma}$-SVM bound, D=0.999	0.859	0.281
RMM bound, D=0	0.643	0.643
RMM bound, D=0.999	1.355	0.315

Table 1: The bound values for the two toy examples. The SVM bound does not distinguish between the two cases. By exploring D values, it is possible to obtain smaller bound values in both cases for $\mathbf{\Sigma}$-SVM and RMM ($D = 0$ in **toy example 1** and D close to one in **toy example 2**).

margin is estimated from a whitened version of the data where the whitening matrix is modified by Lagrange multipliers.

Thus, in the finite sample case, the bounds differ as demonstrated in the above synthetic problem. The bound for the function class $\mathcal{G}^S_{E,D}$ explores a whitening of the data. Suppose we fix $D = 0.999$, the result is a whitening which evens out the spread of the data in all directions. On this whitened data set, the margin $\hat{\gamma}$ appears much larger in **toy example 2** since it is large compared to the spread. This leads to an improvement in the $\mathbf{\Sigma}$-SVM bound over the usual SVM bound. While such differences could be compensated for by appropriate a priori normalization of features, this is not always an easy preprocessing.

Similarly, the RMM bound also considers a whitening of the data however, it shapes the whitening matrix adaptively by estimating λ. This gives further flexibility and rescales data not only along principal eigen-directions but in any direction where the margin is large relative to the spread of the data. By exploring D values, margin can be measured relative to the spread of the data rather than in the absolute sense. Since $\mathbf{\Sigma}$-SVM and RMM are strict generalizations of the SVM, through the use of a proper validation set, it is almost always possible to obtain improvements. The various bounds for the toy examples are shown in Table 1.

5. Experiments

In this section, a detailed investigation of the performance of the RMM[9] on several synthetic and real world datasets is provided.

5.1 Synthetic dataset

First consider a simple two dimensional dataset that illustrates the performance differences between the SVM and the RMM. Since this is a synthetic dataset, the best classifier can be constructed and Bayes optimal results can be reported. Consider sampling data from two different Gaussian distributions[10] corresponding to two different classes. Samples are

9. Code available at `http://www1.cs.columbia.edu/~pks2103/RMM`.
10. Due to such Gaussian assumptions, LDA or generative modeling would be appropriate contenders but are omitted to focus the discussion on margin-based approaches.

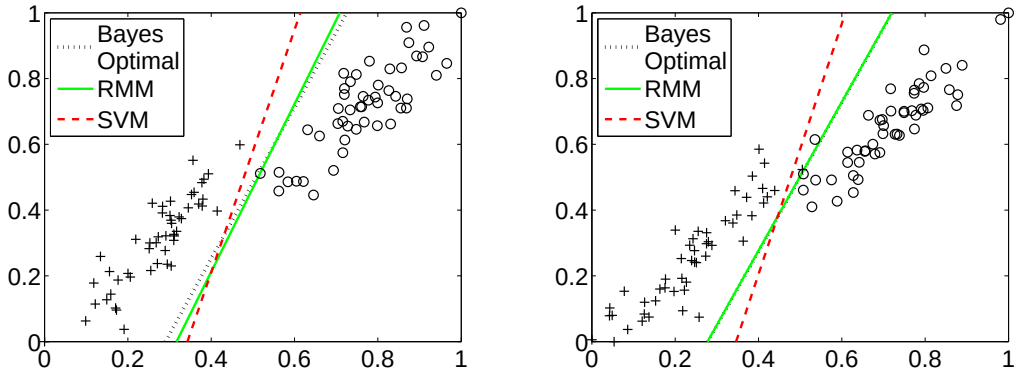

Figure 3: Two typical synthetic datasets (rescaled inside a 0-1 box) with corresponding SVM and RMM solutions are shown along with the Bayes optimal solution. The SVM (the RMM) solution uses the C (C and B) setting that minimized validation error. The RMM produces an estimate that is significantly closer to the Bayes optimal solution.

drawn from the two following Gaussian distributions with equal prior probability:

$$\boldsymbol{\mu}_+ = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{\mu}_- = \begin{bmatrix} 19 \\ 13 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} 17 & 15 \\ 15 & 17 \end{bmatrix}.$$

The Gaussians have different means yet identical covariance. A total of 100,000 samples were drawn from each of Gaussian to create validation and test sets. Large validation and test sets were used to get accurate estimates of validation and test error.

Due to the synthetic nature of the problem, the Bayes optimal classifier is easily recovered (Duda et al., 2000) and is given by the following decision boundary

$$(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} - 0.5(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_+ + \boldsymbol{\mu}_-) = 0. \tag{34}$$

The above formula uses the true means and covariances of the Gaussians (not empirical estimates). It is clear that the Bayes optimal solution is a linear decision boundary which is in the hypothesis class explored by both the RMM and the SVM. Note that the synthetic data was subsequently normalized to lie withing the zero-one box. This rescaling was taken into account while constructing the Bayes optimal classifier (34).

Various C values (and B values) were explored during SVM (RMM) training. The settings with minimum error rate on the validation set were used to compute test error rates. Furthermore, the test error rate for the Bayes optimal classifier was computed. Each experiment was repeated fifty times over random draws of train, test and validation sets. Figure 3 shows an example dataset from this synthetic experiment along with the (cross-validated) SVM, RMM and Bayes optimal classification boundaries. The SVM decision boundary is biased to separate the data in a direction where it has large spread. The RMM is less biased by the spread and is visibly closer to the Bayes optimal solution.

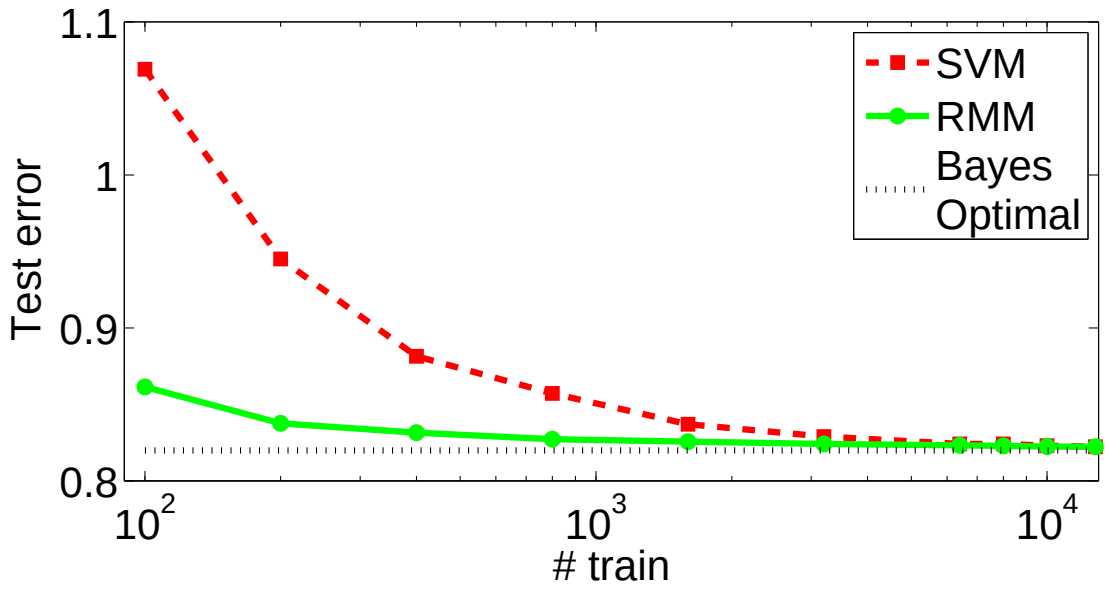

Figure 4: Percent test error rates for the SVM, RMM and Bayes optimal classifier as training data size is increased. The RMM has a statistically significant (at 5% level) advantage over the SVM until 6400 training examples. Subsequently, the advantage remains though with less statistical significance.

Figure 4 shows the test error rates achieved for the SVM, the RMM and the Bayes optimal classifier. The SVM performs significantly worse than the RMM, particularly when training data is scarce. The RMM maintains a statistically significant advantage over the SVM until the number of training examples grows beyond 6400. With larger training sample size n, regularization plays a smaller role in the future probability of error. This is clear, for instance, from the bound (29). The last term goes to zero at $\mathcal{O}(1/\sqrt{n})$, the second term (which is the outcome of regularization) is $\mathcal{O}(\sqrt{\mathrm{tr}(\mathbf{K})/n}\sqrt{1/n})$. Both have an $\mathcal{O}(1/\sqrt{n})$ rate. However, the first term in the bound is the average slack variables divided by the margin which does not go to zero asymptotically with increasing n and eventually dominate the bound. Thus, the SVM and RMM have asymptotically similar performance but have significant differences in the small sample case.

We next explored the effect data scaling in the synthetic experiment. To explore the effect of scaling in a controlled manner, we first examine the projection $\mathbf{w}$ recovered by the Bayes optimal classifier and construct a vector $\mathbf{v}$ orthogonal to $\mathbf{w}$ (i.e., $\mathbf{w}^\top\mathbf{v} = 0$). The samples (training, test and validation) were then projected onto the axes defined by $\mathbf{w}$ and $\mathbf{v}$. Each projection along $\mathbf{w}$ was preserved while the projection along $\mathbf{v}$ was scaled by a factor $s > 1$. This merely elongates the data further along directions orthogonal to $\mathbf{w}$ (i.e, along the Bayes optimal classification boundary). More concisely, given a sample $\mathbf{x}$, we

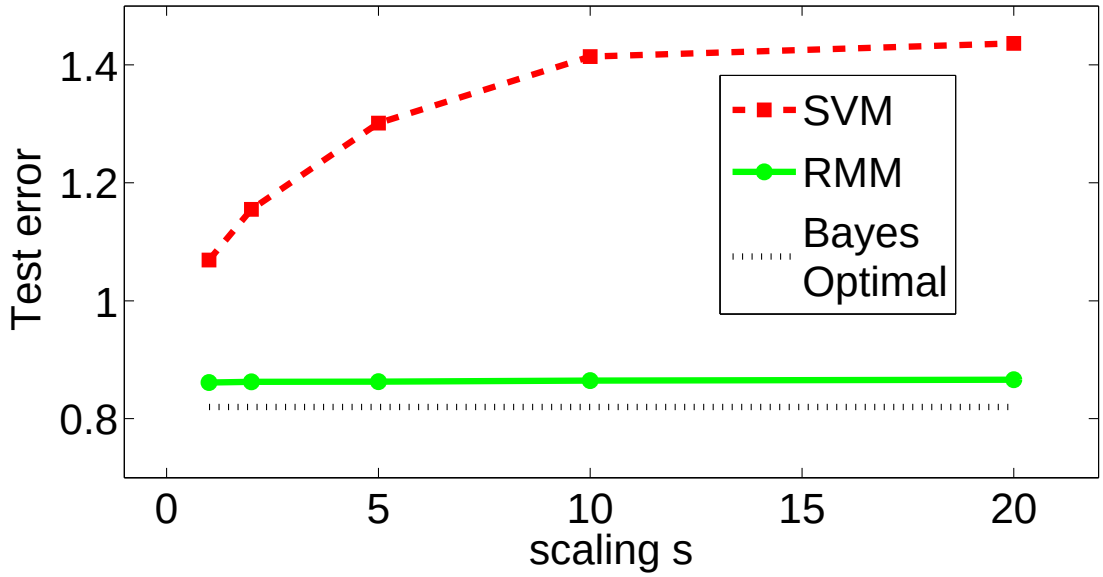

Figure 5: Percent test error rates for the SVM, RMM and Bayes optimal classifier as data is scaled according to (35). The RMM solution remains resilient to scaling while the SVM solution deteriorates significantly. The advantage of the RMM over the SVM is statistically significant (at the 1% level).

apply the following scaling transformation:

$$\begin{bmatrix} \mathbf{w} & \mathbf{v} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} \mathbf{w} & \mathbf{v} \end{bmatrix}^{-1} \mathbf{x}. \tag{35}$$

Figure 5 shows the SVM and RMM test error rate across a range of scaling values s. Here, 100 samples were used to construct the training data. Surprisingly, as s grows, the SVM further deviates from the Bayes optimal classifier and attempts to separate the data along directions of large spread. Meanwhile, the RMM remains resilient to scaling and maintains a low error rate throughout.

To explore the effect of the B parameter, the average validation and test error rate were computed across many settings of C and B. The setting $C = 100$ was chosen since it obtained the minimum error rate on the validation set. The average test error rate of the RMM is shown in Figure 6 at $C = 100$ for multiple settings of the B parameter. Starting from the SVM solution on the right (i.e. large B) the error rate begins to fall until it attains a minimum and then starts to go increase. A similar behavior is seen in many real world datasets. Surprisingly, some datasets even exhibit monotonic reduction in test error as the value of B is decreased. The following section investigates such real world experiments in more detail.

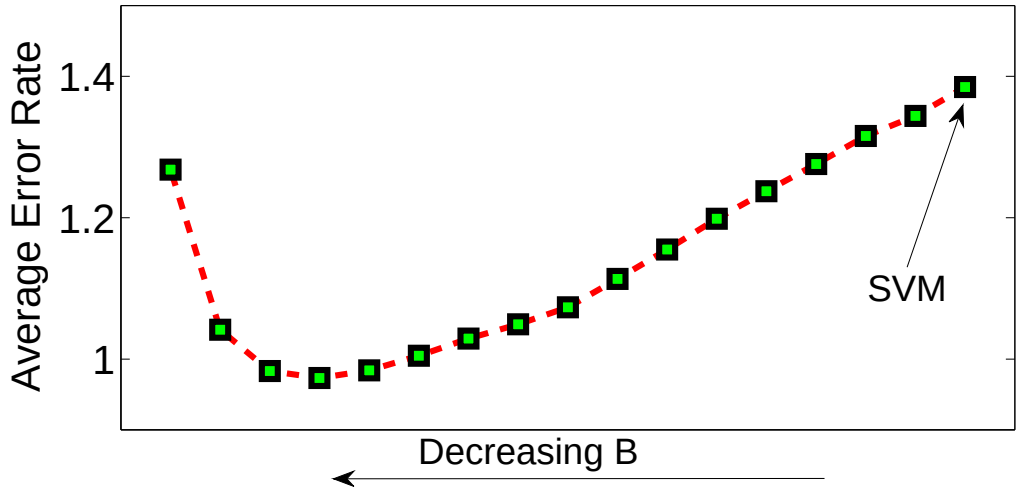

Figure 6: Behavior on the toy dataset with $C = 100$. As the B value is decreased, the error rate decreases to a reasonably wide minimum before starting to increase.

		1	2	3	4	5	6	7	RBF
OPT	SVM	71	57	54	47	40	46	46	51
	Σ-SVM	**61**	48	41	36	35	31	**29**	47
	KLDA	71	57	54	47	40	46	46	**45**
	RMM	71	**36**	**32**	**31**	**33**	**30**	**29**	51
USPS	SVM	145	109	109	103	100	95	93	104
	Σ-SVM	**132**	**108**	99	94	**89**	**87**	**90**	**97**
	KLDA	132	119	121	117	114	118	117	101
	RMM	153	109	**94**	**91**	91	90	**90**	98
1000-MNIST	SVM	696	511	422	380	362	338	332	670
	Σ-SVM	**671**	470	373	341	322	309	303	673
	KLDA	1663	848	591	481	430	419	405	1597
	RMM	689	**342**	**319**	**301**	**298**	**290**	**296**	**613**
2/3-MNIST	SVM	552	237	200	183	178	177	164	166
	RMM	**534**	**164**	**148**	**140**	**123**	**129**	**129**	**144**
Full MNIST	SVM	536	198	170	156	157	141	136	146
	RMM	**521**	**146**	**140**	**130**	**119**	**116**	**115**	**129**

Table 2: The number of misclassifications in three different digit datasets. Various kernels are explored using the SVM, **Σ**-SVM, KLDA and RMM methods.

5.2 Experiments on digits

Experiments were carried out on three datasets of digits - optical digits from the UCI machine learning repository (Asuncion and Newman, 2007), USPS digits (LeCun et al.,

1989) and MNIST digits (LeCun et al., 1998). These datasets vary considerably in terms of their number of features (64 in optical digits, 256 in USPS and 784 in MNIST) and their number of training examples (3823 in optical digits, 7291 in USPS and 60000 in MNIST). In all the multi-class experiments, the one versus one classification strategy was used. The one versus one strategy trains a classifier for every combination of two classes. The final prediction for an example is simply the class that is predicted most often. These results are directly comparable with various methods that have been applied on this dataset. For a fair comparison, results from contender methods that use special preprocessing or domain knowledge are not explored in this article. [11]

In all experiments, the digits were first normalized to have unit norm. This eliminates numerical problems that may arise in kernel functions such as the polynomial kernel $k(\mathbf{u}, \mathbf{v}) = (1 + \mathbf{u}^\top \mathbf{v})^d$. Classification results were then examined for various degrees of the polynomial kernel. In addition, kernel values were further normalized so that the trace of the training Gram matrix was equal to the number of training examples.

All parameters were tuned by splitting the training data according to an 80:20 ratio with the larger split being used for training and the smaller split for validation. The process was repeated five times over random splits to select hyper-parameters (C for the SVM, C and D for the $\mathbf{\Sigma}$-SVM and C and B for the RMM). A final classifier was trained for each of the 45 classification problems with the best parameters found by cross validation using all the training examples in its corresponding pair of classes.

For the MNIST digits experiment, the $\mathbf{\Sigma}$-SVM and kernel LDA (KLDA) methods were too computationally demanding due to their use of matrix inversion. To cater to these methods, a smaller experiment was conducted with 1000 examples per training. For the larger experiments, the $\mathbf{\Sigma}$-SVM and KLDA were excluded. The larger experiment on MNIST involved training on two thirds of the digits (i.e. training with an average of 8000 examples for each pair of digits) for each binary classification task. In both these experiments, the remaining training data was used as a validation set. The classifier that performed best on the validation set was used for testing.

After forming all 45 classifiers (corresponding to each pair of digits), testing was done on the standard separate test sets available for each of these three benchmark problems (1797 examples in the case of optical digits, 2007 examples in USPS and 10000 examples in MNIST). The final prediction for each test example was recovered based on the majority of predictions made by the 45 classifiers on the test example with ties broken uniformly at random.

It is important to note that, on the MNIST test set, an error rate improvement of 0.1% has been established as statistically significant (Bengio et al., 2007; Decoste and Schölkopf, 2002). This corresponds to 10 or more test examples being correctly classified by one method over an other.

Table 2 shows results on all three digits datasets for polynomial kernels under varying degrees as well as for RBF kernels. For each dataset, the number of misclassified examples using the majority voting scheme above is reported. The $\mathbf{\Sigma}$-SVM typically outperforms the SVM yet the RMM outperforms both. Interestingly, with higher degree kernels, the $\mathbf{\Sigma}$-SVM seems to match the performance of the RMM while in most lower-degree kernels,

11. Additional results are reported in `http://yann.lecun.com/exdb/mnist/`.

the RMM outperforms both the SVM and the $\mathbf{\Sigma}$-SVM convincingly. Since the $\mathbf{\Sigma}$-SVM is prohibitive to run on large scale datasets due to the computationally cumbersome matrix inversion, the RMM was clearly the most competitive method in these experiments in terms of both accuracy and computational efficiency.

The best parameters found by validation in the previous experiments were used in a full-scale MNIST experiment which does not have a validation set of its own. All 45 pairwise classifiers (both SVMs and RMMs) were trained with the previously cross-validated parameters using *all* the training examples for each class in MNIST for various kernels. The test results are reported in Table 2; the RMM advantages persist in this full-scale MNIST experiment.

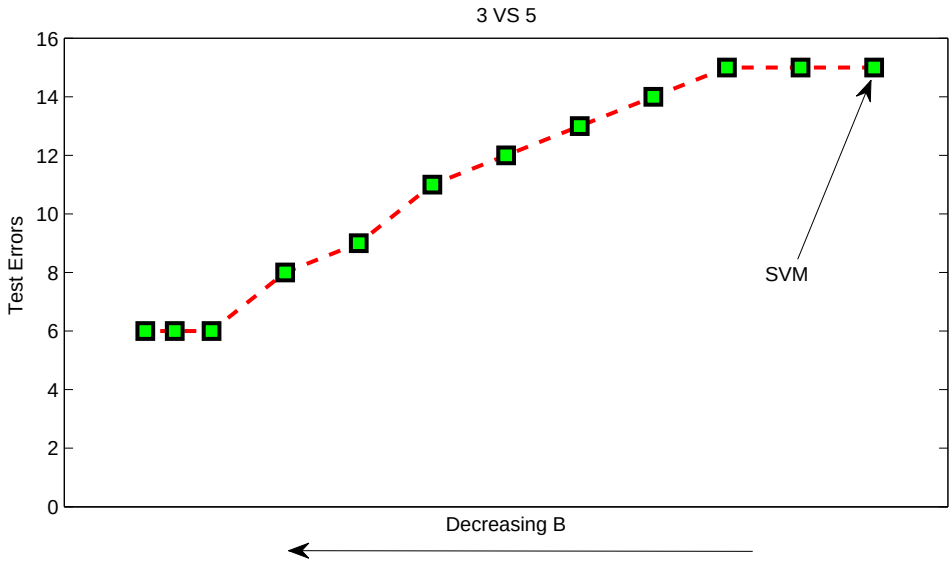

Figure 7: Performance on MNIST test set with digits 3 and 5. The number of errors decreases from 15 to 6 as B decreases from the right.

5.3 Classifying MNIST digits 3 vs 5

This section presents more detailed results on one particular binary classification problem in the MNIST digits dataset: the classification of digit 3 versus 5. Therein, the RMM has a dramatically stronger performance than the SVM. The results reported in this section are with polynomial kernels of degree 5. The parameter C was selected as mentioned above. With the selected C value, an SVM was first trained over the entire MNIST training set containing the digits 3 and 5. After noting the maximum absolute value of the output given on all the training examples, B value was reduced in steps. The number of test errors on the MNIST test set (3 versus 5) was noted. As the B value is reduced, the number of errors starts to diminish as shown in Figure 7. The number of errors produced by the SVM was 15. With the RMM, the number of errors dropped to 6 as the B value approached

one. Clearly, as B decreases, the absolute margin is decreased however the test error rate drops drastically. This empirically suggests that maximizing the relative margin can have a beneficial effect on the performance of a classifier. Admittedly, this is only one example and is provided only for illustrative purposes. However, similar behavior was observed in most of the binary digit classification problems though in some cases the error rate did not go down significantly with decreasing B values. The generalization behavior on all 45 individual problems is explored in more detail in Section 5.4.

5.4 All 45 binary MNIST problems

This section explores RMM performance on the 45 pairwise digit classification problems in isolation. In these experiments, both C and B values were fixed using validation as in previous sections. A total of 45 binary classifiers were constructed using all MNIST training digits. The resulting error rates are shown in Figure 8. On most problems, the RMM obtains a significantly lower error rate than the SVM and, at times, obtains half the error rate.

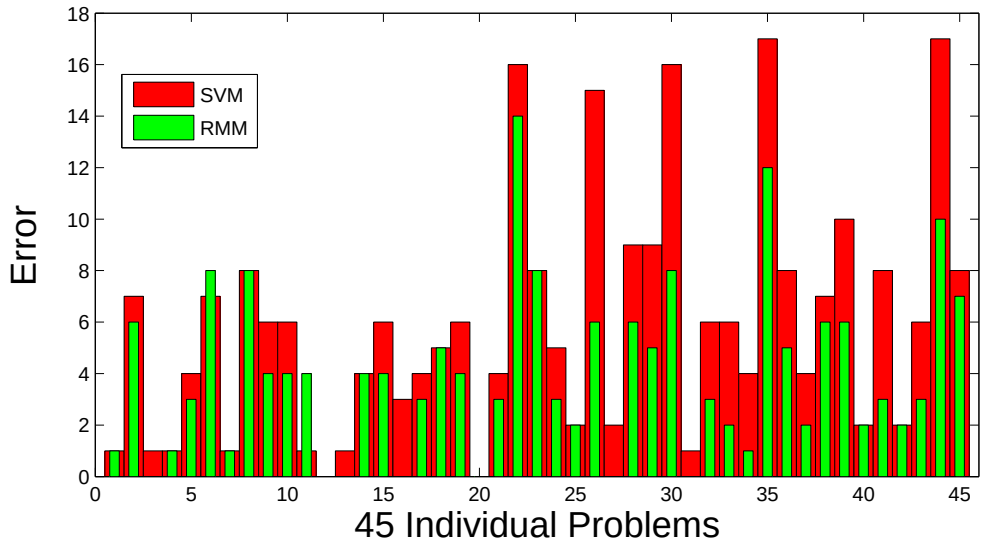

Figure 8: Total test errors on all 45 MNIST classification problems. Various classifiers were trained on the entire MNIST training dataset and evaluated on a standardized separate test set.

5.4.1 A comparison with the Universum method

A new framework known as the Universum (Weston et al., 2006; Sinz et al., 2008) was recently introduced which maximizes the margin while keeping classifier outputs low on an additional collection of non-examples that do not belong to either class of interest. These additional examples are known as Universum examples. Like landmarks, these are

Method	RMM	$\mathfrak{U}$-SVM		
# Universum	-	1000	3000	all
Error rate	1.081	1.059	1.037	1.020
Error Std Dev	0.138	0.142	0.149	0.159
p-value		0.402	0.148	0.031

Table 3: Percentage error rates for the RMM and the $\mathfrak{U}$-SVM. The rate for the SVM was 1.274 with a standard deviation of 0.179; this is significantly larger than all other results in the table (with a p-value of 0.000). The final row reports the p-value of a paired t-test between the RMM error rate and the $\mathfrak{U}$-SVM error rate (corresponding to the Universum size being considered in that column).

samples where a classifier's scalar predictions are forced to remain small. However, these Universum examples are obtained from any other distribution other than the one generating the training data. In the RMM, classification outputs on training examples are bounded; in the Universum, classification outputs on Universum examples are bounded (albeit with a different loss). The following experiments compare the Universum based framework with the RMM.

An MNIST experiment was explored for classifying digits 5 vs 8 using 1000 labeled training examples under the RBF kernel. This setup is identical to the experimental conditions described in Weston et al. (2006). Samples of the digit 3 served as Universum examples since these were reported to be the *best* performing Universum examples in previous work (Weston et al., 2006). The experiments used the standard implementation of the Universum provided by the authors Weston et al. (2006) under the default parameter settings (for variables such as ϵ). The Universum was compared with the RMM which had access to the same 1000 training examples. Furthermore, 3500 examples were used as a test set and another 3500 examples as a validation set to perform model selection. All parameter settings for the RMM and the Universum SVM (or $\mathfrak{U}$-SVM) as well as the variance parameter of the RBF kernel were explored over a wide range of values. The parameter settings that achieved the smallest error on a validation set were then used to evaluate performance on the test set (and vice-versa). This entire experiment was repeated ten-fold over different random draws of the various sets. The average test error rates were compiled for both algorithms.

While the RMM only had access to the 1000 training samples, the $\mathfrak{U}$-SVM was also given a Universum of images of the digit 3. The Universum spanned three different sizes - 1000, 3000 and 6131 samples (i.e. all available images of the digit 3 in the MNIST training set). The results are reported in Table 3. First, observe that both the RMM and the $\mathfrak{U}$-SVM improved the baseline SVM performance significantly (as measured by a paired t-test). With 1000 and 3000 Universum examples, even though the error rate of the $\mathfrak{U}$-SVM was slightly lower, a paired t-test revealed that it did not achieve statistically significant improvement over the RMM. Statistically significant advantages for the $\mathfrak{U}$-SVM only emerged when *all* the available images of the digit 3 were used in the Universum.

Note that there is a slight discrepancy between the errors reported here and those in Weston et al. (2006) even though both methods used the digit 3 to generate Universum examples. This may be because the previous authors Weston et al. (2006) reported the

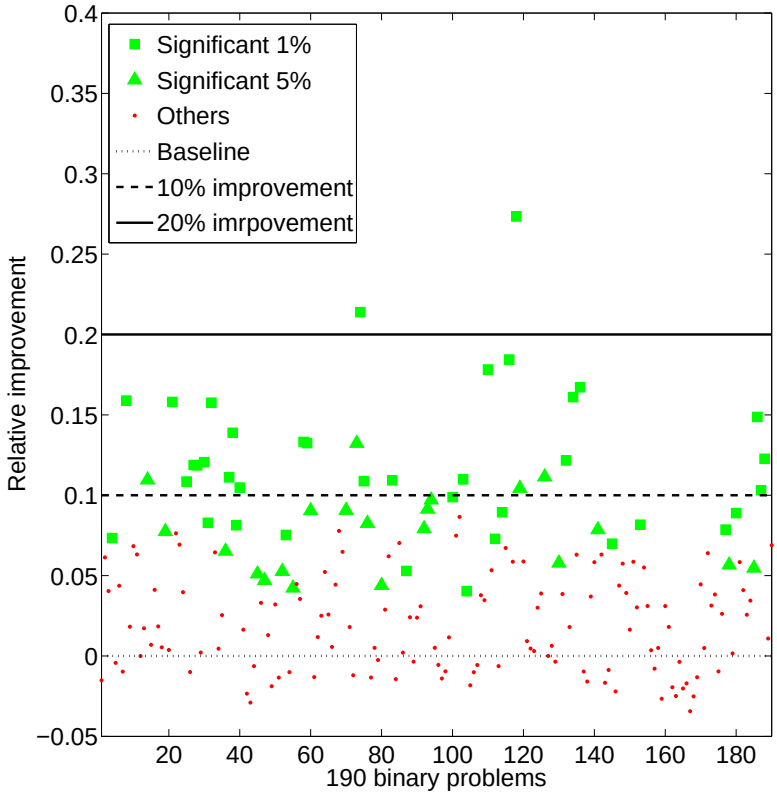

Figure 9: Percentage improvement of the RMM over the SVM on all 190 binary problems. Significance tests were performed using a paired t-test at the indicated levels of significance. On most problems, the RMM shows significant improvement over SVM.

best test error on 1865 examples. In this article, a more conservative approach is taken where a good model is first selected using the validation set and then errors are reported on an unseen test set without further tuning. Clearly, picking the minimum error rate on a test set will give more optimistic results but tuning to the test set can be potentially misleading. This makes it difficult to directly compare test error rates with those reported in the previous paper. While the error rate (using all digits 3 as the Universum examples) in our experiments varied from 0.74% to 1.35%, the authors in Weston et al. (2006) reported an error rate of 0.62%.

With 1000 training examples, the RMM (as in Equation (8)) has 1000 classification constraints and 1000 bounding constraints. With 1000 Universum examples, the $\mathfrak{U}$-SVM also has 1000 bounding constraints in addition to the classification constraints. It is interesting to note that the RMM, with no extra data, is not significantly worse than a $\mathfrak{U}$-SVM endowed with an additional 1000 or 3000 *best-possible* Universum examples.

The authors of Weston et al. (2006) observed that Universum examples help most when they are correlated with the training examples. This, coupled with the results in Table 3 and the fact that training examples are correlated most with themselves (or with examples from the same distribution as the training examples), raises the following question: How much of the performance gain with the $\mathfrak{U}$-SVM is due to the extra examples and how much of it is due to its implicit control of the spread (as with an RMM)? This is left as an open question in this article and as motivation for further theoretical work.

5.5 Text classification

In this section, results are reported on the 20 Newsgroups [12] dataset. This dataset has posts from 20 different Usenet newsgroups. Each post was represented by a vector which counts the number of words that occurred in the document. In the text classification literature, this is commonly known as the bag of words representation. Each feature vector was divided by the total number of words in the document to normalize it.

All 190 binary pairwise classification problems were considered in this experiment. For each problem, 500 examples were used for training. The remaining examples were divided into a validation and test set of the same size. Both SVMs and RMMs were trained for various values of their parameters. After finding the parameter settings that achieved the lowest error on a validation set, the test error was evaluated (and vice-versa). This experiment was repeated ten times for random draws of the train, validation and test sets.

Figure 9 summarizes the results. For each binary classification problem, a paired t-test was performed and p-values were obtained. As can be seen from the plot, the RMM outperforms the SVM significantly in almost 30% of the problems. This experiment once again demonstrates that an absolute margin does not always result in a small test error.

5.6 Benchmark datasets

To compare the performance of the RMM with a number of other methods, experiments were performed on several benchmark datasets. In particular, 100 training and test splits of 13 of these datasets have been previously used in (Raetsch et al., 2001; Mika et al., 1999; Cawley and Talbot, 2003) [13]. The RBF kernel was used in these experiments for all kernel-based methods. To handle the noisy nature of these datasets, the kernelized and relaxed version of the RMM (15) was used. All the parameters were tuned using cross-validation using a similar setup as in (Raetsch et al., 2001)[14]. With the chosen values of these parameters, the error rates were first obtained for all 100 test splits using the corresponding training splits. The results are reported in Table 4. Once again, the RMM exhibits clear performance advantages over other methods.

5.7 Scalability and run-time

While the asymptotic run time behavior was analyzed in Section 3.3, the run time of the RMM is also studied empirically in this section. In particular, the classification of MNIST

12. This dataset is available online at `http://people.csail.mit.edu/jrennie/20Newsgroups/`.
13. These datasets are available at `http://theoval.cmp.uea.ac.uk/~gcc/matlab/default.html#benchmarks`.
14. The values of the selected parameters and the code for the RMM are available for download at `http://www.cs.columbia.edu/~pks2103/ucirmm/`.

Dataset	SVM	KFDA	Σ-SVM	RMM (C=D)	RMM
banana	10.5 ± 0.4	10.8 ± 0.5	10.5 ± 0.4	**10.4 ± 0.4**	**10.4 ± 0.4***
b.cancer	**25.3 ± 4.6***	26.6 ± 4.8	28.8 ± 4.6	25.9 ± 4.5	**25.4 ± 4.6**
diabetes	**23.1 ± 1.7**	**23.2 ± 1.8**	24.2 ± 1.9	**23.1 ± 1.7**	**23.0 ± 1.7***
f.solar	**32.3 ± 1.8**	33.1 ± 1.6	34.6 ± 2.0	**32.3 ± 1.8***	**32.3 ± 1.8***
german	**23.4 ± 2.2**	24.1 ± 2.4	25.9 ± 2.4	**23.4 ± 2.1**	**23.2 ± 2.2***
heart	15.5 ± 3.3	15.7 ± 3.2	19.9 ± 3.6	15.4 ± 3.3	**15.2 ± 3.1***
image	**3.0 ± 0.6**	3.1 ± 0.6	3.3 ± 0.7	3.0 ± 0.6	**2.9 ± 0.7**
ringnorm	1.5 ± 0.1	**1.5 ± 0.1**	1.5 ± 0.1	**1.5 ± 0.1**	**1.5 ± 0.1***
splice	10.9 ± 0.7	10.6 ± 0.7	10.8 ± 0.6	10.8 ± 0.6	10.8 ± 0.6
thyroid	4.7 ± 2.1	**4.2 ± 2.1**	4.5 ± 2.1	**4.2 ± 1.8***	**4.2 ± 2.2**
titanic	22.3 ± 1.1	**22.0 ± 1.3***	23.1 ± 2.2	22.3 ± 1.1	22.3 ± 1.0
twonorm	**2.4 ± 0.1***	**2.4 ± 0.2**	2.5 ± 0.2	2.4 ± 0.1	**2.4 ± 0.1**
waveform	9.9 ± 0.4	9.9 ± 0.4	10.5 ± 0.5	10.0 ± 0.4	**9.7 ± 0.4***

Dataset	RBF	AB	LPAB	QPAB	ABR
banana	10.8 ± 0.4	12.3 ± 0.7	10.7 ± 0.4	10.9 ± 0.5	10.9 ± 0.4
b.cancer	27.6 ± 4.7	30.4 ± 4.7	26.8 ± 6.1	**25.9 ± 4.6**	26.5 ± 4.5
diabetes	24.3 ± 1.9	26.5 ± 2.3	24.1 ± 1.9	25.4 ± 2.2	23.8 ± 1.8
f.solar	34.4 ± 1.9	35.7 ± 1.8	34.7 ± 2.0	36.2 ± 1.8	34.2 ± 2.2
german	24.7 ± 2.4	27.5 ± 2.5	24.8 ± 2.2	25.3 ± 2.1	24.3 ± 2.1
heart	17.1 ± 3.3	20.3 ± 3.4	17.5 ± 3.5	17.2 ± 3.4	16.5 ± 3.5
image	3.3 ± 0.7	**2.7 ± 0.7**	2.8 ± 0.6	**2.7 ± 0.6***	**2.7 ± 0.6***
ringnorm	1.7 ± 0.2	1.9 ± 0.2	2.2 ± 0.5	1.9 ± 0.2	1.6 ± 0.1
splice	9.9 ± 0.8	10.1 ± 0.5	10.2 ± 1.6	10.1 ± 0.5	**9.5 ± 0.6***
thyroid	**4.5 ± 2.1**	**4.4 ± 2.2**	**4.6 ± 2.2**	**4.3 ± 2.2**	4.5 ± 2.2
titanic	23.3 ± 1.3	22.6 ± 1.2	24.0 ± 4.4	22.7 ± 1.0	22.6 ± 1.2
twonorm	2.8 ± 0.3	3.0 ± 0.3	3.2 ± 0.4	3.0 ± 0.3	2.7 ± 0.2
waveform	10.7 ± 1.1	10.8 ± 0.6	10.5 ± 1.0	10.1 ± 0.5	**9.8 ± 0.8**

Table 4: UCI results for a number of classification methods. Results are shown for the SVM, regularized kernel Fisher Discriminant Analysis, the Σ-SVM, the RMM, an RBF network, Adaboost, LP-regularized Adaboost, QP-regularized Adaboost and Regularized Adaboost. The results have been split into two parts due to lack of space. For each dataset, all methods could be placed on the same row in a larger table. For each dataset, the algorithm which gave the minimum error rate is starred. All other algorithms that were not significantly different from (at the 5% significance level based on a paired t-test) the minimum error rate are in boldface.

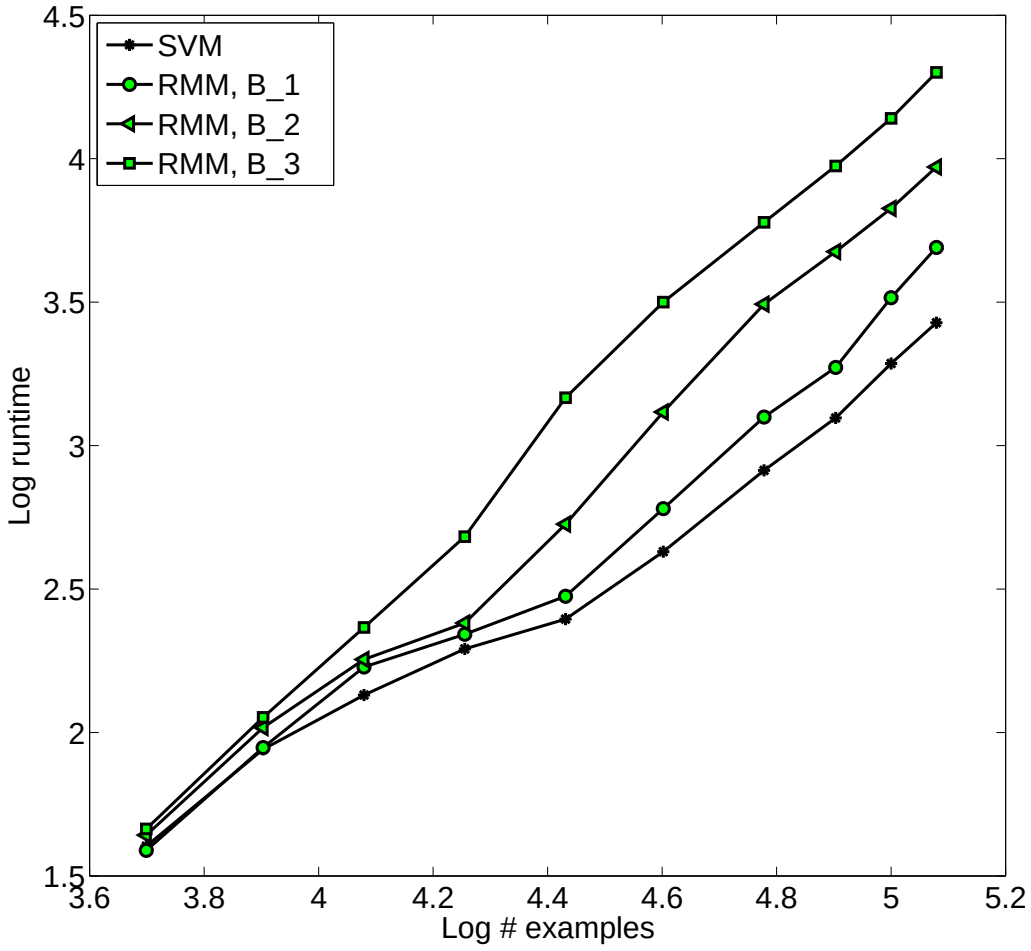

Figure 10: Log run time versus log number of examples. The figure shows that the SVM and the RMM have similar computational requirements overall.

digits 0-4 versus 5-9 with a polynomial kernel of degree five was used to benchmark the algorithms. For both the RMM and the SVM, the tolerance parameter (ϵ mentioned in Section 3.3) was set to 0.001. The size of the sub-problem (13) solved was 800 in all the cases. To evaluate how the algorithms scale, the number of training examples was increased in steps and the training time was noted. Throughout all the experiments, the C value was set to 1. The SVM was first run on the training examples. The value of maximum absolute prediction θ was noted. Three different values of B were then tried for the RMM: $B_1 = 1 + (\theta - 1)/2$, $B_2 = 1 + (\theta - 1)/4$ and $B_3 = 1 + (\theta - 1)/10$. In all experiments, the run time was noted. The experiment was repeated ten times to get an average run time for

each B value. A log-log plot comparing the number of examples to the average run time is shown in Figure 10. Both the SVM and the RMM run time exhibit similar asymptotic behavior.

6. Conclusions

The article showed that support vector machines and maximum margin classifiers can be sensitive to affine transformations of the input data and are biased in favor of separating data along directions with large spread. The relative margin machine was proposed to overcome such problems and optimizes the projection direction such that the margin is large only *relative to* the spread of the data. By deriving the dual with quadratic constraints, a geometric interpretation was also formulated for RMMs and led to risk bounds via Rademacher complexity arguments. In practice, the RMM implementation requires only additional linear constraints that complement the SVM quadratic program and maintain its efficient run time. Empirically, the RMM and maximum relative margin approach showed significant improvements in classification accuracy. In addition, an intermediate method known as $\mathbf{\Sigma}$-SVM was shown that lies between the SVM and the RMM both conceptually and in terms of classification performance.

Generalization bounds with Rademacher averages were derived. The SVM's bound which involves the trace of the kernel matrix was replaced with a more general whitened version of the trace of the kernel matrix. A proof technique using landmark examples led to Rademacher bounds on an empirical data-dependent hypothesis space. Furthermore, the bounds were stated independently of the particular sample of landmarks.

Directions of future work include exploring the connections between maximum relative margin and generalization bounds based on margin distributions (Schapire et al., 1998; Koltchinskii and Panchenko, 2002). By bounding outputs, the RMM is potentially finding a better margin distribution on the training examples. Previous arguments for such an approach were obtained in the context of voting methods (such as boosting) and may also be relevant here.

Furthermore, the maximization of relative margin is a fairly promising and general concept which may be compatible with other popular problems that have recently been tackled by the maximum margin paradigm. These include regression, ordinal regression, ranking and so forth. These are valuable and natural extensions for the RMM. Finally, since the constraints that bound the projections are unsupervised, RMMs can readily apply in semi-supervised and transductive settings. These are all promising directions for future work.

References

A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

M. Belkin, P. Niyogi, and V. Sindhwani. On manifold regularization. In *Artificial Intelligence and Statistics*, 2005.

Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 153–160. MIT Press, Cambridge, MA, 2007.

O. Bousquet, S. Boucheron, and G. Lugosi. *Introduction to Statistical Learning Theory*, volume Lecture Notes in Artificial Intelligence 3176, pages 169–207. Springer, Heidelberg, Germany, 2004.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2003.

G. C. Cawley and N. L. C Talbot. Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers. *Pattern Recognition*, 36:2585–2592, 2003.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. A second-order perceptron algorithm. *SIAM J. Comput.*, 34(3):640–668, 2005.

K. Crammer, M. Dredze, and F. Pereira. Exact convex confidence-weighted learning. In *Advances in Neural Information Processing Systems 21*, Cambridge, MA, 2009a. MIT Press.

K. Crammer, M. Mohri, and F. Pereira. Gaussian margin machines. In *Proceedings of the Artificial Intelligence and Statistics*, 2009b.

D. Decoste and B. Schölkopf. Training invariant support vector machines. *Machine Learning*, pages 161–190, 2002.

M. Dredze, K. Crammer, and F. Pereira. Confidence-weighted linear classification. In *Internationcal Conference on Machine Leaarning*, 2008.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

P. Haffner. Escaping the convex hull with extrapolated vector machines. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 753–760. MIT Press, Cambridge, MA, 2001.

R. Herbrich, T. Graepel, and C. Campbell. Bayes point machines. *Journal of Machine Learning Research*, 1:245–279, 2001.

T. Jaakkola, M. Meila, and T. Jebara. Maximum entropy discrimination. In *Advances in Neural Information Processing Systems 11*, 1999.

T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning*, 1999.

T. Joachims. Training linear SVMs in linear time. In *ACM SIGKDD International Conference On Knowledge Discovery and Data Mining (KDD)*, pages 217–226, 2006.

T. Joachims. Making large-scale support vector machine learning practical. In A. Smola B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

S.S. Keerthi. Efficient tuning of svm hyperparameters using radius/margin bound and iterative algorithms. *IEEE Transactions on Neural Networks*, 13:1225–1229, 2002.

V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30, 2002.

Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L. Jackel. Back-propagation applied to handwritten zip code recognition. *Neural Computation*, 1: 541–551, 1989.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

S. Mika, G. Raetsch, J. Weston, B. Scholkopf, and K.-R. Muller. Fisher discriminant analysis with kernels. In *in Neural Networks for Signal Processing IX*, pages 41–48, 1999.

G. Raetsch, T. Onoda, and K.-R. Muller. Soft margins for adaboost. *Machine Learning*, 43:287–320, 2001.

R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26:322–330, 1998.

B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, USA, 2002.

B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *International Conference on Machine Learning*, 2007.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

P. K. Shivaswamy and T. Jebara. Ellipsoidal kernel machines. In *Proceedings of the Artificial Intelligence and Statistics*, 2007.

P. K. Shivaswamy and T. Jebara. Relative margin machines. In *Advances in Neural Information Processing Systems 21*, Cambridge, MA, 2009a. MIT Press.

P. K. Shivaswamy and T. Jebara. Structured prediction with relative margin. In *International Conference on Machine Learning and Applications*, 2009b.

P. K. Shivaswamy, C. Bhattacharyya, and A. J. Smola. Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research*, 7:1283–1314, 2006.

F. Sinz, O. Chapelle, A. Agarwal, and B. Schölkopf. An analysis of inference with the universum. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1369–1376. MIT Press, Cambridge, MA, 2008.

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

J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 668–674. MIT Press, Cambridge, MA, 2000.

J. Weston, R. Collobert, F. H. Sinz, L. Bottou, and V. Vapnik. Inference with the universum. In *Proceedings of the International Conference on Machine Learning*, pages 1009–1016, 2006.

B. Zhang, X. Chen, S. Shan, and W. Gao. Nonlinear face recognition based on maximum average margin criterion. In *Computer Vision and Pattern Recognition*, pages 554–559, 2005.

Appendix A. McDiarmid's inequality

Assume $X_1, X_2, \ldots, X_n$ are independent random variables from a set $\mathcal{X}$ and $g : \mathcal{X}^n \to \mathbb{R}$. If the function g satisfies $\sup_{X_1,\ldots,X_n,\hat{X}_k} |g(X_1,\ldots,X_n) - g(X_1,\ldots,\hat{X}_k,\ldots,X_n)| \leq c_k$, for all $1 \leq k \leq n$ then, for any $\epsilon > 0$:

$$\Pr\left[g(X_1,\ldots,X_n) - \mathbf{E}[g(X_1,\ldots,X_n)] \geq \epsilon\right] \leq \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n c_i^2}\right), \tag{36}$$

$$\Pr\left[\mathbf{E}[g(X_1,\ldots,X_n)] - g(X_1,\ldots,X_n) \geq \epsilon\right] \leq \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n c_i^2}\right), \tag{37}$$

where the expectations are over the random draws of $X_1, \ldots, X_n$. Here the constants $c_1, c_2, \cdots, c_n$ are called Lipschitz constants.

Appendix B. Lipschitz constants for Section 4.6

Lemma 19 *The upper bound on $\hat{R}(\mathcal{G}^U_{B,D})$, namely $T_1(\mathbf{U},\mathbf{S})$, admits the Lipschitz constant:* $\frac{2\sqrt{2B}}{Dn}\left(\sqrt{\sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i} - \sqrt{\sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i - \frac{DR^2\mu_{max}}{n\bar{D}+DR^2}}\right)$.

Proof The quantity of interest is the worst change in

$$\frac{2\sqrt{2B}}{n}\sqrt{\sum_{i=1}^n \mathbf{x}_i(\bar{D}\mathbf{I} + \frac{D}{n}\sum_{j=1}^n \mathbf{u}_j\mathbf{u}_j^\top)^{-1}\mathbf{x}_i^\top}$$

when $\mathbf{u}_k$ is varied for any setting of $\mathbf{u}_1, \ldots, \mathbf{u}_{k-1}, \mathbf{u}_{k+1}, \ldots, \mathbf{u}_n$. Since $\sum_{j=1, j\neq k}^n \mathbf{u}_j\mathbf{u}_j^\top$ is positive semi-definite and inside the inverse operator, $\mathbf{u}_k$ will have the most extreme effect on the expression when $\sum_{j=1,j\neq k}^n \mathbf{u}_j\mathbf{u}_j^\top = \mathbf{0}$. Thus, consider:

$$\frac{2\sqrt{2B}}{n}\sqrt{\sum_{i=1}^n \mathbf{x}_i^\top\left(\bar{D}\mathbf{I} + \frac{D}{n}\mathbf{u}_k\mathbf{u}_k^\top\right)^{-1}\mathbf{x}_i}.$$

Apply the Woodbury matrix inversion identity to the term inside the square root:

$$\sum_{i=1}^{n} \mathbf{x}_i^\top \left(\bar{D}\mathbf{I} + \frac{D}{n}\mathbf{u}_k\mathbf{u}_k^\top\right)^{-1} \mathbf{x}_i = \frac{1}{\bar{D}} \sum_{i=1}^{n} \mathbf{x}_i^\top \left(\mathbf{I} - \frac{\mathbf{u}_k\mathbf{u}_k^\top}{\frac{n\bar{D}}{D} + \mathbf{u}_k^\top\mathbf{u}_k}\right)\mathbf{x}_i$$
$$= \frac{1}{\bar{D}} \left(\sum_{i=1}^{n} \mathbf{x}_i^\top\mathbf{x}_i - \frac{\sum_{i=1}^{n}(\mathbf{x}_i^\top\mathbf{u}_k)^2}{\frac{n\bar{D}}{D} + \mathbf{u}_k^\top\mathbf{u}_k}\right).$$

The maximum value of this expression occurs when $\mathbf{u}_k = \mathbf{0}$. To find the minimum, write the second term inside the brackets in the above expression as below:

$$\left(\frac{\mathbf{u}_k^\top}{\|\mathbf{u}_k\|} \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|}\right) / \left(\frac{n\bar{D}}{D\mathbf{u}_k^\top\mathbf{u}_k} + 1\right).$$

Clearly, in the numerator, the magnitude of $\mathbf{u}_k$ does not matter. To maximize this expression, $\mathbf{u}_k$ should be set to a vector of maximal length and in the same direction as the maximum eigenvector of $\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top$. Since all samples are assumed to have bounded norm no larger than R, the largest $\mathbf{u}_k$ vector has norm R. Denoting the maximum eigenvalue of $\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top$ by μ_{max}, it is easy to show the claimed value of Lipschitz constant for any k. ■

Lemma 20 *The upper bound on $\hat{R}(\mathcal{H}^V_{E,D})$, namely $T_2(\mathbf{V}, \mathbf{S})$, admits the Lipschitz constant:* $\frac{2\sqrt{2E}}{Dn}\left(\sqrt{\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i} - \sqrt{\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i - \frac{DR^2\mu_{max}}{n\bar{D}+DR^2}}\right)$.

Proof The quantity of interest is the maximum change in the following optimization problem over $\mathbf{u}_k$ for any setting of $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{k-1}, \mathbf{u}_{k+1}, \ldots, \mathbf{u}_{n_v}$:

$$\min_{\lambda \geq 0} \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i^\top \left(\bar{D}\sum_{j=1}^{n_v} \lambda_j\mathbf{I} + D\sum_{j=1}^{n_v} \lambda_j\mathbf{u}_j\mathbf{u}_j^\top\right)^{-1} \mathbf{x}_i + \frac{2}{n}E\sum_{i=1}^{n_v}\lambda_i.$$

As before, this happens when all $\mathbf{u}$'s except $\mathbf{u}_k$ are $\mathbf{0}$. In such a scenario, the expression is minimized for the setting $\lambda_j = 0$ for all $j \neq k$. The minimization only needs to consider variable settings of λ_k. Since this minimization is over a single scalar, it is possible to obtain a closed-form expression for λ_k. The optimal λ_k is merely: $\frac{1}{\sqrt{2E}}\sum_{i=1}^{n} \mathbf{x}_i^\top \left(\bar{D}\mathbf{I} + D\mathbf{u}_k\mathbf{u}_k^\top\right)^{-1}\mathbf{x}_i$. Substituting this into the objective gives an expression which is independent of λ's:

$$\frac{2\sqrt{2E}}{n}\sqrt{\sum_{i=1}^{n} \mathbf{x}_i^\top \left(\bar{D}\mathbf{I} + \frac{D}{n}\mathbf{u}_k\mathbf{u}_k^\top\right)^{-1}\mathbf{x}_i}.$$

This expression is identical to the one obtained in Theorem 19 and the proof follows. ■

Appendix C. Solving for n_v

Let $x = \frac{1}{\sqrt{n_v}}, c = 4R\sqrt{\frac{2E}{D}}$ and $b = \frac{3}{2}\sqrt{\frac{\ln(2/\delta)}{2}}$. Consider solving for x in the expression $x^2 - 2bx = (c+2b)/\sqrt{n}$. Equivalently, solve $(x-b)^2 = b^2 + (c+2b)/\sqrt{n}$. Taking the square root of both sides gives $x = b \pm \sqrt{b^2 + (c+2b)/\sqrt{n}}$. Since $x > 0$, only the positive root is considered. Thus, $\sqrt{n_v} = 1/(b + \sqrt{b^2 + (c+2b)/\sqrt{n}})$ which gives an exact expression for n_v. Dropping terms from the denominator produces the simpler expression: $\sqrt{n_v} \leq 1/\sqrt{(c+2b)/\sqrt{n}}$. Hence, $n_v \leq \frac{\sqrt{n}}{4R\sqrt{\frac{2E}{D}}+3\sqrt{\frac{\ln(2/\delta)}{2}}}$.