Mach Learn (2016) 103:215–237
DOI 10.1007/s10994-015-5542-8

Large margin classification with indefinite similarities

Ibrahim Alabdulmohsin[1] · Moustapha Cisse[1] ·
Xin Gao[1] · Xiangliang Zhang[1]

Received: 23 March 2015 / Accepted: 3 November 2015 / Published online: 7 January 2016

Abstract Classification with indefinite similarities has attracted attention in the machine learning community. This is partly due to the fact that many similarity functions that arise in practice are not symmetric positive semidefinite, i.e. the Mercer condition is not satisfied, or the Mercer condition is difficult to verify. Examples of such indefinite similarities in machine learning applications are ample including, for instance, the BLAST similarity score between protein sequences, human-judged similarities between concepts and words, and the tangent distance or the shape matching distance in computer vision. Nevertheless, previous works on classification with indefinite similarities are not fully satisfactory. They have either introduced sources of inconsistency in handling past and future examples using *kernel approximation*, settled for local-minimum solutions using *non-convex optimization*, or produced non-sparse solutions by *learning in Krein spaces*. Despite the large volume of research devoted to this subject lately, we demonstrate in this paper how an old idea, namely the 1-norm support vector machine (SVM) proposed more than 15 years ago, has several advantages over more recent work. In particular, the 1-norm SVM method is conceptually simpler, which makes it easier to implement and maintain. It is competitive, if not superior to, all other methods in terms of predictive accuracy. Moreover, it produces solutions that are often sparser than more recent methods by several orders of magnitude. In addition, we provide various theoretical justifications by relating 1-norm SVM to well-established learning algorithms such as neural

Editors: Hang Li, Dinh Phung, Tru Cao, Tu-Bao Ho, Zhi-Hua Zhou.

✉ Xiangliang Zhang
xiangliang.zhang@kaust.edu.sa

Ibrahim Alabdulmohsin
ibrahim.alabdulmohsin@kaust.edu.sa

Moustapha Cisse
mouhamadou.cisse@kaust.edu.sa

Xin Gao
xin.gao@kaust.edu.sa

[1] Computer, Electrical, and Mathematical Sciences and Engineering (CEMSE) Divison, King Abdullah University of Science and Technology (KAUST), Thuwal 23955, Saudi Arabia

networks, SVM, and nearest neighbor classifiers. Finally, we conduct a thorough experimental evaluation, which reveals that the evidence in favor of 1-norm SVM is statistically significant.

Keywords Support vector machine · Indefinite kernels · Similarity-based classification · Supervised learning · Linear programming

1 Introduction

Classification is one of the most fundamental problems in machine learning and statistics. In its binary form, it refers to the task of predicting a binary label $y \in \{-1, +1\}$ given an instance $x \in \mathcal{X}$, based on a training set of data whose instances and labels are both known. Classification has many important applications covering a wide spectrum that includes engineering, e-commerce, and medicine. For example, the popular UCI Machine Learning Repository (Lichman 2013) contains about 240 datasets, two thirds of which are for classification problems such as predicting disease severity, recognizing handwritten characters, identifying material types, detecting unsolicited messages, and recognizing signs. Due to its immense popularity, many algorithms have been invented for classification including nearest neighbor classifiers, decision trees, artificial neural networks, and Bayesian methods, to name only a few.

One of the most successful binary classification algorithms in practice today is the support vector machine (SVM) algorithm, which was developed in the 1990s by Vapnik and colleagues (Boser et al. 1992; Cortes and Vapnik 1995; Vapnik 1999). It works by constructing a separating hyperplane in a high-(possibly infinite-) dimensional feature space that separates the positive from the negative instances. Most importantly, it seeks a separating hyperplane, which ensures that most training instances are correctly classified with a *large margin*. The support vector machine (SVM) algorithm and its many variants were inspired by deep theoretical foundations that make use of the Vapnik-Chervonenkis (VC) dimension to establish the generalization ability of such family of classifiers (Burges 1998; Vapnik 1999). In informal terms, by seeking a large margin classifier, SVM tends to reduce its own risk of overfitting. We will return to such a statement later in Sect. 4.

However, one fundamental limiting factor in SVM is the need for positive semidefinite (PSD) similarities (a.k.a kernels). This follows from the fact that SVM is usually solved in its dual form:

$$\begin{aligned}
&\underset{\alpha}{\text{minimize}} && \frac{1}{2}\alpha^T\, Y\, K\, Y\, \alpha - \mathbf{1}^T\, \alpha \\
&\text{subject to} && 0 \leq \alpha \leq C\, \mathbf{1}, \quad y^T\, \alpha = 0
\end{aligned} \tag{1}$$

Here, $Y = \text{diag}(y)$, where $y \in \{-1,\ +1\}^m$ is a vector of m class labels, C is a fixed tradeoff constant, $\mathbf{1} = (1, 1, \ldots, 1)^T \in \mathbb{R}^m$, while $K \in \mathbb{R}^{m\times m}$ is the similarity (kernel) matrix. In the dual-form in (1), the similarity matrix K has to be symmetric positive semidefinite, i.e. satisfies the Mercer condition, in order to guarantee convexity of the optimization problem and the existence of a reproducing Hilbert kernel space (RHKS). When K is positive semidefinite, the optimization problem in (1) can be solved quite efficiently and its optimal solution can be used to construct a large margin separating hyperplane in some implicit feature space. Such advantages are no longer guaranteed when K is indefinite.

In real-life applications, however, many similarity functions exist that are either indefinite or for which the Mercer condition is difficult to verify. For example, one can incorporate the longest common subsequence in defining a distance between genetic sequences, use the

BLAST similarity score between protein sequences, use set operations such as the union and/or intersection in defining a similarity between transactions, use human-judged similarities between concepts and words, use the symmetrized Kullback–Leibler divergence between probability distributions, use dynamic time warping (DTW) for time series, or use the tangent distance and the shape matching distance in computer vision (Chen et al. 2009a; Wu et al. 2005; Ying et al. 2009; Haasdonk 2005). Indefinite similarities are also frequently encountered in psychology, neuroscience, and economics (Graepel et al. 1999). Extending large-margin classification to indefinite similarities will have many important applications.

Because classification with indefinite similarities is a frequently-encountered problem, many algorithms have been proposed in the literature to solve it. These include algorithms that are based on kernel approximation, non-convex optimization, and learning in Krein spaces. Other classical algorithms were also shown to be useful for the task as well such as nearest neighbor classifiers and relevance vector machine (RVM) (Graepel et al. 1999; Tipping 2001; Loosli et al. 2013). Despite the large volume of research devoted to this subject, however, we demonstrate in this paper how an old idea, namely the 1-norm support vector machine (SVM) method proposed more than 15 years ago (Graepel et al. 1999; Zhu et al. 2004), has several advantages over more recent work. In particular, the 1-norm SVM method is conceptually simpler, which makes it easier to implement and maintain. It is also competitive, if not superior to, all other methods in terms of predictive accuracy. Moreover, it produces solutions that are often sparser than more recent methods by several orders of magnitude.

There are several reasons why the 1-norm SVM method is competitive with more recent approaches in its predictive accuracy. Unlike many alternative methods that have been proposed in the literature, 1-norm SVM retains convexity of the optimization problem and treats both training and test examples consistently. It is closely connected to many well-established learning algorithms such as artificial neural networks, nearest neighbor classifiers, and SVM. As will be discussed in more details in the sequel, these connections between 1-norm SVM and those learning algorithms provide a formal justification to the use of 1-norm SVM when learning from indefinite similarities.

In the literature, 1-norm SVM is often used as an *embedded* feature selection method, where learning and feature selection are performed simultaneously (Bradley and Mangasarian 1998; Zhu et al. 2004; Fung and Mangasarian 2004; Zou 2007; Hilario and Kalousis 2008; Liu et al. 2010). It was studied in Zhu et al. (2004), where it was argued that 1-norm SVM has an advantage over the standard form of SVM in (1) when there are redundant noisy features. Despite the fact that it was suggested to be a viable method for classification with indefinite similarities more than 15 years ago (Graepel et al. 1999), it remained a relatively less known method than more-involved less-accurate approaches such as kernel approximation and non-convex optimization. In particular, 1-norm SVM is rarely used as a standard benchmark for the task (see for instance Chen et al. 2009b; Luss and d'Aspremont 2009; Ying et al. 2009; Chen and Ye 2008; Lin and Lin 2003).

The rest of the paper is structured as follows. First, we review the existing literature on learning with indefinite similarities. Second, we describe the 1-norm SVM method and show how it can be adapted to handle binary classification with indefinite similarities. We provide various motivations behind its formulation by relating 1-norm SVM to artificial neural networks, nearest neighbor classifiers, and SVM. We also show that 1-norm SVM can be interpreted as a method of minimizing an upper bound on the expected true risk (prediction error rate). After that, we present experimental results using both synthetic and real datasets, which validate the advantage of using 1-norm SVM in handling indefinite similarities over all other methods.

2 Previous work

Several methods have been proposed in the literature for learning with indefinite similarities. Some of these methods are old, such as non-convex optimization (Lin and Lin 2003), while others are more recent such as eigen-decomposition SVM (ESVM) that was proposed recently in 2013 (Loosli et al. 2013). Other more classical classification algorithms were also shown to be useful for the task as well, such as nearest neighbor classifiers and relevance vector machine (RVM) (Graepel et al. 1999; Tipping 2001; Loosli et al. 2013). Generally speaking, however, the most dominant methods can be grouped into three broad approaches: (1) kernel approximation, (2) non-convex optimization, and (3) learning in Krein spaces. We review each approach next.

2.1 Kernel approximation

The first approach for learning with indefinite similarities is the *kernel approximation* method. In this approach, not only does the learning algorithm look for a hypothesis that can correctly classify training instances with a large margin, but it also approximates the indefinite similarity matrix with a positive semidefinite (PSD) matrix so that the resulting optimization problem can be solved quite efficiently. That is, it is implicitly assumed that the advantage that would be reaped from convexifying a non-convex optimization problem outweigh the information loss we incur by artificially altering (distorting) the similarity matrix.

Two of the earliest kernel approximation methods are the *denoise* and the *flip* methods. Both methods alter the eigenvalues of the similarity matrix of training examples so that it becomes PSD. The two methods differ, however, in how they alter those eigenvalues. On one hand, the *denoise* method sets all negative eigenvalues to zero. The motivation behind such approach is to assume that negative eigenvalues are caused by noise (Pekalska et al. 2001). On the other hand, the *flip* method flips the sign of the negative eigenvalues, hence the name. This method aims at retaining some of the information coded in those negative eigenvalues (Pekalska et al. 2001; Graepel et al. 1999). A third more involved kernel approximation method is to formulate a max-min optimization problem that both seeks support vectors as well as a PSD kernel that approximates the indefinite similarity matrix. The latter approach was introduced by Luss and d'Aspremont in 2007 with improvements in training time reported in the following years (Chen and Ye 2008; Luss and d'Aspremont 2009; Chen et al. 2009b).

All of the kernel approximation methods above guarantee that the optimization problem remains convex during training. During prediction, however, the original indefinite similarity function is used. Hence, past and future examples are treated inconsistently. In addition, such methods are only useful when the similarity matrix is approximable by a PSD matrix. For other similarity functions, such as the sigmoid kernel that can occasionally yield a *negative semidefinite* matrix for certain values of its hyperparameters, the kernel approximation approach cannot be utilized. In fact, and as will be shown later in the evaluations in Sect. 5, the accuracy of some of these methods can become even *worse* than random guessing, especially when the similarity matrix is close to being negative semidefinite.

2.2 Non-convex optimization

The second approach for learning with indefinite similarities is *non-convex optimization*. In contrast to the previous kernel approximation approach, non-convex optimization implicitly assumes that treating training and test examples consistently by keeping the similarity function intact is more important than convexifying the problem. Of course, because the optimization problem is non-convex, however, this approach can terminate at a local minimum.

In the literature, non-convex optimization with indefinite similarities for SVMs has received a fair attention. In the theoretical side, Haasdonk interprets this approach as a method of minimizing the distance between reduced convex hulls in a pseudo-Euclidean space (Haasdonk 2005). In the practical side, SMO-type decomposition methods, which seek a stationary point, have been proposed for indefinite similarity functions such as the sigmoid kernel (Lin and Lin 2003). Nevertheless, because non-convex optimization can terminate at a stationary point that can be quite distant from the globally optimal solution, non-convex optimization does not guarantee learning (Chen et al. 2009a). In addition, this approach only works reasonably well if the similarity matrix is approximately positive semidefinite (PSD).

2.3 Learning in Krein spaces

The last major approach that has been proposed in the literature resolves many of the issues that are inherent in the kernel approximation and non-convex optimization methods. It proposes efficient learning algorithms that treat training and test examples consistently, and achieves a very high accuracy in practice. This fairly-recent approach is based on learning in Krein spaces, in which the similarity function is decomposed into the sum of one positive semidefinite kernel and one negative semidefinite kernel (Ong et al. 2004; Loosli et al. 2013). By taking such decomposition of similarity functions, learning in Krein spaces embraces the idea that the negative part of a similarity function contains viable information (Loosli et al. 2013).

Whereas learning in a Hilbert space can be formulated as a minimization problem, learning in a Krein space is formulated as a *stabilization* problem, where a saddle point to the objective function is found. One fairly recent algorithm that has been proposed to solve the stabilization problem is called eigen-decomposition SVM (ESVM) (Loosli et al. 2013). While this algorithm has been shown to outperform all previous methods, its primary drawback is that it does not produce sparse solutions, hence the entire list of training examples are often needed during prediction.

As will be shown later in the evaluations in Sect. 5, 1-norm SVM and ESVM both outperform all other methods significantly, and their performance is strikingly similar despite the different approaches employed by the two algorithms. Nevertheless, 1-norm SVM has the added advantage of producing a solution that is often sparser than ESVM by many orders of magnitude. Hence, 1-norm SVM will prove to be the most effective method for classification with indefinite similarities.

3 The 1-norm support vector machine

The 1-norm support vector machine (SVM) method was proposed more than 15 years ago (Graepel et al. 1999). It was rediscovered under many guises later, such as for being a special case of the *generalized SVM* (Mangasarian 1998) and for being a method of embedding similarities into features (Chen et al. 2009a; Pekalska et al. 2001). However, it is widely used in the literature for embedded feature selection (Bradley and Mangasarian 1998; Zhu et al. 2004; Fung and Mangasarian 2004; Zou 2007; Hilario and Kalousis 2008; Liu et al. 2010).

3.1 Description

Before we describe the 1-norm SVM method, we recall the binary classification setting. In this setting, we have an instance space $\mathcal{X}$ and a target set $\mathcal{Y} = \{+1, -1\}$. Every observation is a pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ drawn from some fixed unknown distribution $\mathcal{D}$. A classifier $h : \mathcal{X} \to \mathcal{Y}$ is a rule that maps each instance $x \in \mathcal{X}$ to either the positive class or the negative class. Throughout this paper, such a classifier is assumed to be of the form $h(x) = \mathbf{sign}(f(x))$ for some function $f : \mathcal{X} \to \mathbb{R}$, where f is learned on the basis of a set of m training examples $\{(x_i,\ y_i)\}_{i=1,\ldots,m}$ drawn i.i.d. from $\mathcal{D}$.

One natural method of predicting labels is *similarity-based* classification, which is a generalization to nearest neighbor classifiers. Given a similarity function $S : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, we can predict whether $y = +1$ or $y = -1$ for an instance x based on how similar x is to the fixed set of m training examples. A general approach of similarity-based classification is to use a decision rule of the form:

$$f(x) = \lambda_0 + \sum_{i=1}^{m} y_i\, \lambda_i\, S(x,\ x_i) \tag{2}$$

To reiterate, x is the instance whose label we would like to predict, whereas $\{(x_i,\ y_i)\}_{i=1,\ldots,m}$ is a training set of m observations drawn i.i.d. from some fixed unknown distribution $\mathcal{D}$.

To interpret the decision rule in (2), we note that λ_0 is a *biasing* term that is similar to the activation threshold in neural networks or the prior in Bayesian methods, while λ_i for $i \geq 1$ quantifies how important the training example $(x_i,\ y_i)$ is to the classification rule. According to (2), if an instance x is "more" similar to the weighted set of negative training examples, then $f(x)$ would be negative; otherwise, $f(x)$ is positive. Different methods of learning the weights λ yield different learning algorithms.

One of the most successful algorithms for similarity-based classification is the support vector machine (SVM). In SVM, the similarity function S is always chosen to be positive semidefinite, which implies that there exists a *feature mapping* $\phi : \mathcal{X} \to \mathbb{H}$ for some Hilbert space $\mathbb{H}$ endowed with an inner product $\langle\cdot,\ \cdot\rangle$ such that:

$$\forall x_i,\ x_j \in \mathcal{X}\ :\ S(x_i,\ x_j) = \langle\phi(x_i),\ \phi(x_j)\rangle$$

In the literature, $\mathbb{H}$ is often referred to as the *feature space* (Mohri et al. 2012). The existence of a feature space for a similarity function S is equivalent to the statement that S is positive semidefinite (Mohri et al. 2012). In such case, S is often referred to as a *kernel*.

If the similarity function S is positive semidefinite (PSD), then similarity in the instance space $\mathcal{X}$ can be interpreted differently in the feature space $\mathbb{H}$. Specifically, instead of performing a similarity-based classification in the instance space $\mathcal{X}$, one can seek a separating hyperplane with a large functional margin in the feature space $\mathbb{H}$. This is precisely the approach employed by SVM. The *Representer Theorem* states that such approach yields a decision rule that is identical to the similarity-based classification rule in (2) (Schölkopf and Smola 2002; Mohri et al. 2012). Hence, SVM is, indeed, a similarity-based classification algorithm.

Support vector machine (SVM) has been quite successful in practice. However, SVM can only be utilized if the similarity function is PSD as mentioned earlier. When the similarity function is indefinite, a different learning algorithm is required such as the 1-norm SVM. To see how 1-norm SVM can be adapted to handle indefinite similarities, we begin with the formulation proposed in Zhu et al. (2004). In this formulation, we have a dictionary of basis functions $\mathbb{D} = \{h_1(\cdot),\ h_2(\cdot),\ \ldots\}$, where $h_j : \mathcal{X} \to \mathbb{R}$, and consider classification using:

$$f(x) = \lambda_0 + \sum_{j\geq 1} \lambda_j \cdot h_j(x) \tag{3}$$

In the above expression for f, the basis functions $h_j : \mathcal{X} \to \mathbb{R}$ are fixed and the only variables to be optimized are λ_j for $j \geq 0$.

Zhu et al. (2004) proposed the following optimization problem for finding the weights λ (Eq 5 in Zhu et al. (2004)):[1]

$$\begin{aligned} &\underset{\lambda,\ \xi}{\text{minimize}} \ \ || \lambda ||_1 + C\ || \xi ||_1 \\ &\text{subject to } y_i \cdot \left(\lambda_0 + \textstyle\sum_j \lambda_j \cdot h_j(x_i)\right) \geq 1 - \xi_i \\ &\qquad\qquad\ \ \xi_i \ \geq 0, \qquad \text{for all } i = 1,\ 2, \ldots, m \end{aligned}$$

Here, C is a tradeoff parameter between regularization and fitting. We will provide several motivations behind such formulation shortly.

To utilize the above method in handling indefinite similarities, we set $h_j(\cdot) = y_j\ S(x_j,\ \cdot)$, where $S(x_j,\ \cdot) : \mathcal{X} \to \mathbb{R}$ is a function that measures similarity to the instance x_j. In addition, we impose the non-negativity constraint $\lambda_j \geq 0$ for all $j \geq 1$ to ensure that any instance x_j can be representative to its own class y_j only. This gives us the following linear program (LP):

$$\begin{aligned} &\underset{\lambda,\ \xi}{\text{minimize}} \ \ \textstyle\sum_{i=0}^m \lambda_i + C\ \ \sum_{i=1}^m \xi_i \\ &\text{subject to } \left[y,\ \ Q\right] \lambda \geq 1 - \xi \\ &\qquad\qquad\ \ \lambda_i,\ \ \xi_i \ \geq 0, \qquad \text{for all } i = 1,\ 2, \ldots, m \end{aligned} \tag{4}$$

Here, $y \in \{-1,\ +1\}^m$ is a vector of class labels for all m training examples and $Q \in \mathbb{R}^{m\times m}$ is given by:[2]

$$Q_{i,j} = y_i\ y_j\ S(x_i,\ x_j)$$

The above formulation is a simple LP that can be solved quite efficiently using, for example, the Gurobi solver (Gurobi Optimization 2012). Note that unlike the standard formulation of SVM in (1), the LP formulation in (4) remains convex even when the matrix Q is not PSD because both the objective function and inequality constraints are linear in the optimization variables $(\lambda,\ \xi)$. Once the LP is solved, we predict the label of a new instance x using the earlier classification rule in (2).

Training examples x_i with $\lambda_i > 0$ are analogous to the *support vectors* in SVM, and we will refer to them as support vectors here as well. As depicted in Fig. 1, each support vector is 'carefully' placed in the plane to guard a region dominated by its respective class. In practice, because the regularization term in the objective function minimizes the ℓ_1 norm of λ, the vector λ tends to be sparse and the number of support vectors tends to be small.

3.2 Computational complexity

Before we begin to analyze 1-norm SVM, we make a final remark on its computational complexity. The 1-norm SVM method is a linear program (LP), for which many efficient solvers currently exist. These include, for instance, the Gurobi solver (Gurobi Optimization 2012),

[1] In Zhu et al. (2004), Eq. 5, the hinge loss is used explicitly in the objective function, which is equivalent to the use of slack variables in our formulation.

[2] To reiterate, the similarity function $S : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is determined by the application at hand, and not by the learning system. Therefore, we assume a similarity function is given, and do not address whether or not it is suitable for the learning task.

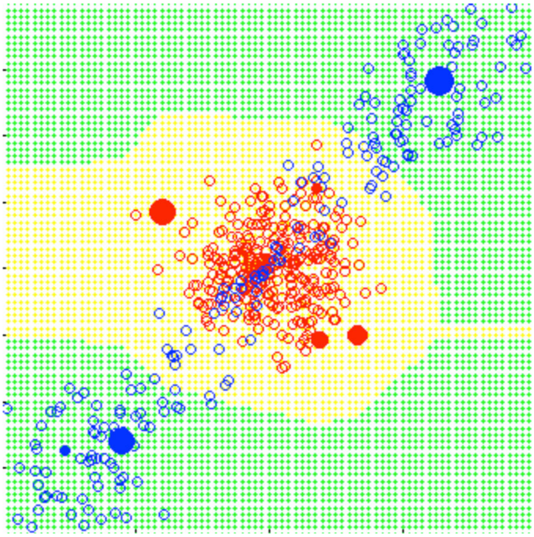

Fig. 1 In this figure, two classes are shown in *red* and *blue*. A *solid* marker is a support vector, whose size is proportional to its weight λ_i. Classification regions for *red* and *blue* classes are shown in *yellow* and *green* respectively (Color figure online)

the CPLEX solver (IBM 2015), and MATLAB's built-in `linprog` command. When using *interior-point* methods, rigorous bounds have been established for the number of computations required to solve a linear program (Boyd and Vandenberghe 2004). In general, it can be shown that the 1-norm SVM method requires, at most, $O(m^3)$ computations in the worst case.

4 Analysis

In this section, we provide several motivations for using the linear program (LP) in (4) to classify with indefinite similarities. First, we show that 1-norm SVM can be interpreted as a method of producing a decision boundary with a large similarity margin in the instance space $\mathcal{X}$. Using theoretical bounds expressed in terms of the margin, large-margin classification is proven to be less susceptible to over-fitting. On a related note, we show that 1-norm SVM is a large-margin classifier because it can also be interpreted as an ℓ_1-regularized linear SVM applied to the empirical kernel maps.

Next, we relate 1-norm SVM to nearest neighbor classifiers, which reveals that 1-NN classification is a special case of the 1-norm SVM. Nearest neighbor classifiers are some of the most important classification algorithms in practice today with provable performance bounds. For example, it has long been established that the true risk (test error rate) of 1-NN is asymptotically bounded from above by twice the Bayes risk (Cover and Hart 1967).

After that, we show how the 1-norm SVM can be interpreted as an approximate implementation of the structural risk minimization (SRM) induction principle (Vapnik 1999). This can be deduced by establishing the connection between 1-norm SVM and two-layer neural networks, which, in turn, justifies the use of ℓ_1 regularization. Similarly, we show that the objective function in (4) can be interpreted as a method of minimizing an upper bound on expected prediction error rate using the leave-one-out (LOO) error estimation method.

4.1 Large-margin classification

We begin by interpreting 1-norm SVM as a large-margin classifier. Given an instance space $\mathcal{X}$, a target set $\mathcal{Y} = \{+1, -1\}$, and a suitable measure of similarity $S : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, we can define similarity between an instance $x \in \mathcal{X}$ and a label $y \in \mathcal{Y}$ using a weighted sum of similarities with all of its training instances that belong to the label y. In other words, we may write:

$$\mathbb{S}(x,\ y;\ \lambda) = \sum_{i=1}^{m} \lambda_i\, S(x,\ x_i) \cdot \mathbb{I}\{y_i = y\}$$

to denote class similarity between x and a class $y \in \{+1, -1\}$. Here, the weight $\lambda_i \geq 0$ represents the *importance* of the training instance x_i to its own class y_i. In addition, we can introduce an offset λ_0 that quantifies prior preference. This offset plays a role that is similar to the *prior* in Bayesian methods, the activation threshold in neural networks, and the offset in SVM. Thus, we consider classification using the rule:

$$\hat{y} = \mathbf{sign}\{\mathbb{S}(x,\ +1;\ \lambda) - \mathbb{S}(x,\ -1;\ \lambda) + \lambda_0\}, \tag{5}$$

which is identical to the classification rule of 1-norm SVM given in (2). Moreover, we define the *similarity margin* M_i for the training example $(x_i,\ y_i)$ in the usual sense:

$$M_i = \mathbb{S}(x_i,\ y_i;\ \lambda) - \mathbb{S}(x_i,\ -y_i;\ \lambda) + y_i\, \lambda_0$$

This notion of similarity margin reduces to the notion of functional margin when the similarity function S is positive semidefinite (PSD). In general, the ith training example $(x_i,\ y_i)$ is classified correctly if and only if its margin is positive (i.e. $M_i > 0$).

Maximizing the minimum similarity margin can be formulated as a linear program (LP). First, we write:

$$\begin{aligned} &\underset{\lambda,\ M}{\text{maximize}}\ M \\ &\text{subject to}\ \mathbb{S}(x_i,\ y_i;\ \lambda) - \mathbb{S}(x_i,\ -y_i;\ \lambda) + y_i\, \lambda_0 \geq M, \quad (\text{for all } 1 \leq i \leq m) \\ &\qquad\qquad\ \lambda \geq 0 \end{aligned}$$

However, the decision rule given by (5) does not change when we multiply the weights λ by any fixed positive constant including constants that are arbitrarily large. This is because the decision rule only looks into the sign of its argument. In particular, we can always rescale the weights λ to be arbitrarily large, for which $M \to \infty$. This degree of freedom implies that we need to maximize the ratio $M/||\lambda||$ instead of maximizing M in absolute terms. Here, any norm $||\cdot||$ suffices but the 1-norm is preferred because it produces sparse solutions and because it gives a better accuracy in practice.[3]

Since our objective is to maximize the ratio $M/||\lambda||_1$, we can fix $M = 1$ and minimize $||\lambda||_1$. This results in the following optimization problem:[4]

$$\begin{aligned} &\underset{\lambda}{\text{minimize}}\ ||\lambda||_1 \\ &\text{subject to}\ \mathbb{S}(x_i,\ y_i;\ \lambda) - \mathbb{S}(x_i,\ -y_i;\ \lambda) + y_i\, \lambda_0 \geq 1, \quad \text{for all } i = 1, 2, \ldots, m \\ &\qquad\qquad\ \lambda_i \geq 0, \qquad \text{for all } i = 1, 2, \ldots, m \end{aligned}$$

[3] Sparse solutions are important for at least two reasons. First, only a small subset of the training set is needed during prediction, and hence prediction can be carried out quite efficiently. Second, minimizing the number of support vectors can be interpreted as a method of minimizing an upper bound on the expected test error rate [see for example Eq. (93) in Burges (1998) in the case of SVM and the discussion in Sect. 4.5 in the case of 1-norm SVM].

[4] An alternative derivation is as follows. We can introduce the coefficient vector $w = \lambda/M$ and fix $||\lambda|| = 1$ to avoid the issue of rescaling. Then, maximizing the margin M becomes equivalent to minimizing an appropriate norm of w such as the ℓ_1 norm. This leads to the same linear program that is used in 1-norm SVM.

In addition, to avoid over-fitting outliers or noisy observations and to be able to handle the case of non-separable classes, slack variables ξ can be introduced as well. This results in the LP formulation of the 1-norm SVM given earlier in (4). Hence, 1-norm SVM can be interpreted as a method of finding a decision boundary with a large similarity margin in the instance space $\mathcal{X}$. Such interpretation holds regardless of whether or not the similarity function is PSD. Thus, we expect 1-norm SVM to perform well even for indefinite similarity functions.

There are various results established in the literature for large-margin classification, which reveal that maximizing the margin can help mitigate the risk of over-fitting. If we define $R_{\mathcal{D}}(h)$ to be the true risk of an inferred hypothesis h, i.e. $R_{\mathcal{D}}(h) = \mathbb{P}_{(x,y)\sim\mathcal{D}}[y\, h(x) < 0]$, then several bounds of the following form can be established (Schapire et al. 1998; Mohri et al. 2012):

$$R_{\mathcal{D}}(h) \le \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}\{y_i\, h(x_i) < M\} + O\Big(\frac{C(\mathcal{H})}{M\sqrt{m}}\Big) \tag{6}$$

Here, the first term measures the margin-based error, which is the fraction of training examples whose margin is below a fixed number M, while the second term is a generalization risk that depends on the margin M, the number of training examples m, and on some appropriate measure of complexity of the hypothesis class $C(\mathcal{H})$. Most importantly, such results hold uniformly for all $M \in (0, 1)$. Because maximizing the margin on the training set for a fixed m and $\mathcal{H}$ reduces *both* terms in the right-hand side simultaneously, large-margin classification, such as by using the 1-norm SVM, tend to perform well in practice.

4.2 Empirical Kernel maps

In addition to being interpreted as a method of producing a decision boundary with a large similarity margin, the 1-norm SVM can also be interpreted as an ℓ_1-regularized SVM applied to the *empirical kernel map*. Given a training set $\{(x_i,\, y_i)\}_{i=1,\ldots,m}$, one can introduce the new mapping:

$$\phi(\cdot) = (y_1\, S(x_1,\,\cdot),\ \ldots,\ y_m\, S(x_m,\,\cdot))\ :\ \mathcal{X} \to \mathbb{R}^m,$$

which is similar to the *empirical kernel map* (Schölkopf and Smola 2002) except for the presence of class labels. Essentially, such mapping turns similarities into features (Chen et al. 2009a; Pekalska et al. 2001).

Given the mapping $\phi\ :\ \mathcal{X} \to \mathbb{R}^m$, we can apply any classification algorithm on the new features $\phi(x)$. In particular, we can use linear SVM, which yields the following optimization problem:

$$\begin{array}{lll} \underset{w\in\mathbb{R}^m,\ \xi\in\mathbb{R}^m,\ b\in\mathbb{R}}{\text{minimize}} & ||\, w\,|| + C\ \sum_{i=1}^m\ \xi_i & \\ \text{subject to} & y_i\ \big(\phi(x_i)^T\ w + b\big) \ge 1 - \xi_i, & \text{for all } i = 1, 2, \ldots, m \\ & \xi_i \ge 0, \quad \text{for all } i = 1, 2, \ldots, m & \end{array} \tag{7}$$

Knowing that the decision boundary is given by:

$$y = \mathbf{sign}(\phi(x)^T\, w + b) = \mathbf{sign}\Big(b + \sum_{i=1}^{m} w_i\, y_i\ S(x, x_i)\Big),$$

we note that b corresponds in our earlier notation to the bias term λ_0 while $w_i = \lambda_i$ for all $i \ge 1$.

With ℓ_1 regularization, the optimization problem in (7) becomes nearly identical to that of 1-norm SVM in (4). The only (minor) difference is the non-negativity constraint $\lambda_i \geq 0$ for $i \geq 1$, which we imposed in 1-norm SVM so that it would also behave like a nearest neighbor classification algorithm as will be discussed later. Consequently, 1-norm SVM can be interpreted as a method of producing a separating hyperplane with a large functional margin applied to the empirical kernel map ϕ. Again, although it is possible to use any norm in the regularization term of the objective function in (7), such as l_2 regularization, ℓ_1 regularization is preferred because it produces a sparse solution so that only a small subset of the training set is needed during prediction.

Applying linear classification to the empirical kernel map can be justified rigorously. In Balcan et al. (2008), it is shown that linear classification via the empirical kernel map, which is identical to our interpretation of the 1-norm SVM method discussed above, is guaranteed to have a small true risk (prediction error rate) as long as the similarity function is reasonable. Here, "reasonable" means that a high similarity exists between objects of the same class and a low similarity exists between objects of different classes. This guarantee on performance holds regardless of whether or not the similarity function is PSD (Balcan et al. 2008). Therefore, we expect 1-norm SVM to perform quite well for indefinite similarities.

4.3 Nearest neighbor classification

Nearest neighbor classifiers are some of the most important data mining algorithms in practice today (Wu 2008), and theoretical results for such algorithms have long been established. For instance, one well-known result shows that the true risk of 1-NN is asymptotically bounded from above by twice the Bayes risk (Cover and Hart 1967). Here, we show that 1-NN classification is a special case of the 1-norm SVM method.

Lemma 1 *Given a semi-metric* $d : \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty)$, *let* $S(x_i, x_j) = \psi(\gamma \cdot d(x_i, x_j)) : \mathcal{X} \times \mathcal{X} \rightarrow (0, 1]$ *for some bandwidth* $\gamma > 0$ *be a radial monotone decreasing function of distance that satisfies* $\psi(0) = 1$ *and the property:* $z_1 > z_2 \Rightarrow \lim_{\gamma \to \infty} \frac{\psi(\gamma z_1)}{\psi(\gamma z_2)} = 0$. *Also, let* $C = 1 + \epsilon$ *for some* $\epsilon > 0$ *and fix* $\lambda_0 = 0$ *(i.e. with no bias term). Then, the behavior of 1-norm SVM can be made arbitrarily close to* 1*-NN using a sufficiently large bandwidth* $\gamma \rightarrow \infty$.

Proof First, because d is a semi-metric on the instance space $\mathcal{X}$, we have $d(x_i, x_i) = 0$ and $S(x_i, x_i) = 1$. By setting $\lambda_0 = 0$, the margin M_i for the training example (x_i, y_i) reduces to:

$$M_i = y_i \sum_j \lambda_j\, y_j\, S(x_i, x_j) = \lambda_i + y_i \sum_{i \neq j} \lambda_j\, y_j\, S(x_i, x_j)$$

Since $S(x_i, x_j) > 0$ and $y_j \in \{+1, -1\}$, we deduce the sandwich inequality:

$$-\sum_{j \neq i} \lambda_j\ S(x_i, x_j)\ \leq\ M_i - \lambda_i\ \leq \sum_{j \neq i} \lambda_j\ S(x_i, x_j) \tag{8}$$

However, if $x_i \neq x_j$, then $S(x_i, x_j) \rightarrow 0$ at the limit $\gamma \rightarrow \infty$ by assumption. Because the size of the training set m is assumed to be finite, $M_i \rightarrow \lambda_i$ as $\gamma \rightarrow \infty$. Thus, the 1-norm SVM optimization problem can be made arbitrarily close to the following LP:

$$\begin{aligned}
&\underset{\lambda,\,\xi}{\text{minimize}} \quad \sum_{i=1}^{m} \lambda_i + (1+\epsilon) \sum_{i=1}^{m} \xi_i \\
&\text{subject to} \\
&\quad \lambda_i + \xi_i \geq 1 \quad (\text{for all } i) \\
&\quad \lambda,\ \xi\ \geq 0
\end{aligned}$$

Because $\epsilon > 0$, the optimal solution is given by $\lambda_i = 1$ for all i.

Next, suppose we have a new observation x, and let x_j be its nearest neighbor in the training set with respect to the semi-metric d. Then:

$$\begin{aligned}
\lim_{\gamma\to\infty} \frac{\sum_{i=1}^{m} S(x, x_i)}{S(x, x_j)} &= 1 + \lim_{\gamma\to\infty} \sum_{i\neq j} \frac{S(x, x_i)}{S(x, x_j)} \\
&= 1 + \sum_{i\neq j} \lim_{\gamma\to\infty} \frac{S(x, x_i)}{S(x, x_j)} = 1
\end{aligned}$$

Here, we interchanged the limit and the summation because the sum is finite. The above equation shows that as $\gamma \to \infty$, class similarity is dominated by the nearest neighbor. Hence, 1-norm SVM using the prediction rule in Eq (2) can be made arbitrarily close to the 1-NN rule using a sufficiently large bandwidth γ. □

Aside from the extreme case in which 1-norm SVM reduces to 1-NN classification, the 1-norm SVM method can, in general, be interpreted as a *weighted* nearest neighbor classification algorithm as depicted in Fig. 1. In this figure, every support vector in 1-norm SVM exerts some influence in its vicinity in the instance space $\mathcal{X}$, where the “amount” of influence is determined by its weight λ_i. For a new instance x, the prediction rule in (2) becomes a weighted nearest neighbor rule, where the weight of a neighbor x_i is determined by the product of its influence λ_i and its similarity $S(x, x_i)$ to the new instance x. Such interpretation of 1-norm SVM holds because $\lambda_i \geq 0$ for all $i \geq 1$. Later in Sect. 5, we will show that 1-norm SVM remains superior in its predictive accuracy over the k-NN classifier, despite the apparent similarity between the two methods.

4.4 Neural networks

In addition to SVM and nearest neighbor classifiers, the 1-norm SVM method is closely connected to neural networks as well. This can be observed in the decision rule in (2) or (5), which can be interpreted as a neural network with one hidden layer and one output node as depicted in Fig. 2. The similarity functions $S_j = S(\cdot, x_j)$ form the activation functions in the hidden nodes, whereas the bias term λ_0 is the activation threshold at the output node. When similarity functions are radial, i.e. functions of distance, such neural networks are commonly referred to as *radial basis function* (RBF) networks.

This connection between 1-norm SVM and two-layer neural networks leads to two important observations that are related to the risk of *under-fitting* and *over-fitting* respectively. The first observation shows that 1-norm SVM is not susceptible to under-fitting because its decision rule has the capacity to approximate any function arbitrarily well if the similarity function is radial. The second observation shows that 1-norm SVM is not susceptible to over-fitting because its regularization term minimizes the “size” of the weights at the output layer of the neural network.

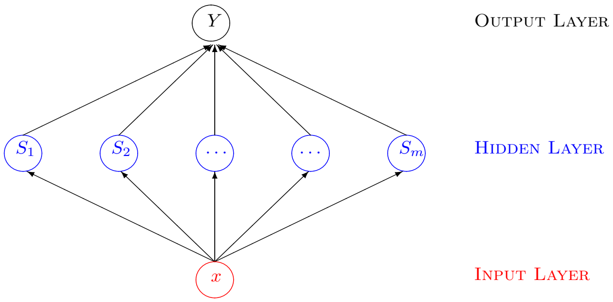

Fig. 2 The 1-norm SVM method can be interpreted as a method of function approximation using two-layer neural networks. In this figure, $S_i = S(\cdot, x_i) : \mathcal{X} \to \mathbb{R}$ is the similarity function with the training instance x_i. The activation threshold at the output node is the bias term λ_0

4.4.1 Universal function approximation

The first key observation is related to function approximation. Ideally, we would like the decision rule in the neural network in Fig. 2 to be as close as possible to the optimal Bayes rule. In principle, therefore, classification is reduced to function approximation. However, it has been established that if the instance space is the Euclidean plane $\mathbb{R}^n$ and the similarity function is radial, then the RBF neural network of Fig. 2 with a fixed bandwidth is capable of *universal function approximation* under mild conditions (Park and Sandberg 1991). Consequently, if the training set is sufficiently large, such that 1-norm SVM can freely choose its support vectors in the plane, then 1-norm SVM has the capacity to approximate the optimal decision rule arbitrarily well. Hence, its risk for under-fitting is limited.

4.4.2 Size of the weights

The second key observation is related to the size of the weights at the output layer. Given an instance space $\mathcal{X}$ and a two-layer neural network with fixed activation functions at the hidden nodes, let $\mathcal{H}$ denotes the set of all hypotheses that can be produced by the two-layer neural network. Suppose $(x, y) \sim \mathcal{D}$ are always drawn i.i.d. Then, Bartlett (1997) has shown that the following bound on true risk (prediction error rate) holds uniformly for all $h \in \mathcal{H}$ with a probability of at least $1 - \delta$ over the random choice of m training examples [Theorem 1 in Bartlett (1997)]:

$$R_{\mathcal{D}}(h) \le \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\{y_i\, h(x_i) < M\} + \epsilon(M, m, \delta),$$

where $R_{\mathcal{D}}(h) = \mathbb{P}_{(x,y)\sim\mathcal{D}}\,(y\, h(x) < 0)$ is the true risk. Using the notions of the fat-shattering dimension and Lipschitz continuity, it can be shown that $\epsilon(M, m, \delta) = \tilde{O}(||\lambda||_1)$ for the decision rule of the 1-norm SVM method (Bartlett 1997). As a result, the true risk is bounded uniformly across the hypothesis space $\mathcal{H}$ by:

$$R_{\mathcal{D}}(h) \le ||\,\lambda\,||_1 + \frac{C(M)}{m} \sum_{i=1}^{m} \mathbb{I}\{y_i\, h(x_i) < M\}, \tag{9}$$

where $C(M)$ is a function of M. A similar bound can be obtained that holds uniformly for all $0 < M < 1$ (Bartlett 1997). Contrasting the latter bound with the earlier margin-based bound in (6) suggests that minimizing $||\lambda||_1$ in the 1-norm SVM method plays a role that is similar to minimizing the complexity of the hypothesis space. A similar conclusion can be inferred more directly using Lagrange duality.[5]

Because the activation functions at the hidden nodes in 1-norm SVM are not fixed, since they do depend on the random choice of training examples, the bound in (9) does not hold for 1-norm SVM. Nevertheless, it provides an informal justification to the use of 1-norm SVM with indefinite kernels since it compares favorably well with the objective function given in (4). In particular, the first term is the ℓ_1 regularization term, which is identical in both expressions. Moreover, the second term in (9) is related to the hinge loss, which is the second term of the objective function in 1-norm SVM. In 1-norm SVM, both terms are minimized.

4.5 Structural risk minimization

The connection between 1-norm SVM and neural networks reveals that 1-norm SVM can be interpreted as a method of striking a balance between under-fitting and over-fitting. A similar conclusion can be established using the *leave-one-out* (LOO) error estimation method. To do this, we begin with the following lemma.

Lemma 2 *Let $S_m = \{(x_1, y_1), \dots, (x_m, y_m)\}$ be a fixed set of m training examples, which is used to train the 1-norm SVM. Let $\lambda^\star$ and $\xi^\star$ be the optimal solutions of the LP in (4). Also, let e_{LOO} be the expected leave-one-out validation error rate on the same training set. Then:*

$$e_{LOO} \leq \frac{||\lambda^\star||_0}{m} + \frac{||\xi^\star||_0}{m} \tag{10}$$

Here, $||z||_0$ denotes the number of non-zero entries in z.

Proof Let $\lambda^\star$ and $\xi^\star$ be the optimal solutions to the 1-norm SVM in (4). If $\xi_i^\star = \lambda_i^\star = 0$, then the ith training example was classified correctly and it will continue to be classified correctly if it is the only example removed from the training set. The latter statement holds because removing the ith example from the training set is equivalent to adding the new constraint $\lambda_i = \xi_i = 0$ to the LP formulation (4), which is the original optimal value of $\lambda_i^\star$ and $\xi_i^\star$. Because the new feasibility region is a subset of the original feasibility region and it contains the original optimal solution, the optimal solution remains unchanged. Hence:

$$e_{LOO} \leq \frac{||\,\lambda^\star + \xi^\star\,||_0}{m} \leq \frac{||\lambda^\star||_0 + ||\xi^\star||_0}{m},$$

□

Theorem 1 *Let h_S be a random variable that stands for the hypothesis produced by 1-norm SVM when trained on a randomly selected training set S. Let $R_{\mathcal{D}}(h_S)$ be the true risk (prediction error rate) of the hypothesis h_S. Then:*

$$\mathbb{E}_{S_{m-1}}[R_{\mathcal{D}}(h_{S_{m-1}})] \leq \frac{\mathbb{E}_{S_m}||\lambda||_0}{m} + \frac{\mathbb{E}_{S_m}||\xi||_0}{m} \tag{11}$$

Here, expectation of the true risk is taken over all possible training sets of size $m-1$ whereas remaining expectations are taken over all possible training sets of size m.

[5] Using Lagrange duality, it is well-known that adding ℓ_p regularization on some optimization variable λ in the objective function is equivalent to setting some upper bound on $||\lambda||_p$ (see for instance Abu-Mostafa et al. 1997). This shows that minimizing $||\lambda||_1$ in the 1-norm SVM mehod indeed plays the role of minimizing the complexity of the hypothesis space.

Proof By the Luntz–Brailovsky theorem (Luntz and Brailovsky 1969; Vapnik and Chapelle 2000), we have:

$$\mathbb{E}_{S_{m-1}}[R_{\mathcal{D}}(h_{S_{m-1}})] = \mathbb{E}_{S_m}[e_{LOO}], \tag{12}$$

where e_{LOO} is the leave-one-out validation error. Using Eq 12 and Lemma 2 yields the desired result. □

The tradeoff in Eq 11 is analogous to the classical tradeoff in estimation between bias and variance (Hastie et al. 2001). On one hand, one can fit the training set perfectly, e.g. by using a radial similarity function with a sufficiently large bandwidth that effectively turns 1-norm SVM into a 1-NN classifier, but the number of support vectors becomes at its worst, hence high variance. On the other hand, one can choose a very small number of support vectors but this tends to increase the empirical risk (training error rate), hence high bias. In the 1-norm SVM formulation in (4), the cost function penalizes both training error (bias) and the number of support vectors (variance) simultaneously by penalizing the $||\cdot||_1$ of slack variables ξ and weights λ. Because minimizing $||\cdot||_1$ promotes *sparsity* (Boyd and Vandenberghe 2004), Corollary 1 states that 1-norm SVM can be interpreted as a method of minimizing the expected true risk. Hence, it is an approximate implementation of the structural risk minimization (SRM) induction principle (Vapnik 1999).

5 Experiments and results

In the previous section, we described theoretically why 1-norm SVM was a viable tool for classification with indefinite similarities. In this section, we present experimental results of applying 1-norm SVM to synthetic and real-world classification problems, which demonstrate its effectiveness in handling indefinite similarity functions.[6]

5.1 Synthetic datasets

First, 1-norm SVM was tested on six synthetic datasets depicted in Fig. 3. In these datasets, the radial basis function (RBF) $S(x_i, x_j) = \exp\{-\gamma\ ||x_i - x_j||_2^2\}$ was used, where the bandwidth parameter γ was selected using a grid search on a separate validation set. Figure 4 plots the test error rate as a function of training set size m, with the Bayes risk for each classification problem indicated in the legend bar. As shown in Fig. 4, the test error rate approaches the optimal Bayes risk for sufficiently large training sets in all six classification problems. This test verifies that 1-norm SVM is capable of producing accurate decision boundaries for various complex mixtures of classes, which is consistent with the universal approximation property of the RBF similarity function discussed earlier in Sect. 4.4.

5.2 Real datasets

For real datasets, we compared the performance of 1-norm SVM against popular classification algorithms for both PSD and non-PSD similarity functions. We will first describe the datasets and test methodology, and discuss the test results afterwards.

[6] The datasets and MATLAB implementation routines will be made available at: http://mine.kaust.edu.sa.

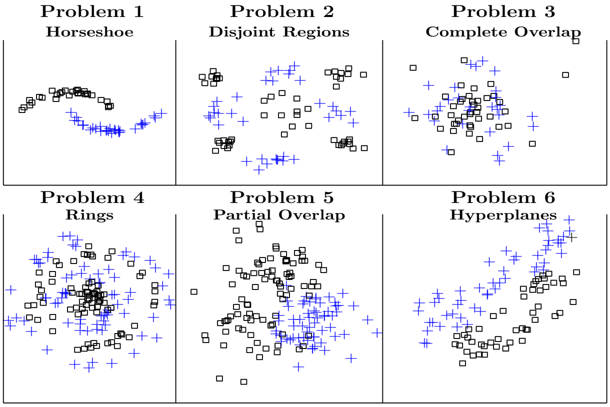

Fig. 3 The six synthetic datasets that are used in evaluating 1-norm SVM. The Bayes risk of each dataset is indicated in the legend bar in Fig. 4

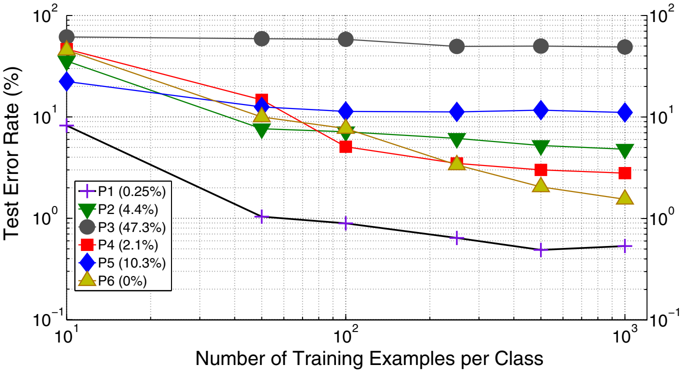

Fig. 4 Performance of 1-norm SVM on the six synthetic datasets. Each error rate, plotted in a log-scale, is an average of five i.i.d training/test sets. P1, ..., P6 stands for Problem 1, ..., Problem 6 shown in Fig. 3. The optimal Bayes risks are indicated in the *legends bar*. In the y-axis, each grid line between 10^z and 10^{z+1} is to be read as $1 \times 10^z,\ 2 \times 10^z, \ldots 9 \times 10^z$. For example, the grid lines from 10^1 to 10^2 correspond to the values $10,\ 20,\ \ldots,80,\ 90$

5.2.1 Datasets

The following datasets and similarity functions were used.

(A) **IMDB**: This is a graph-based dataset that contains movies released between 1996 and 2001 (Macskassy and Provost 2007). The binary class label identifies whether the opening weekend box-office receipts exceeded \$2 million or not. An edge weight between two movies is the number of common production companies, actors, producers, or directors. In our implementation, all edge weights were normalized to fall in the range [0, 1], and the following similarity functions were used:

   (a) *PSD*: The Jaccard index $S_{i,j} = \frac{\sum_k \min\{w_{i,k}, w_{j,k}\}}{\sum_k \max\{w_{i,k}, w_{j,k}\}}$, where $w_{i,k}$ is the edge weight.
   (b) *Non-PSD*: Edge weight $S_{i,j} = w_{i,j}$ and $S_{i,i} = 1$.

(B) **Word-Sim-353**: This dataset contains human-judged similarities between English words (Finkelstein et al. 2002). All similarities are again normalized to fall in the range $[0,\ 1]$ and self-similarity is set to unity. We grouped words into two categories: 'living' versus 'non-living', and used the two similarity functions specified earlier for the IMDB dataset. Examples of the 'living' class include *children*, *Maradona*, *brother*, *carnivore*, and *mammal*.

(C) **Caltech-101**: This dataset contains images of various objects (Fei-Fei et al. 2004). We grouped images of 'Big Cats', 'Winged Insects', and 'Flowers' into three classes and trained three separate binary classifiers between every pair of classes. Each image was converted into a histogram using the two MATLAB commands `rgb2gray` and `imhist`, and Laplace normalization was used. This effectively represents the ith image by a probability distribution p_i. We, then, used the following two similarity functions:

   (a) *PSD*: The intersection (a.k.a. overlapping coefficient), which is given by $S_{i,j} = \sum_k \min\{p_{i,k},\ p_{j,k}\}$.

   (b) *Non-PSD*: We used $S_{i,j} = \max\{0,\ 1 - 0.1 \times D(p_i\,||p_j)\}$, where $D(p_i||p_j)$ is the symmetrized Kullback–Leibler divergence.[7]

(D) **Splice**: This is a biological sequence classification dataset (Noordewier et al. 1991) that was downloaded from the UCI repository (Lichman 2013). Each example is a 60-letter DNA sequence. We performed classification between the two classes EI and IE. The similarity functions used were:

   (a) *PSD*: We used the implementation of string subsequence kernels given in Soman et al. (2009). Because string kernels can grow quite rapidly, we normalized using the cosine similarity: $S_{i,j} = \frac{K_{i,j}}{\sqrt{K_{i,i} \cdot K_{j,j}}}$.

   (b) *Non-PSD*: We used the longest-common-subsequence (LCS) between two strings. Because each string is 60 letters in length, we set $S_{i,j} = LCS(x_i,\ x_j)/60$.

(E) **CNAE-9**: This is a text classification dataset available at the UCI repository, where each text is represented using a bag of words. The dataset contains nine classes and we randomly selected five binary classification problems: 1-versus-5, 5-versus-4, 6-versus-8, 3-versus-9, and 2-versus-7.[8] These are represented by P1 through P5 in Table 2 respectively. The two similarity functions are:

   (a) *PSD*: The cosine similarity $S_{i,j} = \frac{x_i^T\, x_j}{||x_i|| \cdot ||x_j||}$, which is commonly used for text classification tasks (Chen et al. 2009a).

   (b) *Non-PSD*: The second similarity function used is a variant to the first. Specifically, we have $S_{i,j} = \frac{v^T\, v}{||x_i|| \cdot ||x_j||}$, where $v_k = \min\{x_{i,k},\ x_{j,k}\}$.

(F) **Ionosphere, Australian, Breast Cancer, Haberman, and Diabetes**: These are five binary classification problems with numeric features available at the UCI repository. We used the following similarity functions:

   (a) *PSD*: The RBF kernel $S_{i,j} = e^{-\gamma\, ||x_i - x_j||_2^2}$, which is considered the default similarity function for numeric attributes in popular SVM packages such as LIBSVM (Chang and Lin 2001).

[7] The reason behind choosing 0.1 is because 95 % of pairwise distances are less than 10.

[8] We perfomred a random permutation of the set of integers $\{1, 2, \ldots, 9\}$. Each pair of adjacent labels was used as a binary classification problem, where the 9th label is trainied versus the 1$^{\text{st}}$.

(b) *Non-PSD*: The sigmoid kernel $S_{i,j} = \tanh\{\gamma \cdot x_i^T x_j + r\}$, which is popular due to its origins in neural networks. To ensure that the kernel matrix is not PSD, we fixed $r = -1$.[9]

5.2.2 Test methodology and results

When the similarity function is PSD, we compared performance of 1-norm SVM versus the standard form of SVM in (1). For each dataset, the value of the tradeoff constant C was selected using fivefold cross validation for $C \in \{2, 4, 8, 16, 32\}$. When the RBF kernel is used, the bandwidth γ is also selected using fivefold cross validation in the grid $\gamma \in \{2^{-15}, 2^{-14}, \ldots, 2^{-1}, 1\}$. SVM was implemented using the LIBSVM library (Chang and Lin 2001), whereas 1-norm SVM was implemented using the Gurobi solver (Gurobi Optimization 2012). In all classification problems, we reported the average test error rate of five random training-to-test splits, with a training-to-split ratio of 4:1. The same split is always used in both SVM and 1-norm SVM.

When the similarity function is indefinite (non-PSD), we compared the performance of 1-norm SVM against the three dominant methods used in the literature:

1. *Non-convex optimization*: This was implemented using the LIBSVM library with its `-t 4` option. When the similarity matrix is non-PSD, the LIBSVM package seeks a stationary point using non-convex optimization (Lin and Lin 2003).
2. *Kernel approximation*: PSD kernel approximation was tested using the three methods discussed earlier in Sect. 2: (1) the *denoise* method, (2) the *flip* method, and (3) the indefinite SVM proposed by Luss and d'Aspremont (2009). The *denoise* and *flip* methods were implemented by supplying the modified (PSD) kernel matrix to LIBSVM using the `-t 4` option. The indefinite SVM method was tested using the implementation available for download at the authors' website.
3. *SVM in Krien Spaces*: SVM in Krien spaces was implemented using the ESVM algorithm described in Loosli et al. (2013). ESVM comprises of two main steps: (1) eigendecomposition, and (2) SVM training. LIBSVM was used for the SVM training step.

In addition, we also included k-NN to serve as a benchmark for similarity-based classification.[10] In all methods, hyper-parameters were selected using cross validation and grid search, implemented separately for each individual method. Test results for PSD and non-PSD similarity functions are shown in Tables 1 and 2 respectively. Because the datasets are balanced, we used classification error rate as a measure of performance. All results reported here are based on the best selected hyper-parameters of these methods.

5.3 Discussion

In this section, we review the test results when using PSD and non-PSD similarity functions for the 16 real datasets described earlier.

5.3.1 Positive semidefinite similarities

We begin our discussion by looking into the test results for positive semidefinite (PSD) similarities. As shown in columns 2 and 3 of Table 1, when the similarity function is PSD,

[9] It has been shown that the sigmoid kernel is PSD only if $r \geq 0$ (Burges 1999). However, using $r < 0$ tends to perform better (Lin and Lin 2003).

[10] Because ESVM was found to be competitive to relevance vector machine (RVM) (Loosli et al. 2013), RVM was not included in our experiments.

Table 1 Average test error rate results on 16 datasets using the positive semidefinite (PSD) similarity functions described in Sect. 5.2

Dataset (m)	1-norm SVM (%)	SVM (%)	k-NN (%)
IMDB (1441)	16.0	**15.6**	21.2
WORD- SIM-353 (437)	**12.9**	13.7	13.3
CALTECH-101-P2 (368)	26.0	**24.0**	40.0
CALTECH-101-P3 (379)	19.8	**19.2**	33.1
CALTECH-101-P1 (387)	**31.7**	**31.7**	38.7
SPLICE (1527)	6.56	**5.79**	10.3
CNAE-9-P1 (240)	0.56	**0**	0.42
CNAE-9-P5 (240)	2.05	**1.15**	1.67
CNAE-9-P3 (240)	1.67	**0.94**	2.50
CNAE-9-P2 (240)	**0.44**	1.22	1.25
CNAE-9-P4 (240)	1.89	**0.83**	1.67
IONOSPHERE (351)	7.14	**6.57**	14.6
AUSTRALIAN (690)	16.6	16.9	**14.4**
BREAST CANCER (699)	**3.15**	3.51	4.75
HABERMAN (398)	30.2	31.2	**22.5**
DIABETES (768)	28.9	27.9	**26.3**

Bold values indicate the best result (i.e. smallest error rate)
Here, the number of training examples m for each dataset is shown in parentheses

performance of 1-norm SVM is comparable to that of SVM for all 16 datasets. When running statistical significance tests, we find no statistically significant evidence that one method outperforms the other at the 95 % confidence level. For example, the two-tailed Wilcoxon's signed rank test (Demšar 2006) gives a value of $p = 0.155$. By contrast, both algorithms tend to outperform k-NN in classification accuracy. This validation verifies that 1-norm SVM is a viable algorithm for binary classification even when the similarity function is positive semidefinite (PSD). Such experimental evidence agrees with earlier conclusions (Zhu et al. 2004).

5.3.2 Indefinite similarities

In contrast to the previous case, the use of indefinite similarity functions presents an entirely different picture. When comparing the test error rate of 1-norm SVM (shown in column 3 of Table 2) with the other methods (in columns 4-9), we find that 1-norm SVM and ESVM (i.e. learning in Krein spaces) outperform all other methods significantly in nearly all the datasets. Performance of ESVM, however, is very similar to that of 1-norm SVM, which is quite intriguing given the very different approaches employed by the two algorithms.

Nevertheless, unlike ESVM whose solution is quite dense, the 1-norm SVM method yields very sparse solutions so that prediction time is faster. In fact, 1-norm SVM yields solutions that are often 10–20 times, sometimes even 100 times, sparser than ESVM. Table 3 lists the number of support vectors used by both methods.

In order to verify statistical significance at the 95 % confidence level, we used Holm's step-down procedure for multiple comparisons applied to the two-tailed Wilcoxon's signed rank test (Demšar 2006; Holm 1979). More specifically, each null hypothesis H_i asserts

Table 2 Average test error rate results on 16 datasets using the indefinite (non-PSD) similarity functions described in Sect. 5.2

Dataset (m)	β[1]	1-NORM SVM (%)	k-NN (%)	NCO (%)	Kernel approximation			Krein space
				SVM (%)	DEN (%)	FLIP (%)	ISVM (%)	ESVM (%)
IMDB (1441)	0.01	18.8	26.1	17.7	**17.6**	**17.6**	18.1	18.8
WORD- SIM-353 (437)	0.03	**14.7**	14.9	15.6	15.2	15.4	**14.7**	15.8
CALTECH-P2 (368)	0.05	22.1	27.7	44.0	62.0	41.6	37.5	**20.7**
CALTECH-P3 (379)	0.08	**22.8**	26.7	40.7	54.5	36.5	32.0	24.1
CALTECH-P1 (387)	0.09	**30.1**	35.6	40.9	50.8	39.5	38.0	31.7
SPLICE (1527)	0.10	5.70	7.49	5.74	5.97	6.95	5.64	**4.86**
CNAE-9-P1 (240)	0.32	**0**	8.30	13.2	6.17	0.17	**0**	**0**
CNAE-9-P5 (240)	0.33	3.72	6.25	10.9	6.61	1.94	**1.25**	3.50
CNAE-9-P3 (240)	0.34	**2.50**	24.6	25.3	25.4	8.67	3.75	3.33
CNAE-9-P2 (240)	0.34	**0.33**	22.5	21.6	20.7	5.78	2.50	1.67
CNAE-9-P4 (240)	0.34	2.56	4.17	16.3	9.67	2.17	**1.25**	1.67
IONOSPHERE (351)	1.0	**10.3**	18.6	37.7	70.3	66.0	36.0	**10.3**
AUSTRALIAN (690)	1.0	**12.2**	18.4	40.7	40.7	91.9	47.7	15.1
BREAST CANCER (699)	1.0	4.32	**2.88**	32.9	32.9	96.4	30.1	5.76
HABERMAN (398)	1.0	26.6	**24.8**	26.6	40.1	27.2	*[2]	**26.6**
DIABETES (768)	1.0	**22.7**	33.2	33.3	70.3	54.2	35.8	25.9

Bold values indicate the best result (i.e. smallest error rate)

In this table, $0 \leq \beta \leq 1$ is a measure of how indefinite the similarity funciton is. In particular, a value of $\beta = 0$ corresponds to PSD similarity functions while a value of $\beta = 1$ corresponds to negative semidefinite similarities. The rows in the table are ordered by the value of β. The acronym NCO stands for non-convex optimization, DEN for the denoise method, FLIP for the flip method, ISVM for indefinite SVM, while ESVM stands for the eigen-decomposition SVM method

[a] $\beta = \frac{\sum_i |\lambda_i| \cdot \mathbb{I}\{\lambda_i < 0\}}{\sum_i |\lambda_i|}$, where λ_i are eigenvalues of the similarity matrix

[b] The algorithm failed to terminate

that 1-norm SVM and the ith alternative classifier have similar performance. When H_i is tested using the two-tailed Wilcoxon's signed rank test, the resulting p values are shown in Table 4. Using a confidence level of 95 % in Holm's step down procedure, we find that the null hypothesis is rejected for non-convex optimization, k-NN, and all kernel approximation methods. This confirms that 1-norm SVM outperforms non-convex optimization and kernel approximation with a statistically significant evidence.

However, there is no statistically significant evidence at the 95 % confidence level that 1-norm SVM outperforms ESVM in terms of predictive accuracy. Here, it is perhaps worth reiterating that the 1-norm SVM significantly outperforms ESVM in terms of sparsity of solutions as shown in Table 3. Therefore, the 1-norm SVM method achieves the highest predictive accuracy among all methods that learn with indefinite similarities, while also retaining sparsity of the support vector set.

Finally, it is worth pointing out that indefinite similarity functions in our evaluation led to lower test error rates than PSD similarity functions in roughly 50 % of the datasets. This includes, most notably, the datasets: CALTECH- 101- P2, AUSTRALIAN, HABERMAN, and DIABETES. Therefore, even for classification problems where PSD similarity functions are readily

Table 3 The number of support vectors (SVs) used by 1-norm SVM and ESVM for the 16 classification problems with indefinite similarity functions

Datasets	No. of training examples	No. of SVs in 1-norm svm	No. of SVs in esvm
IMDB	1441	630	1438
WORD- SIM- 353	437	26	436
CALTECH- 101- P2	368	39	386
CALTECH- 101- P3	379	47	379
CALTECH- 101- P1	387	35	386
SPLICE	1527	224	1527
CNAE- 9- P1	240	2	233
CNAE- 9- P5	240	21	240
CNAE- 9- P3	240	23	238
CNAE- 9- P2	240	2	235
CNAE- 9- P4	240	23	240
IONOSPHERE	351	39	351
AUSTRALIAN	690	7	690
BREAST CANCER	699	20	699
HABERMAN	398	28	398
DIABETES	768	13	768

Table 4 In this table, the second column lists the p values in increasing order of the two-tailed Wilcoxon's signed rank test

Null hypothesis (H_i)	p value	Adjusted critical value
1-norm SVM versus SVM with non-convex optimization	0.0003	0.0083
1-norm SVM versus denoise	0.0008	0.0100
1-norm SVM versus k-NN	0.0016	0.0125
1-norm SVM versus flip	0.0052	0.0167
1-norm SVM versus indefinite SVM	0.0107	0.0250
1-norm SVM versus ESVM	0.0771	0.0500

The last column shows the critical values when Holm's step-down procedure is used at the 95 % confidence level

available, learning with non-PSD kernels remains important because it can result in a better classification accuracy.

6 Conclusion

Extensive research effort has been devoted lately to classification with indefinite similarities. In this paper, we show theoretically and experimentally how the 1-norm support vector machine is a better method for handling indefinite similarities. The 1-norm SVM method formulates large-margin separation as a convex linear programming (LP) problem without requiring that the similarity function be positive semidefinite (PSD). It uses the indefinite similarity function directly without any transformation, and, hence, it always treats both training and test examples consistently. Furthermore, by relating 1-norm SVM with neural networks and error bounds of the leave-one-out estimation method, 1-norm SVM can be interpreted as an approximate implementation of the structural risk minimization (SRM) induction principle. Hence, it is robust against the risks of under-fitting and over-fitting. Finally, 1-norm SVM indeed achieves the highest accuracy among all previous methods for classification with indefinite similarities with a statistically significant evidence, while also retaining sparsity of the support vector set.

Acknowledgments We would like to express our gratitude to the anonymous reviewers who provided valuable feedback and suggestions that greatly improved the manuscript. We also thank the Saudi Arabian Oil Company (Saudi Aramco) and King Abdullah University of Science and Technology (KAUST) for supporting this research.

References

Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H. T. (2012). *Learning from data*. AMLBook.

Balcan, M. F., Blum, A., & Srebro, N. (2008). A theory of learning with similarity functions. *Machine Learning*, *72*(1–2), 89–112.

Bartlett, P. L. (1997). For valid generalization, the size of the weights is more important than the size. *Advances in Neural Information Processing Systems (NIPS)*, *9*, 134.

Lichman, M. (2013). *UCI machine learning repository*. Irvine, CA: University of California, School of Information and Computer Sciences. http://archive.ics.uci.edu/ml

Boser, B. E., Guyon, I., & Vapnik, V. (1992) A training algorithm for optimal margin classifiers. In *Fifth annual workshop on computational learning theory* (pp. 144–152).

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge university press.

Bradley, P. S., & Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. In *ICML*.

Burges, C. (1999). Geometry and invariance in kernel based methods. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in kernel methods–support vector learning* (pp. 89–116). Cambridge, MA: MIT Press.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*, 121–167.

Chang, C., & Lin, C. J. (2001). *LIBSVM: A library for support vector machines* (online). http://www.csie.ntu.edu.tw/cjlin/libsvm.

Chen, J., & Ye, J. (2008). Training SVM with indefinite kernels. In *Proceedings of ICML* (pp. 136–143).

Chen, Y., Garcia, E. K., Gupta, M. R., Rahimi, A., & Cazzanti, L. (2009a). Similarity-based classification: Concepts and algorithms. *JMLR*, *10*, 747–776.

Chen, Y., Gupta, M. R., & Recht, B. (2009b). Learning kernels from indefinite similarities. In *Proceedings of ICML* (pp. 145–152).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *JMLR*, *7*, 1–30.

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *IEEE CVPR: Workshop on generative-model based vision*.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., et al. (2002). Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, *20*(1), 116–131.

Fung, G. M., & Mangasarian, O. L. (2004). A feature selection Newton method for support vector machine classification. *Computational Optimization and Applications*, *28*, 185–202.

Graepel, T., Herbrich, R., Bollmann-Sdorra, P., & Obermayer, K. (1999). Classification on pairwise proximity data. In M. J. Kearns, S. A. Solla, & D. A. Cohn (Eds.), *Advances in NIPS* (pp. 438–444). MIT Press.

Gurobi Optimization I. (2012). *Gurobi optimizer reference manual*. http://www.gurobi.com.

Haasdonk, B. (2005). Feature space interpretation of svms with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(4), 482–492.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction. Springer series in statistics* (2nd ed.). Springer.

Hilario, M., & Kalousis, A. (2008). Approaches to dimensionality reduction in proteomic biomarker studies. *Briefings in Bioinformatics*, *9*(2), 102–118.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, *6*(2), 65–70.

IBM I. (2015). *Cplex optimizer*. http://www.ibm.com/software/commerce/optimization/cplex-optimizer/.

Lin, H. T., & Lin, C. J. (2003). *A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods*. Tech. rep., Department of Computer Science, National Taiwan University. http://www.csie.ntu.edu.tw/cjlin/papers/tanh.pdf.

Liu, H., Motoda, H., Setiono, R., & Zhao, Z. (2010). Feature selection: An ever evolving frontier in data mining. In *4th workshop on feature selection in data mining (FSDM 10), PAKDD*. pp. 4–13.

Loosli, G., Ong, C. S., & Canu, S. (2013). *SVM in Krein spaces*. Tech. rep. http://hal.archives-ouvertes.fr/hal-00869658/.

Luntz, A., & Brailovsky, V. (1969). On estimation of characters obtained in statistical procedure of recognition. *Technicheskaya Kibernetica*, *3*(6) (in Russian).

Luss, R., & d'Aspremont, A. (2009). Support vector machine classification with indefinite kernels. *Mathematical Programming Computation*, *1*(2–3), 97–118.

Macskassy, S. A., & Provost, F. (2007). Classification in networked data: A toolkit and a univariate case study. *JMLR*, *8*, 935–983.

Mangasarian, O. L. (1998). *Generalized support vector machines*. Tech. Rep. Mathematical Programming Technical Report 98-14, University of Wisconsin.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. Cambridge: MIT Press.

Noordewier, M. O., Towell, G. G., & Shavlik, J. W. (1991). Training knowledge-based neural networks to recognize genes in dna sequences. In R. P. Lippmann, J. E. Moody, & D. S. Touretzky (Eds.), *Advances in NIPS* (pp. 530–536). Morgan-Kaufmann.

Ong, C. S., Mary, X., Canu, S., & Smola, A. J. (2004). Learning with non-positive kernels. In *ICML*.

Park, J., & Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks. *Neural Computation*, *3*(2), 246–257.

Pekalska, E., Paclik, P., & Duin, R. P. (2001). A generalized kernel approach to dissimilarity-based classification. *JMLR*, *2*, 175–211.

Schapire, R. E., Freund, Y., Bartlett, P., & Lee, W. S. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, *26*(5), 1651–1686.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. Cambridge: MIT Press.

Soman, K. P., Loganathan, R., & Ajay, V. (2009). *Machine Learning with SVM and other Kernel methods*. PHI Learning.

Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *JMLR*, *1*, 211–244.

Vapnik, V., & Chapelle, O. (2000). Bounds on error expectation for support vector machines. *Neural Computation*, *12*(9), 2013–2036.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, *10*(5), 988–999.

Wu, G., Zhang, Z., & Chang, E. Y. (2005). *An analysis of transformation on non-positive semidefinite similarity matrix for kernel machines*. Tech. rep., UCSB.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., et al. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, *14*(1), 1–37.

Ying, Y., Campbell, C., & Girolami, M. (2009). Analysis of SVM with indefinite kernels. *Advances in NIPS*, *22*, 2205–2213.

Zhu, J., Rosset, S., Hastie, T., & Tibshirani, R. (2004). 1-norm support vector machines. *Advances in Neural Information Processing Systems (NIPS)*, *16*, 49–56.

Zou, H. (2007). An improved 1-norm SVM for simultaneous classification and variable selection. In *Proceedings of the 11th international conference on artificial intelligence and statistics* (pp. 675–681).