JMLR: Workshop and Conference Proceedings 20 (2011) 97–112 Asian Conference on Machine Learning

Support Vector Machines Under Adversarial Label Noise

Battista Biggio BATTISTA.BIGGIO@DIEE.UNICA.IT
Dept. of Electrical and Electronic Engineering
University of Cagliari
Piazza d'Armi, 09123, Cagliari, Italy and
Blaine Nelson BLAINE.NELSON@WSII.UNI-TUEBINGEN.DE
Dept. of Mathematics and Natural Sciences
Eberhard-Karls-Universität Tübingen
Sand 1, 72076, Tübingen, Germany and
Pavel Laskov PAVEL.LASKOV@UNI-TUEBINGEN.DE
Dept. of Mathematics and Natural Sciences
Eberhard-Karls-Universität Tübingen
Sand 1, 72076, Tübingen, Germany

Editor: Chun-Nan Hsu and Wee Sun Lee

Abstract

In *adversarial classification* tasks like spam filtering and intrusion detection, malicious adversaries may manipulate data to thwart the outcome of an automatic analysis. Thus, besides achieving good classification performances, machine learning algorithms have to be *robust* against adversarial data manipulation to successfully operate in these tasks. While support vector machines (SVMs) have shown to be a very successful approach in classification problems, their effectiveness in adversarial classification tasks has not been extensively investigated yet. In this paper we present a preliminary investigation of the robustness of SVMs against adversarial data manipulation. In particular, we assume that the adversary has control over some training data, and aims to subvert the SVM learning process. Within this assumption, we show that this is indeed possible, and propose a strategy to improve the robustness of SVMs to training data manipulation based on a simple kernel matrix correction.

Keywords: Support Vector Machines, Adversarial Classification, Label Noise

1. Introduction

Machine learning algorithms have been successfully applied in a wide set of applications in the last few decades. What makes them particularly attractive is that they can elicit important properties from a set of given samples (drawn from some underlying, but *unknown* probability distribution) to infer the underlying concept and then classify unseen samples (drawn from the same distribution) with very high accuracies. This is well known in the machine learning and pattern recognition literature as *generalization capability* (Duda et al., 2000; Bishop, 2007). Due to their ability to generalize, machine learning algorithms have been widely adopted in security applications as well; e.g., in spam filtering and intrusion detection. However, these applications differ from the standard machine learning setting since *malicious adversaries* can adaptively manipulate their data to mislead the outcome

of an automatic analysis. More formally, this can be modeled as a *change* in the underlying probability distribution that generates the samples. Thus, in these settings, the classical *stationarity* assumption of machine learning is violated (Dalvi et al., 2004; Barreno et al., 2010). For instance, spammers often modify their emails by obfuscating words which typically appear in known spam (e.g., the word "replica" can be misspelled as "repl1c@"), or by adding words which are likely to appear in legitimate emails (Dalvi et al., 2004; Lowd and Meek, 2005; Kolcz and Teo, 2009). Similarly, hackers can successfully hide the malicious code within specially crafted network packets that can evade an intrusion detection system (IDS) without compromising the effectiveness of their attack (see, e.g., Fogla et al., 2006). The above examples of attacks are based on exploiting specific weaknesses and knowledge of the classification algorithm. As a consequence, analyzing the vulnerabilities of classifiers and their *robustness* to attacks to better understand how their security may be improved, has recently attracted growing interest from the scientific community.

Attacks against learning methods can be carried out either during the training or testing stages. Attacks at the test stage exploit characteristics of the underlying classes that can be modified without affecting the true classification but that deleteriously impact the discriminative model learned from the training data (Globerson and Roweis, 2006; Dekel et al., 2010; Teo et al., 2008). Given knowledge of invariances in the task, the effect of potential testing attacks can be somewhat mitigated by the learning algorithm; however, this comes at a cost of increased complexity of the training problem. Attacks at the training stage attempt to exert a long-lasting impact on learning by modifying the training data. For example, important points in the training data may be changed, so as to make the learning problem more complex. Such attacks introduce *feature noise* to training points. Another possibility is to flip labels of certain points (*label noise*) which has a similar effect on the learning problem. While some previous work has addressed the development of methods robust against feature noise (Bi and Zhang, 2004; Xu et al., 2009), little is known on how learning algorithms are affected by adversarial (rather than random) label noise.

In this paper, we propose a model for the analysis of label noise in support vector learning and develop a modification of the SVM formulation that indirectly compensates for the noise under our model. Our model is based on a simple assumption that any label may be flipped with a fixed probability. We show that this noise can be compensated for by correcting the kernel matrix of SVM with a specially structured matrix, which depends on the noise parameters of our model. Our empirical evaluation shows that the proposed kernel correction is effective against label noise, even in the adversarial case when the flipped labels are carefully selected so as to tilt the separating hyperplane. An advantage of our approach is that, unlike the majority of other robust learning approaches, it does not increase the complexity of SVM training. Moreover, the resulting optimization problem remains quadratic, for which efficient solvers are available.

2. Background

In this section, we briefly review SVMs and previous work that addresses the problem of designing robust SVMs to different kinds of noise.

2.1. Support Vector Machines

Before reviewing SVMs, we formally introduce the classification problem and our notation. The classification problem can be generally stated as the problem of learning a hypothesis or classifier $f : \mathcal{X} \mapsto \mathcal{Y}$ which assigns any input sample $\mathbf{x} \in \mathcal{X}$ to a class $y \in \mathcal{Y}$. In the classical *supervised learning* setting, we are given a set of training data $\{\mathbf{x}_i, y_i\}_{i=1}^n \in \mathcal{X} \times \mathcal{Y}$ drawn from an unknown distribution $P(\mathbf{X}, Y)$, and we want to infer a classifier f which is able to *generalize* well on $P(\mathbf{X}, Y)$, i.e., which can accurately classify unseen samples drawn from that distribution.

In its simplest formulation, an SVM learns a linear classifier for a two-class classification problem. Its decision function can be thus generally written as

$$f(\mathbf{x}; \mathbf{w}, b) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b) \in \{-1, +1\}\,, \tag{1}$$

where $\text{sign}(a) = +1$ (-1) if $a \geq 0$ $(a < 0)$, and $\mathbf{w}$ and b are parameters which determine the position of the decision hyperplane in feature space: its orientation by the hyperplane's normal $\mathbf{w}$ and its displacement by b. The underlying idea of SVMs is to find the hyperplane $(\mathbf{w}, b)$ which has the maximum distance between the nearest training samples of the two classes (a concept known as the classifier's *margin*), since this generally reduces the *generalization error* (Vapnik, 1995). Although originally designed for linearly separable classification tasks (*hard-margin* SVMs), SVMs were subsequently extended to non-linearly separable classification problems by Cortes and Vapnik (1995) (*soft-margin* SVMs), for which some samples are allowed to violate the margin. In particular, a soft margin SVM is learned by solving the following convex quadratic programming (QP) optimization problem:

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

where the margin is maximized by minimizing $\frac{1}{2}\mathbf{w}^\top \mathbf{w}$, and the variables ξ_i, $i = 1, \ldots, n$ (referred to as *slack variables*) represent the extent to which the samples $\mathbf{x}_i$, $i = 1, \ldots, n$, violate the margin. The parameter C tunes the trade-off between the classification error on the training data and margin maximization.

Problem 2 is referred to as the *primal problem*, and it is solved by means of its dual Lagrange multipliers. Its solution is given by $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$, where α_i, $i = 1, \ldots, n$, are the Lagrange multipliers, and by the following constraints: $\sum_{i=1}^n \alpha_i y_i = 0$, and $0 \leq \alpha_i \leq C$, $i = 1, \ldots, n$. Interesting properties of the SVM arise from its *dual*. First, the normal to the hyperplane $\mathbf{w}$ can be expressed as a convex linear combination of the training samples. Second, the solution to the dual problem is *sparse*, and only samples that lie on or within the hyperplane's margin exhibit a non-zero contribution (i.e., their α value is non-zero). In particular, if $\alpha_i = 0$, then the corresponding sample $\mathbf{x}_i$ is correctly classified, lies beyond the margin (i.e., $y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 1$), and is referred to as a *non-support vector*. If $\alpha_i = C$, the corresponding sample violates the margin (i.e., $y_i(\mathbf{w}^\top \mathbf{x}_i + b) < 1$), and it is thus referred to as an *error vector*. Finally, if $0 < \alpha < C$, the sample lies exactly on the margin (i.e., $y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$) and it is called a *support vector*. As a consequence, the displacement b

is typically determined by averaging $\mathbf{w}^\top \mathbf{x}_i - y_i$ over the set of support vectors (where the average is used for improved numerical stability).

The above solution can derived from Problem 2 by the method of Lagrangian multipliers which yields the *dual problem*. In matrix form, this dual can be expressed as

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^\top \mathbf{Q}\alpha - \mathbf{1}_n^\top \alpha \\ \text{s. t.} \quad & 0 \leq \alpha_i \leq C,\ i = 1, \ldots, n, \\ & \sum_{i=1}^{n} \alpha_i y_i = 0, \end{aligned} \tag{3}$$

where $\mathbf{Q} = \mathbf{K} \circ yy^\top$, and $\mathbf{1}_n$ is a column vector of n ones. $\mathbf{K}$ is the kernel matrix, whose elements are $K_{ij} = \mathbf{x}_i^\top \mathbf{x}_j$ in the linear case. Nevertheless, if some non-linear function $\phi : \mathcal{X} \mapsto \Phi$ is used to map training samples into a higher dimensional feature space (where a linear classifier may perform better), then K_{ij} can be generalized to $\phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$. Since only inner products between samples are required to compute the solution and predict the class labels, one does not need to know ϕ explicitly, but only the corresponding *kernel function*. This is known as the *kernel trick*, and, due to its wide adoption, SVMs are often learned by directly solving the dual problem.

2.2. Robust SVMs

To date, many works have proposed modifications to the standard SVM learning algorithm to improve its *robustness* to different kinds of noise, either affecting training or testing data. However, the majority of them require solving non-convex optimization problems, robust optimization problems, or, they require solving problems which have higher complexity than the standard SVM learning algorithm. For instance, even the SVM-like algorithm robust to worst-case feature deletion at testing time proposed by Globerson and Roweis (2006) has a higher computational complexity than standard SVMs, although it is still a convex QP problem.

A set of interesting work on robust SVMs applies robust optimization theory (Bi and Zhang, 2004; Xu et al., 2009). They dealt with *feature noise*, namely, noise which only affects the feature values of each sample. In particular, the main assumption is that each noisy sample will still lie within a ball of given radius with respect to the true (non-noisy) sample, i.e., $\mathbf{x}' = \mathbf{x} + \delta$, and $||\delta|| \leq R$. Notably, Xu et al. (2009) observed that the regularization term in the SVM optimization problem implicitly assumes a spherical noise around samples. In other words, the regularization term is naturally derived from an explicit noise model, and, thus, it should be chosen or modified accordingly.

In addition to approaches which dealt with feature noise, some research also addresses the problem of *label noise*, or investigated the problem of SVMs' robustness under data contamination. Stempfel and Ralaivola (2009) proposed a method to learn a robust SVM assuming that label flips can occur in training data. Their method is based on a modified *loss function* which explicitly accounts for label noise (i.e., they adopt a different definition of the slack variables in Problem 2). However, the corresponding optimization problem is no longer convex and thus they provided an approximate solution technique based on gradient descent. Xu et al. (2006) proposed a robust learning algorithm for SVMs to mitigate the

effect of outliers in training data. Similarly to the previous case, this method is also based on a different definition of the loss function, which yields a non-convex optimization problem, approximately solved through a convex relaxation.

From a theoretical standpoint, robustness was studied in the context of both classical statistics and machine learning. The robust statistics approach (Huber, 1981; Hampel et al., 1986; Maronna et al., 2006) has studied general properties of statistical estimators under the change of the underlying distributions. A well-known instrument of such analysis is the so-called *influence function*. The robustness issues of margin-based learning methods have been studied by Christmann and Steinwart (2004). In particular, they studied the behavior of SVM-like algorithms under small perturbations of training data and proved that under some conditions, the influence function of SVMs can be bounded.

3. Label Noise Robust SVMs

In this section we introduce our approach, *Label Noise robust SVMs* (LN-robust SVMs), to improve SVMs' robustness to label noise in training data. We point out that, with respect to previous works, this approach does not affect the computational complexity of the standard SVM learning algorithm, as it only yields a simple kernel matrix correction.

Label noise can be explicitly modelled by assuming that the labels in the training set $\{\mathbf{x}_i, y_i\}_{i=1}^n \in \mathcal{X} \times \{-1, +1\}$ can be flipped. To this end, we first introduce a set of random variables $\varepsilon_i \in \{0, 1\}$, $i = 1, \ldots, n$, which represent whether the corresponding label y_i is flipped (1) or not (0). Accordingly, we then replace y_i with $y_i' = y_i(1 - 2\varepsilon_i)$ such that if $\varepsilon_i = 1$, $y_i' = -y_i$ (label flip), while $y_i' = y_i$ otherwise.

In the dual SVM problem (Problem 3) the class labels solely affect the matrix $\mathbf{Q} = \mathbf{K} \circ yy^\top$. In particular, taking into account label noise, we can write its elements as

$$Q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)(1 - 2\varepsilon_i)(1 - 2\varepsilon_j)\,. \tag{4}$$

Note that, in the absence of noise $\varepsilon_i = 0$, $i = 1, \ldots, n$, and, thus, the elements of $\mathbf{Q}$ are simply $Q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, as in the standard SVM formulation.

If we assume that every label is independently flipped with the same probability, then ε_i, $i = 1, \ldots, n$, are n i.i.d. Boolean random variables, whose mean μ is simply the probability of $\varepsilon_i = 1$, and whose variance is $\sigma^2 = \mu(1 - \mu)$. Within this assumption, we can compute the expected value of $\mathbf{Q}$ from Eq. 4, which is given by

$$\mathbb{E}_\varepsilon[Q_{ij}] = \begin{cases} y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)(1 - 4\sigma^2), & \text{if } i \neq j, \\ y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), & \text{otherwise.} \end{cases} \tag{5}$$

Now, we can use the expected value of $\mathbf{Q}$ (which is still a positive semi-definite kernel matrix) to solve the SVM problem. This should reasonably improve the robustness of the learned SVM to label flip noise. The proposed method only yields a kernel matrix correction (Eq. 5), and does not modify the standard SVM problem. However, it is an heuristic method and it is thus not guaranteed to fulfill any optimality criterion (e.g., being optimal under the considered noise model).

The solution is symmetric with respect to $\mu = 0.5$, i.e., the α values obtained for $\mu = \mu^*$ and $\mu = 1 - \mu^*$ are the same, and are exactly the same as the standard SVM solution when

μ is either 0 or 1 (as the corresponding kernel correction is zero). Moreover, the equations of $\mathbf{w}$ and b obtained by solving the standard SVM problem have to be multiplied by $1-2\mu$ (as we take their expectations over label noise). Thus, when $\mu > 0.5$, $\mathbf{w}$ and b are multiplied by a negative factor. This represents the fact that more than half of the training points are assumed to have a wrong label, and, thus, the decision regions are inverted. For instance, when $\mu = 1$, the solution is exactly given by $-\mathbf{w}^*$ and $-b^*$ (being $\mathbf{w}^*$ and b^* the standard SVM solution): we are in fact assuming that all samples are wrongly labelled in the training set, and, consequently, the hyperplane obtained by the standard SVM results rotated by 180°. Note lastly that, if $\mu = 0.5$, $\mathbf{w} = \mathbf{0}$. This is a degenerate case in which labels in training data are assumed to be completely random, so the SVM is not able to determine, on average, any meaningful decision hyperplane.

3.1. Dual problem and α equalization

We are now in a position to better analyze the change induced by the kernel correction of Eq. 5 in the SVM dual problem. Indeed, the dual problem can be re-written as

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^\top (\mathbf{Q} \circ \mathbf{M})\alpha - \mathbf{1}^\top \alpha \\ \text{s. t.} \quad & 0 \leq \alpha_i \leq C,\ i = 1, \ldots, n, \\ & \sum_{i=1}^{n} \alpha_i y_i = 0, \end{aligned} \tag{6}$$

where the elements M_{ij} of $\mathbf{M}$ are given by

$$M_{ij} = \begin{cases} 1, & \text{if } i = j \\ 1 - S, & \text{otherwise,} \end{cases} \tag{7}$$

where we use $S = 4\sigma^2$ to simplify notation. The matrix $\mathbf{M}$ can be further decomposed as $\mathbf{M} = (1-S)\mathbf{1}_{n\times n} + S\mathbb{I}_{n\times n}$, where $\mathbf{1}_{n\times n}$ is a $n \times n$ matrix whose elements are all ones, and $\mathbb{I}_{n\times n}$ is the $n \times n$ identity matrix. Substituting this decomposition of $\mathbf{M}$ into Problem 6, adding and subtracting $S\sum_{i=1}^{n}\alpha_i$ from the Lagrangian, and dividing it by $1-S$, yields the following (equivalent) dual problem:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^\top \mathbf{Q}\alpha - \mathbf{1}^\top \alpha + \frac{S}{1-S}\left[\frac{1}{2}\alpha^\top (\mathbf{Q} \circ \mathbb{I}_{n\times n})\alpha - \mathbf{1}^\top \alpha\right] \\ \text{s. t.} \quad & 0 \leq \alpha_i \leq C,\ i = 1, \ldots, n, \\ & \sum_{i=1}^{n} \alpha_i y_i = 0, \end{aligned} \tag{8}$$

where the only difference with the standard SVM formulation is given by an additional term weighted by $\frac{S}{1-S}$. This reveals some interesting insights about the effect of the proposed kernel correction. First, note that as μ increases from 0 to 0.5, $\frac{S}{1-S}$ approaches infinity, namely, the α values are only determined by minimizing the latter term in Problem 8. Second, this term does not depend on the class labels, as it only involves α and the diagonal of $\mathbf{Q}$ (which is indeed equal to the diagonal of $\mathbf{K}$). The above observations highlight that, as

μ tends to 0.5, our solution will rely more on the diagonal of the kernel matrix, as it is the only component which is not affected by label noise. Note also that this is not equivalent to increase the diagonal of the kernel matrix, as commonly done for improving numerical stability. The effect of the proposed correction is instead to reduce the impact of the out-of-diagonal elements of the kernel matrix, which are the only ones affected by label noise. Last, the term weighted by $\frac{S}{1-S}$ in Problem 8 can be rewritten as

$$\frac{1}{2}\sum_{i=1}^{n}\alpha_i^2 K(\mathbf{x}_i,\mathbf{x}_i) - \sum_{i=1}^{n}\alpha_i\,. \tag{9}$$

Although this term does not have an immediate relationship to the variance of the α values, it has some similar properties. In particular, when we solve Problem 8 for some $S > 0$, we are, in fact, penalizing solutions which either increase α_i^2, $i = 1,\ldots,n$, or decrease the average value of α (w.r.t. the standard SVM solution). In other words, increasing S effectively acts to reduce the variance of the α values. This provides an intuitive explanation of our approach, which however requires further theoretical investigation. In particular, we observe that when the variance of the α values is reduced, every training sample is more likely to be a support vector. Thus, the proposed kernel correction functions as a further regularization term. That is, the LN-robust SVM *equalizes* the contribution of the training points to the solution (its hyperplane incorporates a higher number of support vectors). This intuitively reduces the influence of potential outlying samples in the training set. This concept becomes more clear when one considers the interpretation of SVMs in terms of forces and torques (see, e.g., Burges, 1998), in which the solution to the SVM optimization problem corresponds to a mechanical (translational and rotational) equilibrium. From this viewpoint, each α represents the force applied by each support vector toward the hyperplane. By spreading the α values over more points, we are effectively distributing the overall force onto a higher number of points, so that every point contributes a lower force. This intuitively leads to an improved stability of the hyperplane, with respect to changes in the set of its support vectors. This effect is quantitatively evaluated in the toy example of Sect. 5.

4. Attack strategies

We now show how the proposed kernel correction strategy behaves in the presence of potential label flipping attacks. We adopt two different strategies for contaminating the training set through label flipping: *random* and *adversarial* label flips. In both cases, we bound the adversary's effort by assuming that she can only flip the labels of a given percentage of training samples.

Random Label Flips. In the first case, we randomly select a number of samples from the training data (chosen according to a fixed percentage of data that can be manipulated by the adversary), and flip their labels. This can be regarded as a non-adversarial kind of noise, since it is not dependent on the given classifier.

Adversarial Label Flips. In the second case, instead, given a number of allowed label flips, the adversary aims to find the combination of label flips which maximizes the classification error on the *untainted* testing data. However, the problem of finding the optimal (worst-case) combination of label flips to attack the SVM learning algorithm is not trivial,

and thus we resort to a heuristic approach which has shown to be quite effective on our set of experiments (see Sect. 6). The idea behind the *adversarial label flip* attack is first to flip labels of samples with non-uniform probabilities, depending on how well they are classified by the SVM learned on the untainted training set; and, second, to repeat this process a number of times, eventually retaining the label flips which maximally decreased performance. In particular, we increase the probability of flipping labels of samples which are classified with very high confidence (i.e., non-support vectors), and decrease the probability of flipping labels of support vectors and error vectors (inversely proportional to their α value). The reason is that the former (mainly, the non-support vectors) are more likely to become support vectors or error vectors when the SVM is learned on the tainted training set, and, consequently, the decision hyperplane will be closer to them. This will reflect a considerable change in the SVM solution, and, potentially, in its classification accuracy. Furthermore, the labels of samples in different classes can be flipped in a correlated way, to force the hyperplane to rotate as much as possible. To this end, one can draw a random hyperplane $\mathbf{w}^{\text{rnd}}$, b^{rnd} in feature space, and further increase the probability of flipping the label of a positive sample $\mathbf{x}^+$ (respectively, a negative one $\mathbf{x}^-$), if $\left(\mathbf{w}^{\text{rnd}}\right)^\top \mathbf{x}^+ + b^{\text{rnd}} > 0$ ($\left(\mathbf{w}^{\text{rnd}}\right)^\top \mathbf{x}^- + b^{\text{rnd}} < 0$). We implemented the above described attack as Algorithm 4, using two weighting parameters β_1 and β_2, set to 0.1 (based on some preliminary experimental observations, we found that these values achieved good results). A simple example of application of this attack strategy is reported in the next section.

5. Toy example

We present here a simple toy example to demonstrate the adversarial label-flipping attack, and how the kernel correction proposed in Sect. 3 can effectively counteract both random and adversarial label flips. We generate a two-dimensional data set of 100 samples, where samples of class $z \in \{-1, +1\}$ are drawn from a Normal distribution with mean $[z, 0]^\top$ and (diagonal) covariance matrix equal to $\frac{1}{2}\mathbb{I}$. An SVM with linear kernel is learned on this (untainted) training set, as depicted in Fig. 1 (first plot from left in the top row). Then, we flip labels of 10 samples using the adversarial label flip attack described in the previous section. Note from Fig. 1 (second to fourth plot from left in the top row, label flips are highlighted with green circles) that: (1) the adversarial label flips mainly affect samples which are farther away from the untainted SVM decision boundary, and (2) the correlation imposed between label flips of samples of different classes induces a substantial change in the tainted SVM decision boundary (second plot from left in the top row). Besides this, note how the SVMs learned using $\mu = 0.1$ and $\mu = 0.5$ [1] (third and fourth plot in the top row) are able to compensate for the adversarial label flips, although not completely.

To better understand of this behavior and confirm the correctness of the observations in Sect. 3.1, we also plot the α values of standard SVMs and of the proposed LN-robust SVMs against the scores (i.e., the distances from the hyperplane) assigned by each SVM to each training sample (see Fig. 1, bottom row). The mean and variance of the α values for each SVM are also reported. As expected, the variance of the α values of the LN-robust SVMs decreases with respect to the standard SVMs, and decreases more as μ approaches

1. From now on, with $\mu = 0.5$ we will implicitly assume $\mu = 0.5 - \epsilon$, with $\epsilon > 0$ but small enough (e.g., 0.001), so that $\mathbf{w}$ does not degenerate to $\mathbf{0}$.

Algorithm 1 Adversarial label flip attack.

Input: the untainted training data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$, the regularization parameter C (and the kernel's parameters, if any), the number of label flips L, the number of repetitions R, and the weighting parameters β_1 and β_2.

Output: the tainted labels $y'_1, \ldots, y'_n$.

1: $(\alpha, b) \leftarrow$ train an SVM on $\mathcal{D}$

2: **for** $i = 1, \ldots, n$, **do** $s_i \leftarrow y_i[\sum_{j=1}^n y_j \alpha_j K(x_i, x_j) + b]$, **end for**

3: normalize scores $(s_1, \ldots, s_n)$ in $[0, 1]$, dividing by $\max(s_1, \ldots, s_n)$

4: $(\alpha^{\rm rnd}, b^{\rm rnd}) \leftarrow$ generate a random SVM (draw $n+1$ numbers from a uniform distribution)

5: **for** $i = 1, \ldots, n$, **do** $q_i \leftarrow y_i[\sum_{j=1}^n y_j \alpha_j^{\rm rnd} K(x_i, x_j) + b^{\rm rnd}]$, **end for**

6: normalize scores $(q_1, \ldots, q_n)$ in $[0, 1]$, dividing by $\max(q_1, \ldots, q_n)$

7: **for** $i = 1, \ldots, n$, **do** $v_i \leftarrow \alpha_i / C - \beta_1 s_i - \beta_2 q_i$, **end for**

8: $(k_1, \ldots, k_n) \leftarrow$ sort $(v_1, \ldots, v_n)$ in ascending order, and return the corresponding indexes

9: $(y'_1, \ldots, y'_n) \leftarrow (y_1, \ldots, y_n)$

10: **for** $i = 1, \ldots, L$, $y'_{k_i} = -y_{k_i}$, **end for**

11: train an SVM on $\{\mathbf{x}_i, y'_i\}_{i=1}^n$

12: estimate its training error on $\mathcal{D}$

13: repeat R times from point 4, and return the set of labels $y'_1, \ldots, y'_n$ which yielded the maximum training error.

14: **return** $y'_1, \ldots, y'_n$

0.5. This confirms that the solution of the LN-robust SVM is expected to be less sparse than the standard SVM, and thus, less sensitive to outliers in training data.

Before concluding this section, we show that the proposed LN-robust SVM can be effective against *random label flips* as well. To this aim, we conduct a simple artificial experiment similar to the previous case. We consider SVMs with linear kernel, and each class to be normally distributed with mean $[z, 0, \ldots, 0]^\top$ and (diagonal) covariance matrix equal to $\frac{1}{2}\mathbb{I}$. However, this time we consider 300 features and 400 training samples, since we need a higher features to samples ratio for the random label flip attack to be effective. Note that the optimal (Bayes) classifier in this case is simply given by $\mathbf{w} = [1, 0, \ldots, 0]$, and $b = 0$. We vary the percentage of random label flips in the training set up to 40%, and plot the corresponding testing accuracy (evaluated on a separate untainted test set of $1,000$ samples). The results are averaged over 5 repetitions, and reported in Fig. 2, for four different values of the regularization parameter C: $0.1, 1, 10, 100$. Note how, in this case (and for all values of C) the LN-robust SVM is able to significantly outperform the standard SVM (in particular when $\mu = 0.5$), and that, surprisingly, this does not cause a decrease of the classification accuracy attained on the untainted data set (i.e., when the percentage of flipped labels is zero). Notably, this highlights that there need not necessarily be a trade-off between accuracy on untainted data and robustness to attacks.

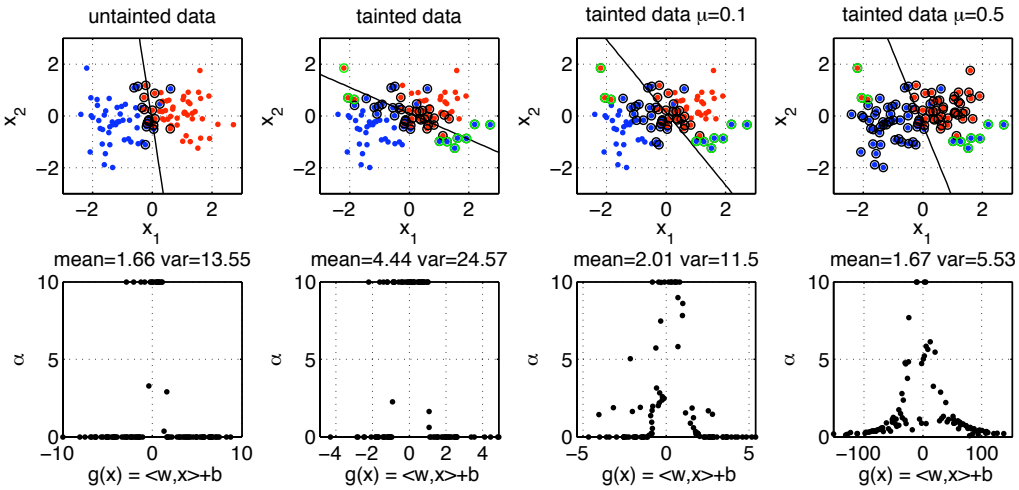

Figure 1: (*top row*) Standard SVM trained on untainted and tainted data (first and second plot, respectively), robust SVM with $\mu = 0.1$ and $\mu = 0.5$ trained on untainted data (third and fourth plot, respectively); (*bottom row*) α values of each training sample versus its distance to the hyperplane $g(\mathbf{x})$, corresponding to the SVM and data shown in the above plots. Mean and variance of the α values are also reported. Data is tainted by performing 10 *adversarial label flips*, highlighted with green circles. The support vectors of each SVM are circled in black.

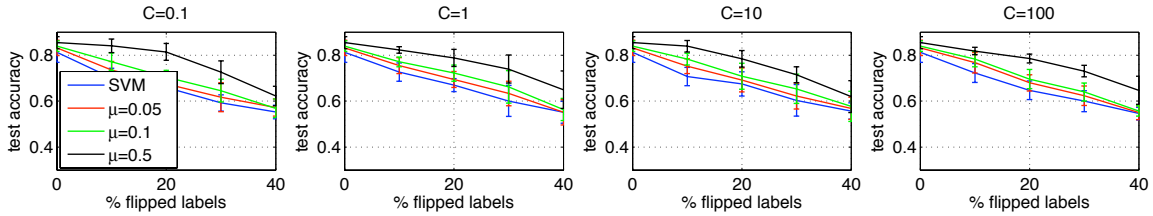

Figure 2: Classification accuracy on artificial normal data (untainted) for the SVM and the LN-robust SVMs with linear kernel. LN-robust SVMs were trained with $\mu = 0.05$, $\mu = 0.1$ and $\mu = 0.5$. Results are shown for different percentages of *random* label flips in training data, and different values of the regularization parameter C.

6. Experiments

We report experimental results to empirically validate the soundness of the proposed approach. We consider a number of real data sets, and compare the LN-robust SVM to the standard SVM learning algorithm, either with linear or radial basis function (RBF) kernels, under *random* and *adversarial* label flips. We report the classification accuracy attained

by each classifier on untainted test data, as a function of the percentage of label flips in training data; the more gracefully the performance decreases, the more robust the classifier is.

Data sets. We downloaded 7 two-class data sets from LibSVM and UCI repositories, with feature values already scaled in $[-1, +1]$ [2] (Chang and Lin, 2001; Asuncion and Newman, 2007). Their characteristics are summarized in Table 1. Within these experiments, every data set was randomly split 5 times into different training (TR) and testing (TS) set pairs, respectively with 60% and 40% of samples each. The results were then averaged over these 5 trials.

Name	# samples	# features
Breast-cancer	683	10
Australian	690	14
Diabetes	768	8
Fourclass	862	2
Heart	270	13
Ionosphere	351	34
Sonar	208	60

Table 1: Main characteristics of the data sets

Setup. We considered four different values of the regularization parameter C, namely, $C = 0.1,\ 1,\ 10,\ 100$, for both the linear and RBF kernels. Moreover, when the RBF kernel was used, for any fixed C value, the parameter γ was selected among the values $\{0.01, 0.1, 1, 10, 100\}$ by performing a 5-fold cross validation on training data. In each plot, we report the performance of the standard SVM and of three LN-robust SVMs respectively trained with $\mu = 0.05,\ 0.1,\ 0.5$. When the *adversarial* label flip attack is considered, the labels to be flipped are determined using the standard SVM solution. However, we also noted that there was not any relevant difference in the results when the same attack was computed using the solutions of the LN-robust SVMs.

Results. Results for SVMs with the linear kernel against *adversarial* label flips and *random* label flips are respectively reported in Fig. 3 and 4. Due to lack of space, we omit results for the RBF kernel, which exhibit similar behavior and lead to similar conclusions. First, note that, as expected, adversarial label flips generally decrease the performance with fewer flips than random flipping. As the standard SVM is naturally somewhat robust to random label noise (see Fig. 4), the resilience of the LN-robust SVM is most pronounced for adversarial label flips although it also generally outperforms the standard SVM with random flips. Second, for low values of C (i.e., 0.1 and 1), the LN-robust SVM does not generally improve the performance over the standard SVM; indeed, sometimes it is even less robust to label flips (see, e.g., the "diabetes" dataset under adversarial label flips, Fig. 3). On the other hand, the LN-robust SVM can significantly improve the robustness when C is relatively high (see, e.g., "australian", "breast-cancer", "fourclass", "heart" and "ionosphere"). The reason for this is that when the regularization parameter C is high, the SVM tends to find a hard-margin solution which is clearly more sensitive to label noise, and, in this case,

2. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

training an LN-robust SVM can successfully mitigate this problem (whereas, it may not be helpful if the regularization parameter is low). Third, notice that the LN-robust SVM with $\mu = 0.5$ is able to significantly outperform the other classifiers on several data sets, particularly when it achieves similar performances to the other classifiers on the untainted training data (i.e., for 0% flipped labels). On the contrary, it may significantly worsen the performance if its classification accuracy on untainted data is already low. This suggests two conclusions: first, there is a trade-off between accuracy on untainted data and robustness to noise; and, second, the parameter μ in our method can not be tuned with standard performance evaluation techniques like cross-validation. The latter observation may seem a major issue for the application of the proposed method. However, this problem exists for any method which assumes some form of non-stationarity in the underlying data distribution, as standard performance evaluation clearly relies on stationarity. In other words, to estimate any parameter which depends on a *potential* change in the underlying data distribution, standard performance evaluation techniques can not be adopted, and one needs to exploit different criteria. For instance, for the LN-robust SVM, one may try to increase μ as long as the estimated performance is still acceptable (or μ has already reached 0.5). This can be done, for example, by still using cross validation but defining a proper utility function based on both the classification accuracy and a term proportional to μ, which can be interpreted as the robustness against *potential* label flips.

7. Discussions and open issues

In this section, we further discuss two open issues mentioned throughout the paper which may be of interest for investigation from a theoretical perspective.

In Sect. 5, we observed that sometimes increasing μ can improve the performances over the standard SVM even if the training data is not tainted. This is somewhat surprising as one would expect that a trade-off between accuracy (on untainted data) and robustness (under contamination) normally exists, as witnessed by robust statistics (Huber, 1981; Hampel et al., 1986; Maronna et al., 2006). This raises the interesting question of whether the proposed kernel correction acts as a further regularization term. We provide a partial interpretation to this in Sect. 3.1 in discussing our method's tendency to equalize the α values among training points. However, an in-depth investigation of this aspect may reveal some interesting properties related to the general notion of robustness. To this end, it might be possible to exploit the work by Xu et al. (2009), where the authors found a clear relation between robustness to noise and regularization (as mentioned in Sect. 2.2).

Another open issue is related to the choice of the parameter μ of our method, which, as pointed out in the previous section, can not be done through standard performance evaluation techniques (e.g., cross-validation). This generally raises the issue of how to assess performance of classification algorithms in adversarial environments, or when the underlying data distribution is subject to *change*. However, this is a relevant open issue which can not be addressed thoroughly in this paper.

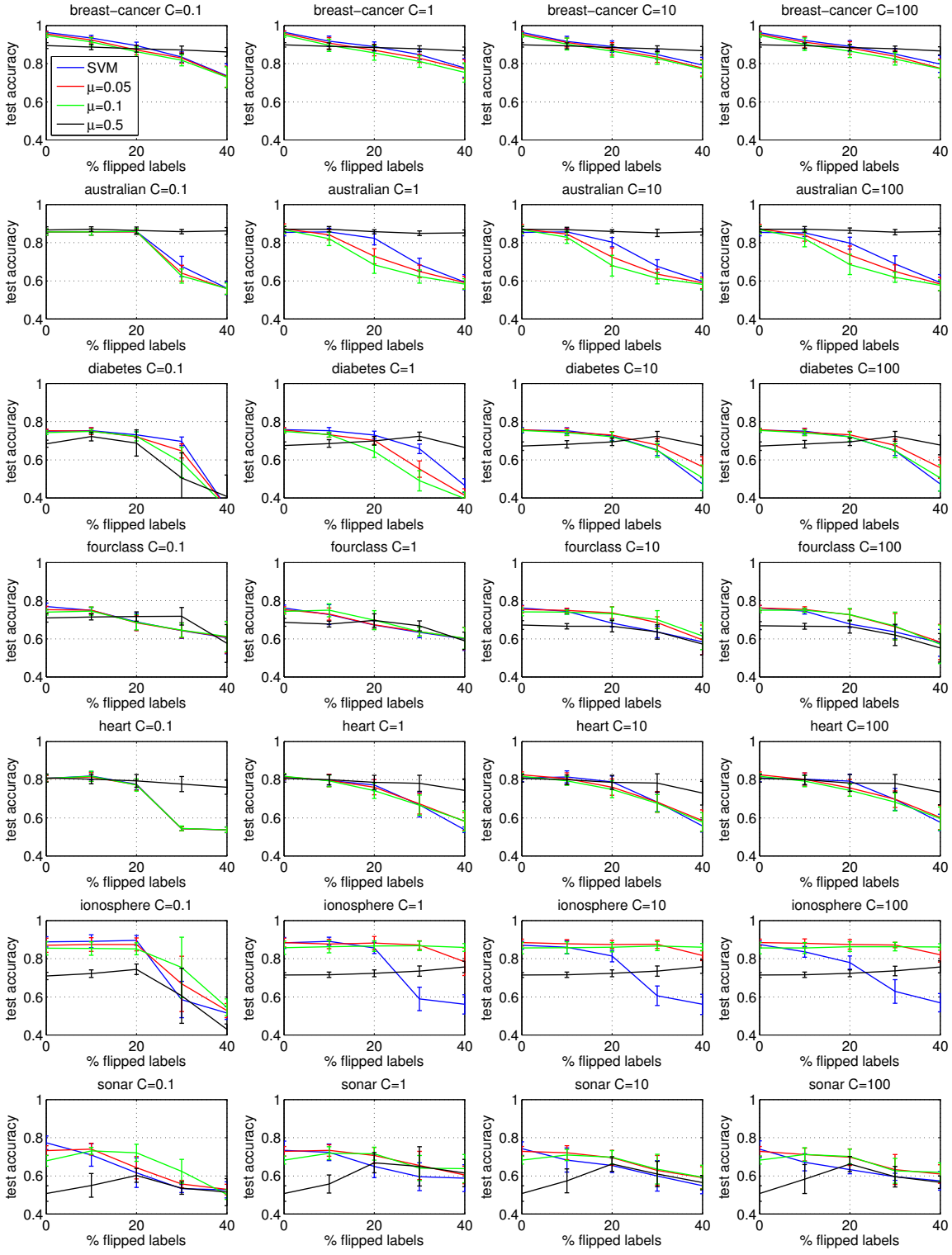

Figure 3: Adversarial label flip attack against SVM and LN-robust SVMs (with $\mu = 0.05, 0.1, 0.5$) with linear kernel, for different values of C, and percentage of noise.

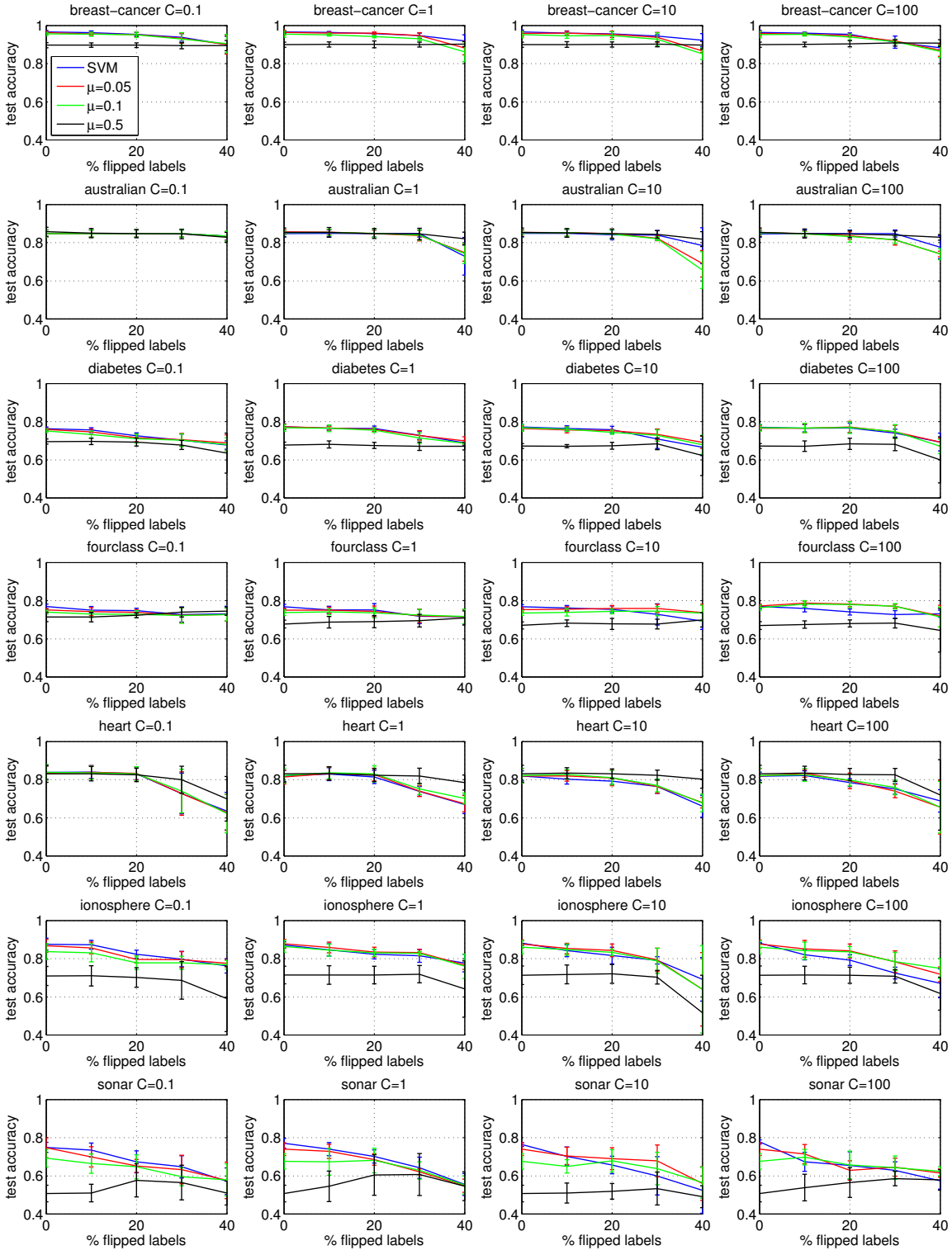

Figure 4: Random label flip attack against SVM and LN-robust SVMs (with $\mu = 0.05, 0.1, 0.5$) with linear kernel, for different values of C, and percentage of noise.

8. Conclusions and future works

Throughout this paper, we have investigated the robustness of SVMs under adversarial label noise and proposed a method to improve it based on a simple kernel matrix correction. We showed the effectiveness of the proposed approach on several artificial and real data sets. We empirically observed that our method leads to equalization of α values in SVMs, which intuitively hedges the influence of individual points and leads to a more robust estimator. Our experimental results support the common observation that robustness exhibits a trade-off with classification accuracy.

A current limitation of our method is the need to a-priori agree on a potential degree of label contamination. While some ad-hoc heuristics are conceivable for setting the corresponding parameter μ in practice, the investigation of theoretically sound methods for selecting an optimal "robustness level" would be an interesting issue for future work, as well as considering our method in real adversarial problems like spam filtering and intrusion detection, and comparing it with other SVM implementations which are meant to be robust against label noise (e.g., Stempfel and Ralaivola, 2009).

Acknowledgments

The authors wish to acknowledge the Alexander von Humboldt Foundation and the Heisenberg Fellowship of the Deutsche Forschungsgemeinschaft (DFG) for providing financial support to carry out this research. This work was also partly supported by a grant awarded to B. Biggio by Regione Autonoma della Sardegna, PO Sardegna FSE 2007-2013, L.R. 7/2007 "Promotion of the scientific research and technological innovation in Sardinia". The opinions expressed in this paper are solely those of the authors and do not necessarily reflect the opinions of any sponsor.

References

A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. http://www.ics.uci.edu/~mlearn/MLRepository.html.

M. Barreno, B. Nelson, A. Joseph, and J. Tygar. The security of machine learning. *Machine Learning*, 81:121–148, 2010.

J. Bi and T. Zhang. Support vector classification with input data uncertainty. In *Advances in Neural Information Processing Systems 17*, 2004.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed., 2007.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2:121–167, 1998.

C.-C. Chang and C.-J. Lin. LibSVM: a library for support vector machines, 2001. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

A. Christmann and I. Steinwart. On robust properties of convex risk minimization methods for pattern recognition. *J. of Machine Learning Research*, 5:1007–1034, 2004.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma. Adversarial classification. In *10th ACM Int'l Conf. on Knowl. Disc. and Data Min.*, pp. 99–108, 2004.

O. Dekel, O. Shamir, and L. Xiao. Learning to classify with missing and corrupted features. *Machine Learning*, 81:149–178, 2010.

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

P. Fogla, M. Sharif, R. Perdisci, O. Kolesnikov, and W. Lee. Polymorphic blending attacks. In *Proc. of the 15th Conf. on USENIX Sec. Symp.*, 2006.

A. Globerson and S. T. Roweis. Nightmare at test time: robust learning by feature deletion. In William W. Cohen and Andrew Moore, eds, *ICML*, vol. 148 of *ACM Int'l Conf. Proc. Series*, pp. 353–360, 2006.

F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Probability and Mathematical Statistics. John Wiley and Sons, 1986.

P. Huber. *Robust Statistics*. Probability and Mathematical Statistics. John Wiley and Sons, 1981.

A. Kolcz and C. H. Teo. Feature weighting for improved classifier robustness. In *6th Conf. on Email and Anti-Spam (CEAS)*, 2009.

D. Lowd and C. Meek. Adversarial learning. In *Proc. of the 11th ACM Int'l Conf. on Knowl. Disc. and Data Min.*, pp. 641–647, 2005.

R. A. Maronna, R. D. Martin, and V. J. Yohai. *Robust Statistics: Theory and Methods*. Probability and Mathematical Statistics. John Wiley and Sons, 2006.

G. Stempfel and L. Ralaivola. Learning SVMs from sloppily labeled data. In *Proc. of the 19th Int'l Conf. on Artificial Neural Netw.*, ICANN '09, pp. 884–893, 2009.

C. H. Teo, A. Globerson, S. Roweis, and A. Smola. Convex learning with invariances. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, eds, *Advances in Neural Information Processing Systems 20*, pp. 1489–1496, 2008.

V. Vapnik. *The nature of statistical learning theory*. Springer-Verlag NY, 1995.

H. Xu, C. Caramanis, and S. Mannor. Robustness and regularization of support vector machines. *J. of Machine Learning Research*, 10:1485–1510, 2009.

L. Xu, K. Crammer, and D. Schuurmans. Robust support vector machine training via convex outlier ablation. In *Proc. of the 21st National Conf. on Artificial Intell.*, vol. 1, pp. 536–542. AAAI Press, 2006.