Adversarial Label Flips Attack on Support Vector Machines

Han Xiao and **Huang Xiao** and **Claudia Eckert**[1]

Abstract. To develop a robust classification algorithm in the adversarial setting, it is important to understand the adversary's strategy. We address the problem of label flips attack where an adversary contaminates the training set through flipping labels. By analyzing the objective of the adversary, we formulate an optimization framework for finding the label flips that maximize the classification error. An algorithm for attacking support vector machines is derived. Experiments demonstrate that the accuracy of classifiers is significantly degraded under the attack.

1 INTRODUCTION

We focus on the binary classification for security applications, in which a *defender* attempts to separate *instances* into malicious and benign classes. The threat is that the *adversary* will manipulate instances to mislead the decision of a classifier [7]. According to the capability of the adversary, attacks may be either *exploratory* in that they exploit the blind spot of a classifier but do not affect training, or they may be *causative* in that they subvert the learning process by controlling the training data [1]. For example, in an exploratory attack, the adversary disguises the spam by adding unrelated words to evade the spam filter [9, 10, 14]. In a causative attack, the adversary flags every legitimate mail as spam while the defender is gathering the training data. Consequently, the spam filter trained on such data is likely to cause a false alarm and may block all legitimate mails [12, 11].

The causative attack has recently attracted growing interest from the scientific community due to its long-lasting impact on learning algorithms. In general, if one attempt to harness human resources for training models, then the training data is in danger of contamination. Specifically, the adversary can carry out the causative attack either by introducing *feature noise* or *label noise* to the training data. Different types of feature noise have been extensively studied in several literature [4, 6, 9, 11]. However, little is known on how adversarial label noise is induced. Most of previous work either assume that labels are erased at random [3], or they restrict the underlying distribution of label noise to certain families without considering the attack strategy from the adversary's perspective [5, 8]. Recently, a label flips strategy based on heuristics is proposed to attack support vector machines (SVMs) [2].

This paper formalizes the problem of *adversarial label flips attack* in the supervised learning setting, where the adversary contaminates the training data through flipping labels. More exactly, the adversary aims to find a combination of label flips under a given *budget* so that a classifier trained on such data will have maximal classification error. Motivated by Tikhonov regularization, we present an optimization framework for solving this problem. We then devise an algorithm for attacking support vector machine, which can be efficiently solved as two minimization problems. Experiments demonstrate that our attack maximally degrades the accuracy of SVMs with different kernels.

While solving problems for adversaries may seem counterproductive, we believe that investigating the strategy of the adversary and the vulnerability of the defender is the only way to develop a robust learning algorithm in the future. The rest of this paper is organized as follows. The problem of adversarial label flips is described in Section 2. A framework for finding the near-optimal label flips is presented in Section 3. The algorithm for attacking SVMs is derived in Section 4, followed by experimental results on both synthetic and real-world data in Section 5. Section 6 provides conclusions and discussions.

2 PROBLEM STATEMENT

In the supervised classification problem, we have a training set of n instances $S := \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathcal{X},\, y_i \in \mathcal{Y}\}_{i=1}^{n}$, with the *input space* $\mathcal{X}$ and the *label space* $\mathcal{Y} := \{-1, 1\}$. Given a *hypothesis space* $\mathcal{H}$ and a *loss function* V, the goal is to find a classification hypothesis $f_S \in \mathcal{H}$ by solving Tikhonov regularization problem

$$f_S := \arg\min_f \gamma \sum_{i=1}^{n} V\left(y_i, f(\mathbf{x}_i)\right) + \|f\|_{\mathcal{H}}^2, \tag{1}$$

where f_S denotes the classifier trained on S, and γ is a fixed positive parameter for quantifying the trade off. Remark that the first term in (1) reflects the empirical loss of f on S, and the second term reflects the generalization ability of f. Given an instance $\mathbf{x} \in \mathcal{X}$, the classification decision is made according to the sign of $f_S(\mathbf{x})$.

To express the label flips, we first introduce a set of variables $z_i \in \{0, 1\}, i = 1, \ldots, n$. Then replace y_i with $y'_i := y_i(1-2z_i)$ so that if $z_i = 1$ then the label is flipped $y'_i = -y_i$, otherwise $y'_i = y_i$. Denote $S' := \{(\mathbf{x}_i, y'_i)\}_{i=1}^{n}$ the *tainted* training set, which shares the same instances as S but with some flipped labels. The adversary constructs S' in such a way that the resulting $f_{S'}$ yields maximal loss on some test set T. Thus, the problem of finding the near-optimal label flips can be formulated as

$$\max_{\mathbf{z}} \quad \sum_{(\mathbf{x},y)\in T} V\left(y, f_{S'}(\mathbf{x})\right), \tag{2}$$

$$\text{s.t.} \quad f_{S'} \in \arg\min_f \gamma \sum_{i=1}^{n} V\left(y'_i, f(\mathbf{x}_i)\right) + \|f\|_{\mathcal{H}}^2, \tag{3}$$

$$\sum_{i=1}^{n} c_i z_i \le C, \quad z_i \in \{0, 1\}, \tag{4}$$

[1] Institute of Informatics, Technische Universität München, Germany. {**xiaoh**,xiaohu,claudia.eckert}@in.tum.de

where $c_i \in \mathbb{R}_{0+}$ is the cost (or risk) of flipping label y_i from the adversary's viewpoint. Constraint (4) limits the total adversarial cost of label flips to C. Unfortunately, the above *bilevel* optimization problem is intrinsically hard due to the conflict and the interaction between (2) and (3). The conflict arises from the fact that for a given training set the defender learns a classifier with minimal empirical loss and good generalization ability, whereas the adversary expects that the classifier has maximal loss and poor generalization ability. That is, the beneficial outcome in one of them is associated with a detrimental outcome in another. Moreover, since any single flipped label may lead to a change to the classifier, the greedy strategy that flips labels based merely on the current classifier is ineffective. Essentially, the adversary has to evaluate each combination of label flips and selects the one that deteriorates the classifier the most.

As solving even the simplest linear bilevel problem is strong $\mathcal{NP}$-hard [13] and an exhaustive search on all combinations of flips is prohibitive, we resort to a relaxed formulation of finding the near-optimal label flips. In particular, we assume that the adversary only maximizes the empirical loss of the classifier on the original training set, yet indulges the defender in maximizing the generalization ability of the classifier. To obtain a set of label flips that jointly deteriorates the classifier's performance to the greatest extent, the adversary must foresee the reaction of the defender to the flipped labels. With these considerations in mind, we relax the original bilevel problem and present a loss minimization framework in the next section.

3 LABEL FLIPS ATTACK FRAMEWORK

Let A and B be two sets of labeled instances, we first define an auxiliary loss function

$$g(B, f_A) := \gamma \sum_{(\mathbf{x},y)\in B} V\left(y, f_A(\mathbf{x})\right) + \|f_A\|^2_{\mathcal{H}}, \tag{5}$$

where f_A denotes the classifier trained on A. Note that the first term in (5) reflects the empirical loss incurred by f_A over the set B, which differs from (1).

To maximally degrade the classifier's performance, we select S' so that it has maximal loss under the original classifier f_S but yields minimal loss under the tainted classifier $f_{S'}$. The intuition is as follows: the adversary shifts the classification hypothesis so that the "terribly" mislabeled instances in S' asserted by the original classifier are now identified as "perfectly" labeled instances by the tainted classifier. With this strategy, the adversary can proactively cause the defender to produce a classifier whose loss is low on S' but high on S, which in turn has high loss on the test set. Formally, this idea can be represented as

$$\begin{aligned} \min_{\mathbf{z}} \quad & g(S', f_{S'}) - g(S', f_S), \\ \text{s.t.} \quad & \sum_{i=1}^{n} c_i z_i \le C, \quad z_i \in \{0,1\}. \end{aligned} \tag{6}$$

Remark that given *any* training set the defender *always* finds the optimal classifier by solving Tikhonov regularization problem. Thus, the first term in (6) reflects the defender's destined action on the training set S'. The second term quantifies the empirical loss on S' using the classifier f_S trained on the original set S, which represents the adversary's strategy of selecting instances with high loss.

We further refine the objective function and constraints of (6) for the algorithmic convenience. Denote U the expanded representation of S so that each instance in S is duplicated with a flipped label. Formally, the set $U := \{(\mathbf{x}_i, y_i)\}_{i=1}^{2n}$ is constructed as follows

$$\begin{aligned} (\mathbf{x}_i, y_i) \in S, &\quad i = 1, \dots, n, \\ \mathbf{x}_i := \mathbf{x}_{i-n}, &\quad i = n+1, \dots, 2n, \\ y_i := -y_{i-n} &\quad i = n+1, \dots, 2n. \end{aligned}$$

We introduce an indicator variable $q_i \in \{0,1\}, i = 1, \dots, 2n$ for each element in U, where $q_i = 1$ denotes that $(\mathbf{x}_i, y_i) \in S'$, and $q_i = 0$ denotes that it is not. Replace S' by U and substitute (5) into (6), we can rewrite the near-optimal label flips problem as

$$\begin{aligned} \min_{\mathbf{q}, f} \quad & \gamma \sum_{i=1}^{2n} q_i \left[V\left(y_i, f(\mathbf{x}_i)\right) - V\left(y_i, f_S(\mathbf{x}_i)\right) \right] + \|f\|^2_{\mathcal{H}}, \\ \text{s.t.} \quad & \sum_{i=n+1}^{2n} c_i q_i \le C, \\ & q_i + q_{i+n} = 1, \quad i = 1, \dots, n, \\ & q_i \in \{0,1\}, \quad i = 1, \dots, 2n. \end{aligned} \tag{7}$$

We ignore $\|f_S\|^2_{\mathcal{H}}$ as it is a constant with respect to the optimization variables. Indicator variables $q_{n+1}, \dots, q_{2n}$ correspond to $z_1, \dots, z_n$ in the previous bilevel formulations, respectively. The constraint $q_i + q_{i+n} = 1$ reflects that only one label can be chosen for the instance $\mathbf{x}_i$. Due to the acquiescence on the defender's behavior of maximizing the generalization ability of the tainted classifier, the conflicting objectives of the defender and the adversary are now incorporated into one minimization problem. Given a training set we can employ the above framework to compute the set of label flips that will jointly degrade the classifier's accuracy without exceeding a specified budget. Recall that SVMs can be considered as a special case of Tikhonov regularization, it is straightforward to develop an attack on SVMs subject to this framework, as we shall see in the next section.

4 ATTACK ON SVM

SVMs project the original training instances from the input space $\mathcal{X}$ to the *feature space* $\mathcal{F}$ by $\Phi : \mathcal{X} \to \mathcal{F}$. In general, SVMs trained on S has the form

$$f_S(\mathbf{x}) := \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b,$$

where K is a *Mercer Kernel* which satisfies the property $K(\mathbf{x}, \mathbf{x}_i) = \Phi(\mathbf{x})^\top \Phi(\mathbf{x}_i)$ and $b \in \mathbb{R}$ denotes the bias. The classifier can be also rewritten as

$$f_S(\mathbf{x}) := \mathbf{w}^\top \mathbf{x} + b,$$

where $\mathbf{w} := \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i)$ and $\mathbf{w} \in \mathcal{F}$. Thus, the classification boundary of a SVM is a hyperplane in $\mathcal{F}$ with normal vector $\mathbf{w}$. Given the *hinge loss* function $V\left(y, f(\mathbf{x})\right) := \max(0, 1 - y f(\mathbf{x}))$, Tikhonov regularization for SVMs is a constrained quadratic programming (QP) problem

$$\begin{aligned} \min_{\mathbf{w}, \boldsymbol{\xi}, b} \quad & \gamma \sum_{i=1}^{n} \xi_i + \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \ge 1 - \xi_i, \quad \xi_i \ge 0, \quad i = 1, \dots, n, \end{aligned} \tag{8}$$

where ξ_i represents the hinge loss of $(\mathbf{x}_i, y_i)$ resulting from the classifier f_S. Denote $\epsilon_i := \max(0, 1 - y_i f_{S'}(\mathbf{x}_i))$ the hinge loss of

$(\mathbf{x}_i, y_i)$ resulting from the tainted classifier $f_{S'}$. By plugging (8) into (7), we have

$$\begin{aligned}
\min_{\mathbf{q},\mathbf{w},\boldsymbol{\epsilon},b} \quad & \gamma \sum_{i=1}^{2n} q_i(\epsilon_i - \xi_i) + \frac{1}{2}\|\mathbf{w}\|^2 \qquad (9)\\
\text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0, \quad i = 1, \ldots, 2n,\\
& \sum_{i=n+1}^{2n} c_i q_i \leq C,\\
& q_i + q_{i+n} = 1, \quad i = 1, \ldots, n,\\
& q_i \in \{0, 1\}, \quad i = 1, \ldots, 2n.
\end{aligned}$$

Observe that (9) involves an integer programming problem which is in general $\mathcal{NP}$-hard. Therefore, we first relax it into a continuous optimization problem by allowing all q_i to take values between $[0, 1]$. Then we decompose (9) into two sub-problems and devise an iterative approach to minimize them alternatively. On the one hand, by fixing $\mathbf{q}$, the minimization over $\mathbf{w}, \boldsymbol{\epsilon}, b$ is reduced to the following QP problem

$$\begin{aligned}
\min_{\mathbf{w},\boldsymbol{\epsilon},b} \quad & \gamma \sum_{i=1}^{2n} q_i \epsilon_i + \frac{1}{2}\|\mathbf{w}\|^2 \qquad (10)\\
\text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0, \quad i = 1, \ldots, 2n.
\end{aligned}$$

On the other hand, by fixing $\mathbf{w}, b$ and using the computed $\boldsymbol{\epsilon}$ the minimization over $\mathbf{q}$ can be described as a linear programming (LP) as follows

$$\begin{aligned}
\min_{\mathbf{q}} \quad & \gamma \sum_{i=1}^{2n} q_i(\epsilon_i - \xi_i) \qquad (11)\\
\text{s.t.} \quad & \sum_{i=n+1}^{2n} c_i q_i \leq C,\\
& q_i + q_{i+n} = 1, \quad i = 1, \ldots, n,\\
& 0 \leq q_i \leq 1, \quad i = 1, \ldots, 2n.
\end{aligned}$$

It is easy to see that by minimizing (10) and (11) the objective function (9) decreases monotonically. Note that ξ_i can be computed beforehand, the algorithm can be implemented efficiently with off-the-shelf QP and LP solvers. After the algorithm converges, we greedily select the largest subset of $\{q_{n+1}, \ldots, q_{2n}\}$ meeting the given budget and flip the corresponding labels. The complete procedure is summarized in Algorithm 1, which we denote as ALFA.

5 EXPERIMENTAL RESULTS

We demonstrate the label flips attack on SVMs with linear kernel and radial basis function (RBF) kernel using two sets of experiments. First, we employ some two-dimensional synthetic data to visualize the decision boundaries of SVMs under the label flips. The second set of experiments is conducted on ten real-world data sets, where we concentrate the influence of label flips on SVMs with respect to different budgets. In all experiments, the proposed ALFA is compared with the following three label flip strategies

- Uniform random flip: instances are uniformly chosen at random from the training set and their labels are flipped. This can be regarded as introducing label noise to the training set from the non-adversarial perspective.
- Nearest-first flip: instances that have small distances to the decision hyperplane in the feature space are first flipped. This corresponds to a thoughtless labeler who erroneously labels instances that are difficult to be distinguished.
- Furthest-first flip: instances that have large distances to the decision hyperplane in the feature space are first flipped. In this way, we simulate a malicious labeler who deliberately gives wrong labels on instances that are easy to be distinguished.

Algorithm 1: Adversarial Label Flips Attack on SVMs (ALFA)

```
Input : original training set $S$, adversarial cost $c_1, \ldots, c_n$,
        budget $C$, parameter $\gamma$
Output: tainted training set $S'$ with flipped labels
Find $f_S$ by solving (8) on $S$;                              /* QP */
foreach $(\mathbf{x}_i, y_i) \in U$ do
    $\xi_i \leftarrow \max(0, 1 - y_i f_S(\mathbf{x}_i))$;
    $\epsilon_i \leftarrow 0$;
repeat
    Find $q_1, \ldots, q_{2n}$ by solving (11);                /* LP */
    Find $\epsilon_1, \ldots, \epsilon_{2n}$ by solving (10);  /* QP */
until convergence;
$L \leftarrow$ Sort($[q_{n+1}, \ldots, q_{2n}]$, "desc");
/* $L$ is an array of sorted indices */
for $i \leftarrow 1$ to $n$ do $y'_i \leftarrow y_i$;
$j \leftarrow 1$;
while $\sum_{i=1}^{j} q_{L[i]} \leq C$ do
    $y'_{L[j]-n} \leftarrow -y_{L[j]-n}$;                      /* Flip label */
    $j \leftarrow j + 1$;
return $S' \leftarrow \{(\mathbf{x}_i, y'_i)\}_{i=1}^{n}$;
```

The adversarial cost is set as $c_i := 1$ for $i = 1, \ldots, n$. Thus, given a budget C one can flip at most $\min(\lfloor C \rfloor, n)$ labels. Experiments are conducted as follows. First, we randomly select the same number of instances from two classes and construct the training set and the test set, respectively. Second, the training set is tainted by performing different flip strategies. Third, we train SVMs (with $\gamma := 1$) on the original training set and four tainted training sets. Finally, the classification error of each SVM is measured on the test set, respectively. As our test set is balanced, the worst performance of a classifier is with 50% error rate, which corresponds to the random guess. Hence, an error rate around 50% indicates an effective attack strategy on SVMs.

In the experiments, the convergence of ALFA typically occurs in $5 \sim 10$ iterations. On a training set with 300 instances, our MATLAB implementation[2] without special code-level optimization takes about 3 seconds for computing the near-optimal label flips[3].

5.1 Synthetic Examples

We generate linear and parabolic patterns in two dimensional space for this experiment. From each pattern, we select 100 instances as the training set and 800 instances as the test set. Let $C := 20$, decision

[2] MATLAB implementation and more experimental results are available at http://home.in.tum.de/~xiaoh

[3] We tried an exhaustive search to find the groundtruth optimal label flips. For example, To obtain the optimal 20 label flips out of 300 training instances, our program has to check over 7×10^{30} combinations. Due to the extremely slow progress, we terminated the program after one month running on a 12-cores workstation.

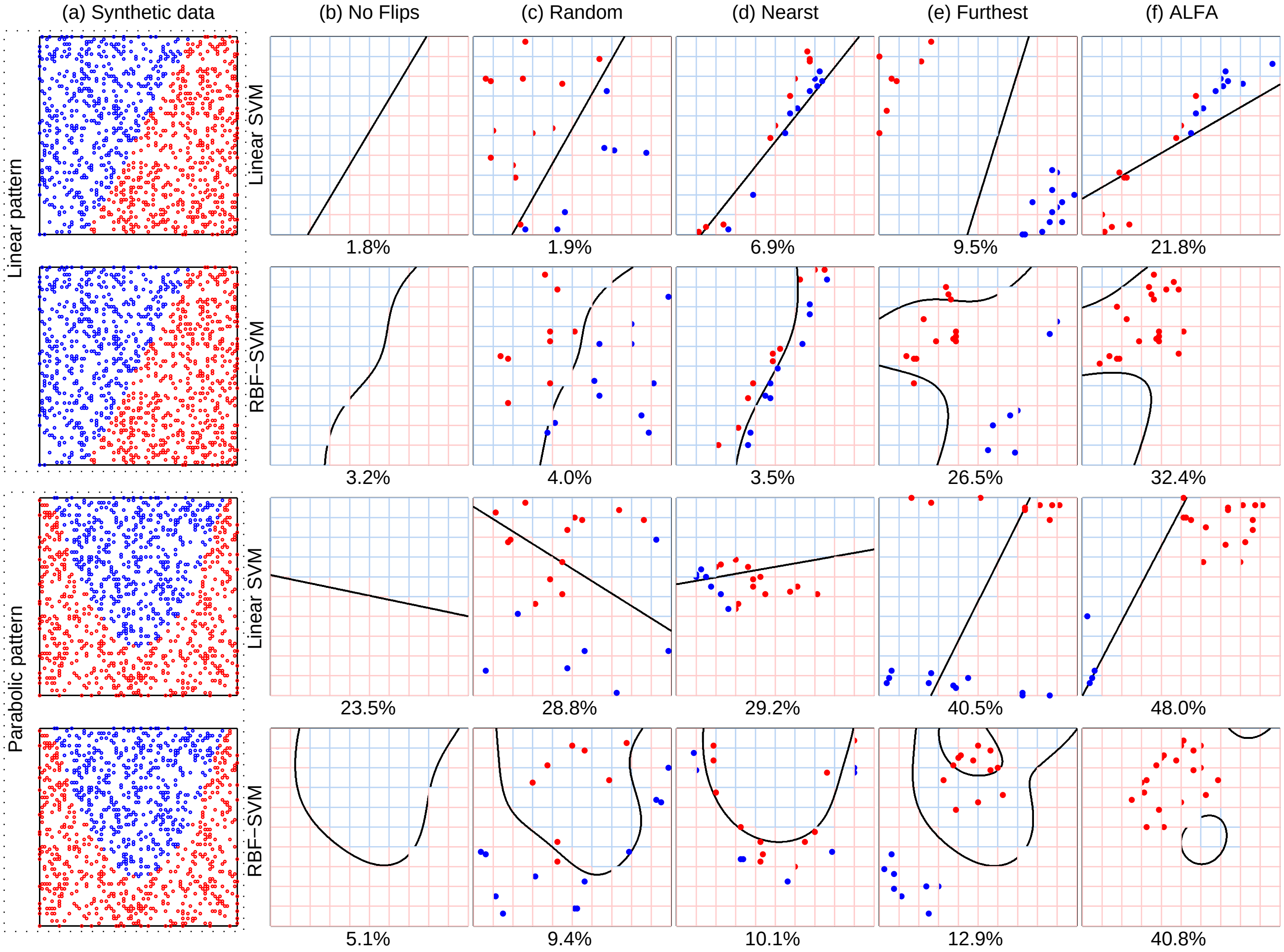


Figure 1. Decision boundaries of SVMs under different flip strategies. The first and second rows illustrate results on the linear pattern, the third and fourth rows illustrate results on the parabolic pattern. For each strategy, the number of flipped labels is fixed to 20 (i.e. 20% of the training data). Each point represents an instance. Labels are denoted in red and blue. In each plot, decision regions of SVMs are shaded in different colors. Only flipped instances in the training set are highlighted. The percentage under each plot indicates the error rate of SVM measured on the test set, respectively. **(a)** The synthetic data generated for the experiment. **(b)** Decision boundaries of SVMs trained on the original training set without label flips. **(c)** Decision boundaries of SVMs under random label flips. **(d)** Decision boundaries of SVMs under nearest-first flip strategy. **(e)** Decision boundaries of SVMs under furthest-first flip strategy. **(f)** Decision boundaries of SVMs under ALFA.

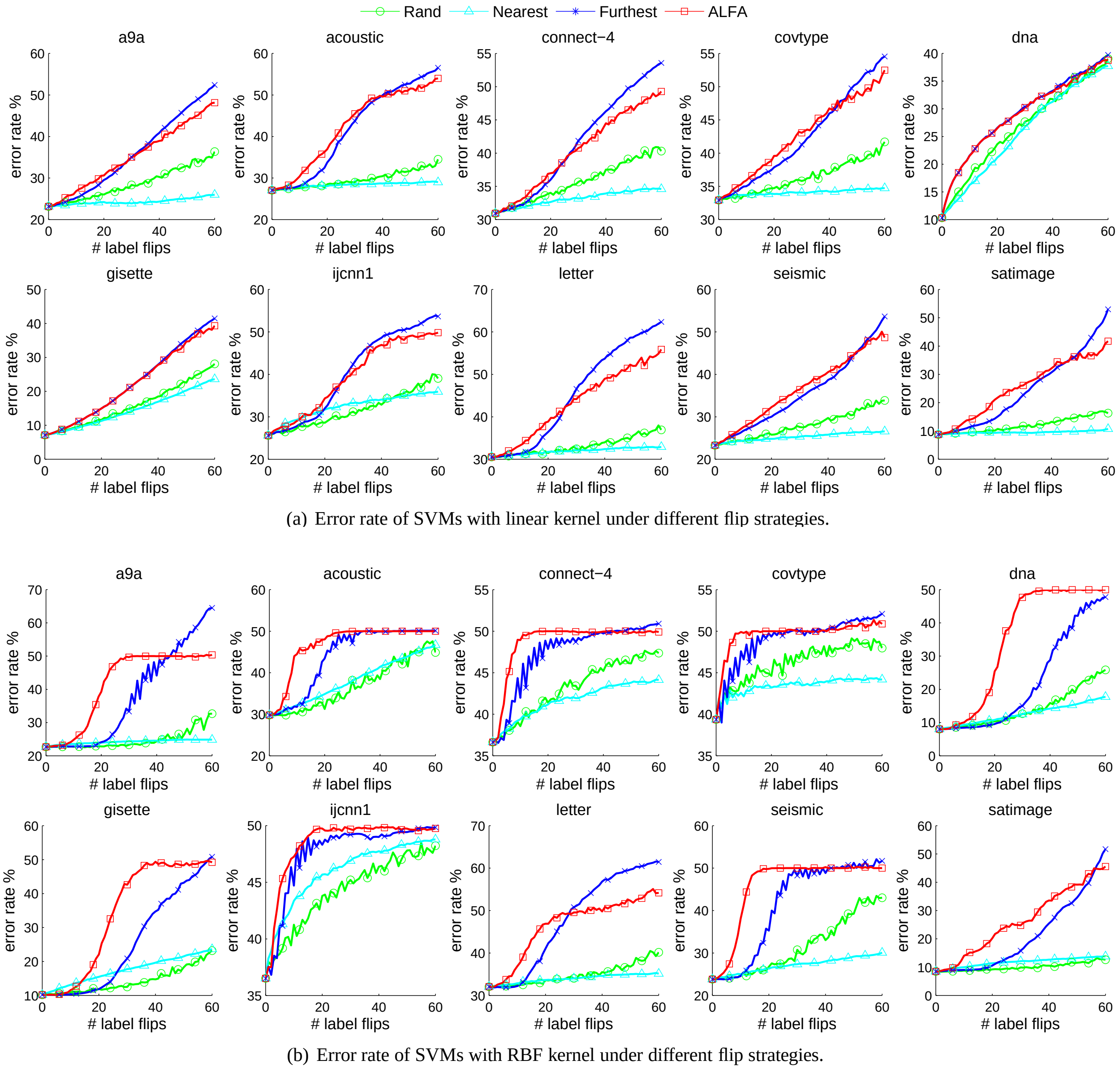
boundaries of SVMs under different flip strategies are illustrated in Fig. 1.

By comparing Fig. 1(b) with Fig. 1(f), one can clearly observe the dramatic changes on decision boundaries of SVMs under ALFA. For instance, the original decision plane of linear SVM on the parabolic pattern is almost tilted by 90 degrees under ALFA (see the $3^{\rm rd}$ row of Fig. 1). Moreover, when ALFA is applied to SVMs with RBF kernel, the error rate increases from 3.2% to 32.4% on the linear pattern and 5.1% to 40.8% on the parabolic pattern. Not surprisingly, the nearest-first strategy is least effective due to the tolerance nature of soft-margin SVMs. While the furthest-first strategy increases the classification error as well, it is less compelling than ALFA. Further note that the performance of SVMs is quite stable under the uniform random label noise and the error rate hardly changes with 20 flipped labels, as shown in Fig. 1(c). This implies that previous robust learning algorithms based on the assumption of random label noise may be too optimistic as they underestimate the adversary's impact on the classifier's performance.

5.2 On Real-World Data

We continue the investigation of different flip strategies using 10 real-world data sets, which are downloaded from LIBSVM website. For each data set, we randomly select 200 instances as the training set and 800 instances as the test set. As in practice the adversary usually controls only a small portion of the training data, we demonstrate the effectiveness of label flips with respect to different budgets, especially with low budget.

Figure 2 depicts the error rate of SVMs up to 60 label flips (i.e. $C := 1, \ldots, 60$). As expected, the error rate of SVMs increases with the growth of label flips. While SVMs sometimes show the resilience to the random label noise, the error rate significantly increases under ALFA and the furthest-first strategy due to their adversarial nature. The advantage of ALFA is most significant when SVMs are trained with RBF kernel. On many data sets, by flipping only 20 labels (i.e. 10% of training data) with ALFA the error rate of RBF-SVM rises to 50%, which is turned into the random guess. Moreover, we remark that ALFA is more cost-effective than the furthest-first strategy es-



(a) Error rate of SVMs with linear kernel under different flip strategies.

(b) Error rate of SVMs with RBF kernel under different flip strategies.

Figure 2. Error rate of SVMs as a function of the number flipped labels. Within each experiment, the training set consists of 200 instances (100 for each class) selected randomly. The adversary can flip at most 60 labels (i.e. 30% of the training data). The classification error is measured on 800 test instances with balanced labels. Results are averaged over 60 repetitions. Note that 50% error rate corresponds to the random guess.

pecially with small flips. When the number of flipped labels is large, ALFA keeps trapping SVMs with worst performance at 50% error rate. On the contrary, the furthest-first strategy increases the error rate over 50% (see Fig. 2(b) `a9a`,`connect-4`,`letter`), which in fact regains the predictive power of SVMs. This behavior is due to the fact that our framework captures the classifier's reaction to flipped labels, whereas the furthest-first strategy merely considers the information about the current classifier.

From the perspective of a cost-averse adversary, it is also interesting to know the required budget for turning a SVM into a random guess. Table 1 shows the required percentage of label flips when the tainted SVM reaches 50% error rate on the test set. First of all, observe that the required percentage of label flips greatly depends on data sets, or how training instances are distributed in the feature space. Moreover, comparing with the linear kernel it is easier to taint SVMs with RBF kernel. This is because by mapping instances to the infinite dimensional feature space, instances are more sparsely distributed. Hence, flipping a label will result a significant change on the separating hyperplane. Furthermore, in both cases ALFA flips less labels than other strategies. For the linear kernel the required percentage of label flips is roughly stable with respect to the size of the training set. That is, the required flips rises linearly when the size of training set increases. On the contrary, for RBF kernel the required percentage increases as the training set becomes larger.

Finally, we adapt ALFA to attack the label noise robust SVM (LN-SVM) based on a simple kernel matrix correction [2]. Our experiment indicates that, although LN-SVM shows resilience to the random noisy labels, it still greatly suffers from ALFA.

Table 1. The percentage of flipped labels when a SVM reaches 50% error rate. Experiment is conducted on ten data sets with $100, 200$ and 300 training instances, respectively. The classification error is measured on the randomly selected test set with 800 instances. From the adversary's viewpoint, smaller percentage value indicates a more cost-effective flip strategy as it requires lower budget. For each data set, the most effective strategy is highlighted with the boldface. Results are averaged over 60 repetitions.

	100				200				300			
Data sets	Rand.	Near.	Furt.	ALFA	Rand.	Near.	Furt.	ALFA	Rand.	Near.	Furt.	ALFA
SVM with linear kernel												
a9a	41.9	70.4	**29.5**	31.5	43.7	72.2	**27.1**	29.8	44.5	72.9	**26.7**	29.9
acoustic	38.5	77.6	19.2	**17.1**	41.5	77.4	18.8	**17.3**	42.5	76.6	18.8	**17.4**
connect-4	38.2	67.7	**27.7**	29.1	40.1	73.7	**24.4**	27.5	42.2	77.3	**21.4**	25.2
covtype	32.1	73.7	25.0	**23.8**	37.0	74.4	24.6	**22.6**	36.9	75.1	23.9	**21.7**
dna	**43.4**	47.6	50.7	47.8	**42.5**	51.6	45.8	44.2	43.5	54.6	**42.6**	43.2
gisette	47.7	56.6	43.7	**43.6**	47.0	61.8	**37.9**	37.9	47.6	63.8	**35.6**	35.6
ijcnn1	33.9	62.6	26.5	**25.4**	37.9	72.7	21.5	**20.8**	38.2	76.4	19.7	**17.6**
letter	36.7	80.6	**18.2**	19.0	40.2	82.6	**17.1**	18.6	41.5	82.1	**17.4**	19.1
seismic	38.7	73.8	26.3	**25.5**	40.7	71.3	**28.3**	28.7	41.3	70.7	28.8	**28.1**
satimage	44.5	70.5	**30.0**	32.2	45.4	70.3	29.8	**25.5**	46.4	69.2	30.6	**22.3**
SVM with RBF kernel												
a9a	21.6	65.3	12.8	**7.7**	31.5	74.9	18.8	**12.0**	36.1	76.1	20.4	**14.1**
acoustic	6.3	14.7	4.1	**2.9**	16.3	36.8	10.2	**7.1**	22.6	52.7	13.7	**7.8**
connect-4	7.2	33.8	3.7	**2.8**	18.5	68.8	8.7	**5.3**	25.2	76.2	12.3	**6.8**
covtype	2.5	13.2	1.8	**1.4**	6.6	55.8	4.3	**2.2**	11.6	71.2	7.3	**3.9**
dna	27.6	53.6	20.8	**11.6**	40.9	63.7	31.6	**17.0**	46.7	66.5	32.6	**19.2**
gisette	29.4	68.9	23.4	**14.1**	38.7	70.8	28.4	**17.8**	43.4	69.2	29.0	**19.3**
ijcnn1	8.1	27.2	4.2	**3.5**	19.4	41.0	13.6	**8.4**	25.0	40.3	20.4	**10.4**
letter	22.6	78.0	11.7	**8.0**	31.0	84.4	14.1	**10.9**	35.3	84.5	14.2	**11.9**
seismic	11.0	33.4	6.4	**4.3**	24.0	64.4	13.5	**7.4**	29.3	69.0	16.4	**9.6**
satimage	39.1	69.2	25.5	**23.7**	41.8	68.8	28.7	**22.3**	43.4	67.8	30.3	**23.3**

6 CONCLUSIONS AND DISCUSSIONS

If we hope to develop a robust learning algorithm under adversarial conditions, it is incumbent on us to understand the adversary's strategy. Throughout this paper, we have investigated the problem of adversarial label flips in the supervised learning setting, where an attacker contaminates the training data through flipping labels. We present an optimization framework for the adversary to find the near-optimal label flips that maximally degrades the classifier's performance. The framework simultaneously models the adversary's attempt and the defender's reaction in a loss minimization problem. Based on this framework, we develop an algorithm for attacking SVMs. Experimental results demonstrate the effectiveness of the proposed attack on both synthetic and real-world data set.

Comparing with the random label noise, the adversarial label noise has been shown to be more influential to the classifier's performance. Thus, the proposed framework can be used as a baseline for evaluating the robustness of a learning algorithm under the noisy condition. The framework can be also extended to the active learning and online learning settings, where labels are usually committed by massive annotators with various motivations. Another relevant scenario is the crowdsourcing platform (e.g. Amazon's Mechanical Turk), where the labeled data can be obtained quickly from crowds of human workers. In such settings, the adversarial label noise is inevitable due to the limitation of quality control mechanisms. As a part of future work, it would be interesting to formulate this learning problem as a n-player hybrid game, which contains both cooperative and non-cooperative players. By categorizing players into coalitions and modeling the worst-case behavior of each coalition, one may develop an algorithm that learns from good labelers yet shows resilience to malicious labelers.

REFERENCES

[1] M. Barreno, B. Nelson, A.D. Joseph, and JD Tygar, 'The security of machine learning', *Machine Learning*, **81**(2), 121–148, (2010).

[2] B. Biggio, B. Nelson, and B. Laskov, 'Support vector machines under adversarial label noise', in *Proc. of 3rd ACML*, pp. 97–112, (2011).

[3] O. Chapelle, B. Schölkopf, A. Zien, et al., *Semi-supervised learning*, MIT Press, 2006.

[4] O. Dekel and O. Shamir, 'Learning to classify with missing and corrupted features', in *Proc. of 25th ICML*, pp. 216–223, (2008).

[5] O. Dekel and O. Shamir, 'Good learners for evil teachers', in *Proc. of 26th ICML*, pp. 233–240. ACM, (2009).

[6] A. Globerson and S. Roweis, 'Nightmare at test time: robust learning by feature deletion', in *Proc. of 23rd ICML*, pp. 353–360. ACM, (2006).

[7] M. Kearns and M. Li, 'Learning in the presence of malicious errors', in *Proc. of 20th STOC*, pp. 267–280. ACM, (1988).

[8] A.R. Klivans, P.M. Long, and R.A. Servedio, 'Learning halfspaces with malicious noise', *JMLR*, **10**, 2715–2740, (2009).

[9] D. Lowd and C. Meek, 'Adversarial learning', in *Proc. of 11th SIGKDD*, pp. 641–647. ACM, (2005).

[10] D. Lowd and C. Meek, 'Good word attacks on statistical spam filters', in *Proc. of 2nd Conference on Email and Anti-Spam*, pp. 125–132, (2005).

[11] B. Nelson, M. Barreno, F.J. Chi, A.D. Joseph, B.I.P. Rubinstein, U. Saini, C. Sutton, JD Tygar, and K. Xia, 'Exploiting machine learning to subvert your spam filter', in *Proc. of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, p. 7, (2008).

[12] J. Newsome, B. Karp, and D. Song, 'Paragraph: Thwarting signature learning by training maliciously', in *Recent Advances in Intrusion Detection*, pp. 81–105. Springer, (2006).

[13] L. Vicente, G. Savard, and J. Júdice, 'Descent approaches for quadratic bilevel programming', *Journal of Optimization Theory and Applications*, **81**(2), 379–399, (1994).

[14] Han Xiao, T. Stibor, and C. Eckert, 'Evasion attack of multi-class linear classifiers', in *Proc. of 16th PAKDD*, pp. 207–218, (2012).