Evasion Attack of Multi-Class Linear Classifiers

Han Xiao[1,2], Thomas Stibor[2], and Claudia Eckert[2]

[1] CeDoSIA of TUM Graduate School
[2] Chair for IT Security
Technische Universität München, Germany
{xiaoh,stibor,claudia.eckert}@in.tum.de

Abstract. Machine learning has yield significant advances in decision-making for complex systems, but are they robust against adversarial attacks? We generalize the evasion attack problem to the multi-class linear classifiers, and present an efficient algorithm for approximating the optimal disguised instance. Experiments on real-world data demonstrate the effectiveness of our method.

1 Introduction

Researchers and engineers of information security have successfully deployed systems using machine learning and data mining for detecting suspicious activities, filtering spam, recognizing threats, etc. [2, 12]. These systems typically contain a classifier that flags certain instances as malicious based on a set of features. Unfortunately, evaded malicious instances that fail to be detected are inevitable for any known classifier. To make matters worse, there is evidence showing that adversaries have investigated several approaches to evade the classifier by disguising malicious instance as normal instances. For example, spammers can add unrelated words, sentences or even paragraphs to the junk mail for avoiding detection of the spam filter [11]. Furthermore, spammers can embed the text message in an image. By adding varied background and distorting the image, the generated junk message can be difficult for OCR systems to identify but easy for humans to interpret [7]. As a reaction to adversarial attempts, authors of [5] employed a cost-sensitive game theoretic approach to preemptively adapt the decision boundary of a classifier by computing the adversary's optimal strategy. Moreover, several improved spam filters that are more effective in adversarial environments have been proposed [7, 3].

The ongoing war between adversaries and classifiers pressures machine learning researchers to reconsider the vulnerability of classifier in adversarial environments. The problem of evasion attack is posed and a query algorithm for evading linear classifiers is presented [10]. Given a malicious instance, the goal of the adversary is finding a disguised instance with the minimal cost to deceive the classifier. Recently, the evasion problem has been extended to the binary convex-inducing classifiers [13].

We continue investigate the vulnerability of classifiers to the evasion attack and generalize this problem to the family of multi-class linear classifiers; e.g. linear support vector machines [4, 6, 9]. Multi-class linear classifiers have become one of the most promising learning techniques for large sparse data with a huge number of instances and features. We propose an adversarial query algorithm for searching minimal-cost

disguised instances. We believe that revealing a scar on the multi-class classifier is the only way to fix it in the future. The contributions of this paper are:

1. We generalize the problem of evasion attack to the multi-class linear classifier, where the instance space is divided into multiple convex sets.
2. We prove that effective evasion attack based on the linear probing is feasible under certain assumption of the adversarial cost. A description of the vulnerability of multi-class linear classifiers is presented.
3. We propose a query algorithm for disguising an adversarial instance as any other classes with minimal cost. The experiment on two real-world data set shows the effectiveness of our algorithm.

2 Problem Setup

Let $\mathcal{X} = \{(x_1, \ldots, x_D) \in \mathbb{R}^D \mid L \leq x_d \leq U \text{ for all } d\}$ be the *feature space*. Each component of an *instance* $\mathbf{x} \in \mathcal{X}$ is a *feature* bounded by L and U which we denote as x_d. A basis vector of the form $(0, \ldots, 0, 1, 0, \ldots, 0)$ with a 1 only at the d^{th} feature terms $\boldsymbol{\delta}_d$. We assume that the feature space representation is known to the adversary, thus the adversary can query any point in $\mathcal{X}$.

2.1 Multi-Class Linear Classifier

The target classifier f is a mapping from feature space $\mathcal{X}$ to its response space $\mathcal{K}$; i.e. $f : \mathcal{X} \to \mathcal{K}$. We restrict our attention to *multi-class linear classifiers* and use $\mathcal{K} = \{1, \ldots, K\}, K \geq 2$ so that

$$f(\mathbf{x}) = \underset{k}{\operatorname{argmax}}\ \mathbf{w}_k \mathbf{x}^{\mathrm{T}} + b_k, \tag{1}$$

where $k = 1, \ldots, K$ and $\mathbf{w}_k \in \mathbb{R}^D, b_k \in \mathbb{R}$. Decision boundaries between class k and other classes are characterized by $\mathbf{w}_k$ and b_k. We assume that $\mathbf{w}_1, \ldots, \mathbf{w}_K$ are linearly independent. The classifier f partitions $\mathcal{X}$ into K sets; i.e. $\mathcal{X}_k = \{\mathbf{x} \in \mathcal{X} \mid f(\mathbf{x}) = k\}$.

2.2 Attack of Adversary

As a motivating example, consider a text classifier that categorizes incoming emails into different topics; e.g. sports, politics, lifestyle, spam, etc. An advertiser of pharmacological products is more likely to disguise the spam as lifestyle rather than politics in order to attract potential consumers while remaining inconspicuous.

We assume the adversary's attack will be against a fixed f so the learning method of decision boundaries and the training data used to establish the classifier are irrelevant. The adversary does not know any parameter of f but can observe $f(\mathbf{x})$ for any $\mathbf{x}$ by issuing a *membership query*. In fact, there are a variety of domain specific mechanisms that an adversary can employ to observe the classifier's response to a query. Moreover, the adversary is only aware of an adversarial instance $\mathbf{x}^{\mathrm{A}}$ in some class, and has no information about instances in other classes. This differs from previous work which require at least one instance in each binary class [10, 13]. In practice, $\mathbf{x}^{\mathrm{A}}$ can be seen as the most desired instance of adversary; e.g. the original spam. The adversary attempts to disguise $\mathbf{x}^{\mathrm{A}}$ so that it can be recognized as other classes.

2.3 Adversarial Cost

We assume that the adversary has the access to an *adversarial cost function* $a(\mathbf{x}, \mathbf{y}) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{0+}$. An adversarial cost function measures the distance between two instances $\mathbf{x}, \mathbf{y}$ in $\mathcal{X}$ from the adversary's prospective. We focus on a linear cost function which measures the weighted ℓ_1 distance so that

$$a(\mathbf{x}, \mathbf{y}) = \sum_{d=1}^{D} e_d |x_d - y_d|, \tag{2}$$

where $0 < e_d < \infty$ represents the cost coefficient of the adversary associates with the d^{th} feature, allowing that some features may be more important than others. In particular, given the adversarial instance $\mathbf{x}^{\text{A}}$, function $a(\mathbf{x}, \mathbf{x}^{\text{A}})$ measures different costs of using some instances as compared to others. Moreover, we use $\mathcal{B}(\mathbf{y}, C) = \{\mathbf{x} \in \mathcal{X} \mid a(\mathbf{x}, \mathbf{y}) \leq C\}$ to denote the cost ball centered at $\mathbf{y}$ with cost no more than C.

In generalizing work [10], we alter the definition of *minimal adversarial cost* (MAC). Given a fixed classifier f and an adversarial cost function a we define the MAC of class k with respect to an instance $\mathbf{y}$ to be the value

$$\text{MAC}(k, \mathbf{y}) = \min_{\mathbf{x}:\mathbf{x} \in \mathcal{X}_k} a(\mathbf{x}, \mathbf{y}), \quad k \neq f(\mathbf{y}).$$

2.4 Disguised Instances

We now introduce some instances with special adversarial cost that the adversary is interested in. First of all, instances with cost of $\text{MAC}(k, \mathbf{y})$ are termed *instances of minimal adversarial cost* (IMAC), which is formally defined as

$$\text{IMAC}(k, \mathbf{y}) = \{\mathbf{x} \in \mathcal{X}_k \mid a(\mathbf{x}, \mathbf{y}) = \text{MAC}(k, \mathbf{y}), k \neq f(\mathbf{y})\}.$$

Ideally, the adversary attempts to find $\text{IMAC}(k, \mathbf{x}^{\text{A}})$ for all $k \neq f(\mathbf{x}^{\text{A}})$. The most naive way for an adversary to find the IMAC is performing a brute-force search. That is, the adversary randomly samples points in $\mathcal{X}$ and updates the best found instance repetitively. To formulate this idea, we further extend the definition of IMAC. Assume $\widetilde{\mathcal{X}}$ is the set of adversary's sampled or observed instances so far and $\widetilde{\mathcal{X}} \subset \mathcal{X}$, we define *instance of sample minimal adversarial cost* (ISMAC) of class k with respect to an instance $\mathbf{y}$ to be the value

$$\text{ISMAC}(k, \mathbf{y}) = \operatorname*{argmin}_{\mathbf{x}:\mathbf{x} \in \widetilde{\mathcal{X}} \cap \mathcal{X}_k} a(\mathbf{x}, \mathbf{y}), \quad k \neq f(\mathbf{y}).$$

Note, that in practice the exact decision boundary is unknown to the adversary, thus finding exact value of IMAC becomes an infeasible task. Nonetheless, it is still tractable to approximate IMAC by finding ϵ-IMAC, which is defined as follows

$$\epsilon\text{-IMAC}(k, \mathbf{y}) = \{\mathbf{x} \in \mathcal{X}_k \mid a(\mathbf{x}, \mathbf{y}) \leq (1 + \epsilon) \cdot \text{MAC}(k, \mathbf{y}), k \neq f(\mathbf{y}), \epsilon > 0\}.$$

That is, every instance in ϵ-$\text{IMAC}(k, \mathbf{y})$ has the adversarial cost no more than a factor of $(1 + \epsilon)$ of the $\text{MAC}(k, \mathbf{y})$. The goal of the adversary now becomes finding ϵ-$\text{IMAC}(k, \mathbf{x}^{\text{A}})$ for all classes $k \neq f(\mathbf{x}^{\text{A}})$ while keeping ϵ as small as possible.

3 Theory of Evasion Attack

We discuss the evasion attack from a theoretical point of view. Specifically, by describing the feature space as a set of convex polytopes, we show that IMAC must be attained on the convex surface. Under a reasonable assumption of adversarial cost function, effective evasion attack can be performed by linear probing. Finally, we derive bounds for quantitatively studying the vulnerability of multi-class linear classifiers to linear probing.

Lemma 1. *Let $\mathcal{X}_k = \{\mathbf{x} \in \mathcal{X} \mid f(\mathbf{x}) = k\}$, where the classifier f is defined in (1). Then $\mathcal{X}_k$ is a closed convex polytope.*

Proof. Let $\mathbf{x}$ be a point in $\mathcal{X}_k$. As $\mathbf{x} \in \mathcal{X}$ it follows that

$$\mathbf{x}^{\mathrm{T}} \geq L \cdot \mathbb{1}_D \qquad \text{and} \qquad -\mathbf{x}^{\mathrm{T}} \geq U \cdot \mathbb{1}_D, \tag{3}$$

where $\mathbb{1}_D$ is a D-dimensional unit vector $(1, \ldots, 1)$. Moreover, since $f(\mathbf{x}) = k$, it follows that

$$\begin{pmatrix} \mathbf{w}_k - \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_k - \mathbf{w}_K \end{pmatrix} \mathbf{x}^{\mathrm{T}} \geq \begin{pmatrix} b_1 - b_k \\ \vdots \\ b_K - b_k \end{pmatrix}. \tag{4}$$

Thus, the foregoing linear inequalities define an intersection of at most $(K + 2D - 1)$ half-spaces. Denote $H_i^+ = \{\mathbf{x} \in \mathcal{X} \mid \widetilde{\mathbf{w}_i}\mathbf{x}^{\mathrm{T}} \geq \widetilde{b_i}\}$, where $1 \leq i \leq (K + 2D - 1)$. We have $\mathcal{X}_k = \bigcap_i H_i^+$, which establishes a half-space representation of convex polytope [8, 14]. □

Lemma 1 indicates that the classifier f decomposes $\mathbb{R}^D$ into K convex polytopes. Following the notations and formulations introduced in [8], we represent a hyperplane H_i as the boundary of a half-space ∂H_i^+; i.e. $H_i = \partial H_i^+ = \{\mathbf{x} \in \mathcal{X} \mid \widetilde{\mathbf{w}_i}\mathbf{x}^{\mathrm{T}} = \widetilde{b_i}\}$. Let $\mathcal{X}_k = \bigcap_{p=1}^{P_k} H_p^+$, where $\{H_1^+, \ldots, H_{P_k}^+\}$ is *irredundant*[3] to $\mathcal{X}_k$. Let $\mathcal{H}_k = \{H_1^+, \ldots, H_{P_k}^+\}$ be an irredundant set that defines $\mathcal{X}_k$, then $\mathcal{X}_k \subset \operatorname{int} \mathcal{X}$ provided that none half-space in $\mathcal{H}_k$ is defined by (3). Moreover, we define the p^{th} *facet* of $\mathcal{X}_k$ as $F_{kp} = H_p \cap \mathcal{X}_k$, and the *convex surface* of $\mathcal{X}_k$ as $\partial\mathcal{X}_k = \bigcup_{p=1}^{P_k} F_{kp}$.

Theorem 1. *Let $\mathbf{y}$ be an instance in $\mathcal{X}$ and $k \in \mathcal{K} \setminus f(\mathbf{y})$. Let $\mathbf{x}$ be an instance in IMAC$(k, \mathbf{y})$ as defined in Section 2.3. Then $\mathbf{x}$ must be attained on the convex surface $\partial\mathcal{X}_k$.*

Proof. We first show the existence of IMAC$(k, \mathbf{y})$. By Lemma 1, $\mathcal{X}_k$ defines a feasible region. Thus minimizing $a(\mathbf{x}, \mathbf{y})$ on $\mathcal{X}_k$ is a solvable problem. Secondly, $\mathcal{X}_k$ is bounded in each direction of the gradient of $a(\mathbf{x}, \mathbf{y})$, which implies that IMAC$(k, \mathbf{y})$ exists.

We now prove that $\mathbf{x}$ must lie on $\partial\mathcal{X}_k$ by contrapositive. Assume that $\mathbf{x}$ is not on $\partial\mathcal{X}_k$ thus is an interior point; i.e. $\mathbf{x} \in \operatorname{int} \mathcal{X}_k$. Let $\mathcal{B}(\mathbf{y}, C)$ denote the ball centered at $\mathbf{y}$ with cost no more than $a(\mathbf{x}, \mathbf{y})$. Due to the convexity of $\mathcal{X}_k$ and $\mathcal{B}(\mathbf{y}, C)$, we have $\operatorname{int} X_k \cap \operatorname{int} \mathcal{B}(\mathbf{y}, C) \neq \emptyset$. Therefore, there exists at least one instance in $\mathcal{X}_k$ with cost less than $a(\mathbf{x}, \mathbf{y})$, which implies that $\mathbf{x}$ is not IMAC$(k, \mathbf{y})$. □

[3] Let $\mathcal{C}$ be a convex polytope such that $\mathcal{C} = \bigcap_{i=1}^{n} H_i^+$. The family $\{H_1^+, \ldots, H_n^+\}$ is called *irredundant* to $\mathcal{C}$ provided that $\bigcap_{1 \leq j \leq n, j \neq i} H_j^+ \neq \mathcal{C}$ for each $j = 1, \ldots, n$.

Theorem 1 restricts the searching of IMAC to the convex surface. In particular, when cost coefficients are equal, e.g. $e_1 = \cdots = e_D$, we can show that searching in all axis-aligned directions gives at least one IMAC.

Theorem 2. *Let* $\mathbf{y}$ *be an instance in* $\mathcal{X}$ *such that* $\mathcal{X}_{f(\mathbf{y})} \subset \operatorname{int} \mathcal{X}$*. Let* P *be the number of facets of* $\mathcal{X}_{f(\mathbf{y})}$ *and* F_p *be the* p^{th} *facet, where* $p = \{1, \ldots, P\}$*. Let* $G_d = \{\mathbf{y} + \theta \boldsymbol{\delta}_d \,|\, \theta \in \mathbb{R}\}$*, where* $d \in \{1, \ldots, D\}$*. Let* $\mathcal{Q} = \{G_d \cap F_p \,|\, d = 1, \ldots, D, p = 1, \ldots, P\}$*, in which each element differs from* $\mathbf{y}$ *on only one dimension. If the adversarial cost function defined in (2) has equal cost coefficients, then there exists at least one* $\mathbf{x} \in \mathcal{Q}$ *such that* $\mathbf{x}$ *is IMAC*$(f(\mathbf{x}), \mathbf{y})$.

Proof. Let H_p be the hyperplane defining the p^{th} facet F_p. Consider all the points of intersection of the lines G_d with the hyperplanes H_p; i.e. $\mathcal{I} = \{G_d \cap H_p \,|\, d = 1, \ldots, D, p = 1, \ldots, P\}$. Let $\mathbf{x} = \operatorname{argmin}_{\mathbf{x} \in \mathcal{I}} a(\mathbf{x}, \mathbf{y})$. Then $\mathbf{x}$ is our desired instance.

We prove that $\mathbf{x} \in \mathcal{Q}$ by contrapositive. Suppose $\mathbf{x} \notin \mathcal{Q}$, due to the convexity of $\mathcal{X}_{f(\mathbf{y})}$, the line segment $[\mathbf{x}, \mathbf{y}]$ intersects $\partial \mathcal{X}_{f(\mathbf{y})}$ at a point on another facet. Denote this point as $\mathbf{z}$, then $\mathbf{z}$ differs from $\mathbf{y}$ on only one dimension and $a(\mathbf{z}, \mathbf{y}) < a(\mathbf{x}, \mathbf{y})$.

Next, we prove $\mathbf{x}$ is IMAC$(f(\mathbf{x}), \mathbf{y})$ by contrapositive. Let $\mathcal{B}(\mathbf{y}, C)$ denote the *regular* cost ball centered at $\mathbf{y}$ with cost no more than $a(\mathbf{x}, \mathbf{y})$. That is, each vertex of the cost ball has the same distance of C with $\mathbf{y}$. Suppose $\mathbf{x}$ is not IMAC$(f(\mathbf{x}), \mathbf{y})$, then there exists $\mathbf{z} \in \mathcal{X}_{f(\mathbf{x})} \cap \operatorname{int} \mathcal{B}(\mathbf{y}, C)$. By Theorem 1, $\mathbf{z}$ and $\mathbf{x}$ must lie on the same facet, which is defined by a hyperplane H^*. Let $\mathcal{Q}^*$ be intersection points of H^* with lines $G_1, \ldots, G_D$; i.e. $\mathcal{Q}^* = \{G_d \cap H^* \,|\, d = 1, \ldots, D\}$. Then there exists at least one point $\mathbf{v} \in \mathcal{Q}^*$ such that $\mathbf{v} \in \operatorname{int} \mathcal{B}(\mathbf{y}, C)$. Due to the regularity of $\mathcal{B}(\mathbf{y}, C)$, we have $a(\mathbf{v}, \mathbf{y}) < a(\mathbf{x}, \mathbf{y})$. □

We now define special convex sets for approximating ϵ-IMAC near the convex surface. Given $\epsilon > 0$, the interior parallel body of $\mathcal{X}_k$ is $\mathcal{P}_{-\epsilon}(k) = \{\mathbf{x} \in \mathcal{X}_k \,|\, \mathcal{B}(\mathbf{x}, \epsilon) \subseteq \mathcal{X}_k\}$ and the corresponding exterior parallel body is defined as $\mathcal{P}_{+\epsilon}(k) = \bigcup_{\mathbf{x} \in \mathcal{X}_k} \mathcal{B}(\mathbf{x}, \epsilon)$. Moreover, the interior margin of $\mathcal{X}_k$ is $\mathcal{M}_{-\epsilon}(k) = \mathcal{X}_k \setminus \mathcal{P}_{-\epsilon}(k)$ and the corresponding exterior margin is $\mathcal{M}_{+\epsilon}(k) = \mathcal{P}_{+\epsilon}(k) \setminus \mathcal{X}_k$. By relaxing the searching scope from the convex surface to a margin in the distance ϵ, Theorem 1 and Theorem 2 immediately imply the following results.

Corollary 1. *Let* $\mathbf{y}$ *be an instance in* $\mathcal{X}$ *and* $k \in \mathcal{K} \setminus f(\mathbf{y})$*. For all* $\epsilon > 0$ *such that* $M_{-\epsilon}(k) \neq \emptyset$*,* ϵ*-IMAC*$(k, \mathbf{y}) \subseteq \mathcal{M}_{-\epsilon}(k)$.

Corollary 2. *Let* $\mathbf{y}$ *be an instance in* $\mathcal{X}$ *and* ϵ *be a positive number such that* $\mathcal{P}_{+\epsilon}(f(\mathbf{y})) \subset \operatorname{int} \mathcal{X}$*. Let* P *be the number of facets of* $\mathcal{P}_{+\epsilon}(f(\mathbf{y}))$ *and* F_p *be the* p^{th} *facet, where* $p = \{1, \ldots, P\}$*. Let* $G_d = \{\mathbf{y} + \theta \boldsymbol{\delta}_d \,|\, \theta \in \mathbb{R}\}$*, where* $d \in \{1, \ldots, D\}$*. Let* $\mathcal{Q} = \{G_d \cap F_p \,|\, d = 1, \ldots, D, p = 1, \ldots, P\}$*, in which each element differs from* $\mathbf{y}$ *on only one dimension. If adversarial cost function defined in (2) has equal cost coefficients, then there exists at least one* $\mathbf{x} \in \mathcal{Q}$ *such that* $\mathbf{x}$ *is in* ϵ*-IMAC*$(f(\mathbf{x}), \mathbf{y})$.

Corollary 1 and Corollary 2 point out an efficient way of approximating ϵ-IMAC with linear probing, which forms the backbone of our proposed algorithm in Section 4.

Finally, we consider the vulnerability of a multi-class linear classifier to linear probing. The problem arises of detecting convex polytopes in $\mathcal{X}$ with a random line. As one

can easily scale any hypercube to a unit hypercube with edge length 1, our proof is restricted to the unit hypercube in $\mathbb{R}^D$.

Definition 1 (Vulnerability to Linear Probing). *Let $\mathcal{X} = [0,1]^D$, and $\mathcal{X}_1, \ldots, \mathcal{X}_K$ be the sets that tile $\mathcal{X}$ according to the classifier $f : \mathcal{X} \to \{1, \ldots, K\}$, with $K \geq 2$. Let G be a random line in $\mathbb{R}^D$ that intersects $\mathcal{X}$. Denote Z the number of sets intersect G, the vulnerability of classifier f to linear probing is measured by the expectation of Z.*

When $\mathbb{E}\, Z$ is small, a random line intersects small number of decision regions and not much information is leaked to the adversary. Thus, a robust multi-class classifier that resists linear probing should have a small value of $\mathbb{E}\, Z$.

Theorem 3. *Let f be the multi-class linear classifier defined in (1), then the expectation of Z is bounded by $1 < \mathbb{E}\, Z < 1 + \frac{\sqrt{2}(K-1)}{2D}$.*

Proof. By Lemma 1, we have K convex polytopes $\mathcal{X}_1, \ldots, \mathcal{X}_K$. Let $\mathcal{F}$ be the union of all facets of polytopes. Observe that each time the line touches a convex polytope, it only touches its surface twice. The exit point is the entrance point for a new polytope, except at the end-point. Thus, the variable that we are interested in can be represented as

$$Z = |\mathcal{F} \cap G|,$$

where $|\cdot|$ represents the cardinality of a set. Obviously, $\mathbb{E}\, Z$ is bounded by $1 < \mathbb{E}\, Z < K$. We will give a tighter bound in the sequel.

Let $\mathcal{G}$ be the class of all lines of $\mathbb{R}^D$, and μ be the measure of $\mathcal{G}$. Following the notation introduced in [15], we denote the measure of $\mathcal{G}$ that meet a fixed bounded convex set $\mathcal{C}$ as $\mu(\mathcal{G}; \mathcal{G} \cap \mathcal{C} \neq \emptyset)$. Considering an *independent Poisson point process* on $\mathcal{G}$ intensity measure μ, let N be the number of lines intersecting $\mathcal{X}$. We emphasize that N is a finite number, so that one can label them independently $G_1, \ldots, G_N$. It follows that $G_n, n = 1, \ldots, N$ are *i.i.d.*. Given a fixed classifier f, we have

$$\begin{aligned}
\mathbb{E} \sum_{n=1}^{N} |\mathcal{F} \cap G_n| &= \mathbb{E} \sum_{n=1}^{N} \left[P(N = n) \sum_{i=1}^{n} |\mathcal{F} \cap G_i| \right] \\
&= \sum_{n=1}^{N} [P(N = n) \cdot n \cdot \mathbb{E}|\mathcal{F} \cap G_1|] \\
&= \mathbb{E}\, N \cdot (\mathbb{E}\, Z). \qquad (5)
\end{aligned}$$

Remark that $G_1, \ldots, G_N$ follow the Possion point process, we have $\mathbb{E}\, N = \mu(\mathcal{G}; \mathcal{G} \cap \mathcal{X} \neq \emptyset)$. Therefore we can rewrite (5) as,

$$\mathbb{E}\, Z = \frac{\mathbb{E} \sum_{n=1}^{N} |\mathcal{F} \cap G_n|}{\mu(\mathcal{G}; \mathcal{G} \cap \mathcal{X} \neq \emptyset)}. \qquad (6)$$

Next, we compute $\mathbb{E}\sum_{n=1}^{N}|\mathcal{F}\cap G_n|$. Let $M=|\mathcal{F}|$. Due to the convexity of $\mathcal{X}_k$, any given line can hit a facet no more than once. Therefore, we have

$$\begin{aligned}\mathbb{E}\sum_{n=1}^{N}|\mathcal{F}\cap G_n| &= \mathbb{E}\sum_{n=1}^{N}\sum_{m=1}^{M}|F_m\cap G_n| \\ &= \sum_{m=1}^{M}\mathbb{E}\Big|\Big\{n\in\{1,\dots,N\}|F_m\cap G_n\neq\emptyset\Big\}\Big| \\ &= \sum_{m=1}^{M}\mu(\mathcal{G};\mathcal{G}\cap F_m\neq\emptyset). \end{aligned} \tag{7}$$

By substituting (7) into (6) we obtain

$$\mathbb{E}\,Z=\frac{\sum_{m=1}^{M}\mu(\mathcal{G};\mathcal{G}\cap F_m\neq\emptyset)}{\mu(\mathcal{G};\mathcal{G}\cap\mathcal{X}\neq\emptyset)}. \tag{8}$$

Assume that μ is translation invariant, by Cauchy-Crofton formula we can rewrite (8) as

$$\mathbb{E}\,Z=\frac{\sum_{m=1}^{M}A(F_m)}{A(\mathcal{X})}, \tag{9}$$

where $A(\cdot)$ denotes the surface area[4]. Note, that the numerator of (9) depends on the shape of each polytope and relates to the training method of classifier. Thus, it is difficult to compute the exact value of $\mathbb{E}\,Z$. Nonetheless, we can bound the expectation by using the fact $A(\mathcal{X})<\sum_{m=1}^{M}A(F_m)<A(\mathcal{X})+\sqrt{2}(K-1)$ (see [1] for the upper bound). Remark that the surface area $A(\mathcal{X})$ of a unit hypercube is $2D$. We yield

$$1<\mathbb{E}\,Z<1+\frac{\sqrt{2}(K-1)}{2D},$$

which concludes our proof. □

We remark that Theorem 3 implies a way to construct a robust classifier that resists evasion algorithm based on linear probing, e.g. by jointly minimizing (9) and the error function in the training procedure.

4 Algorithm for Approximating ϵ-IMAC

Based on theoretical results, we present an algorithm for deceiving the multi-class linear classifier by disguising the adversarial instance $\mathbf{x}^{\mathrm{A}}$ as other classes with approximately minimal cost, while issuing polynomially many queries in: the number of features, the range of feature, the number of classes and the number of iterations.

An outline of our searching approach is presented in Algorithms 1 to 3. We use a $K\times D$ matrix Ψ for storing ISMAC of K classes and an array C of length K for

[4] The surface area in $\mathbb{R}^D$ is the $(D-1)$-dimensional Lebesgue measure.

the corresponding adversarial cost of these instances. The scalar value W represents the maximal cost of all optimum instances. Additionally, we need a $K \times I$ matrix T for storing the searching path of optimum instances in each iteration. The k^{th} row of matrix Ψ is denoted as $\Psi[k,:]$. We consider Ψ, T, C, W as global variables so they are accessible in every scope. After initializing variables, our main routine `MLCEvading` (Algorithm 1 line 4) first invokes `MDSearch` (Algorithm 2) to search instances that is close to the starting point $\mathbf{x}^{\text{A}}$ in all classes and saves them to Ψ. Then it repetitively selects instances from Ψ as new starting points and searches instances with lower adversarial cost (Algorithm 3 line 6–7). The whole procedure iterates I times. Finally, we obtain $\Psi[k,:]$ as the approximation of ϵ-IMAC$(k, \mathbf{x}^{\text{A}})$.

We begin by describing `RBSearch` in Algorithm 3, a subroutine for searching instances near decision boundaries along dimension d. Essentially, given an instance $\mathbf{x}$, an upper bound u and a lower bound l, we perform a recursive binary search on the line segment $\{\mathbf{x} + \theta\boldsymbol{\delta}_d \,|\, l \leq \theta \leq u\}$ through $\mathbf{x}$. The effectiveness of this recursive algorithm relies on the fact that it is impossible to have $\mathbf{x}^{\text{u}}$ and $\mathbf{x}^{\text{l}}$ in the same class while $\mathbf{x}^{\text{m}}$ is in another class. In particular, if the line segment meets an exterior margin $\mathcal{M}_{+\epsilon}(k)$ and ϵ-IMAC$(k, \mathbf{x})$ is the intersection, then `RBSearch` finds an ϵ-IMAC. Otherwise, when the found instance $\mathbf{y}$ yields lower adversarial cost than instance in Ψ does, Algorithm 4 is invoked to update Ψ. The time complexity of `RBSearch` is $\mathcal{O}(\frac{u-l}{\epsilon})$.

We next describe Algorithm 2. Given $\mathbf{x}$ which is known as ISMAC$(k, \mathbf{x}^{\text{A}})$ and the current maximum cost W, the algorithm iterates $(D-1)$ times on $\mathcal{P}_{+\epsilon}(\mathcal{X}_{f(\mathbf{x})})$ for finding instances with cost lower than W. Additionally, we introduce two heuristics to prune unnecessary queries. First, the searched dimension in the previous iteration of $\mathbf{x}$ is omitted. Second, we restrict the upper and lower bound of the searching scope on each dimension. Specifically, knowing W and $a(\mathbf{x}, \mathbf{x}^{\text{A}}) = c$, we only allow `RBSearch` to find instance in $[x_d - \frac{W-c}{e_d}, x_d + \frac{W-c}{e_d}]$ since any instance lying out of this scope gives adversarial cost higher than W. This pruning is significant when we have obtained ISMAC for every class. Special attention must be paid to searched dimensions of $\mathbf{x}$ (see Algorithm 2 line 5–7). Namely, if d is a searched dimension before the $(i-1)^{\text{th}}$ iteration, then we relax the searching scope to $[x_d^{\text{A}} - \frac{W-c}{e_d}, x_d^{\text{A}} + \frac{W-c}{e_d}]$ so that no low-cost instances will be missed.

Algorithm 1: Query algorithm for evasion of multi-class linear classifiers

$(\Psi, C) \leftarrow$`MLCEvading(`$\mathbf{x}^{\text{A}}, \mathbf{e}, D, L, U, K, I, \epsilon$`)`:

1 **for** $k \leftarrow 1$ **to** K **do**
2 $\Psi[k,:] \leftarrow \mathbf{0}$, $T[k,:] \leftarrow \mathbf{0}$, $C[k] \leftarrow +\infty$
3 $C[1] \leftarrow 0$
4 `MDSearch(`$\mathbf{x}^{\text{A}}, \mathbf{x}^{\text{A}}, \mathbf{e}, 1, 0, D, L, U, 1, \epsilon$`)`
5 **for** $i \leftarrow 2$ **to** I **do**
6 **for** $k \leftarrow 2$ **to** K **do**
7 `MDSearch(`$\Psi[k,:], \mathbf{x}^{\text{A}}, \mathbf{e}, k, C[k], D, L, U, i, \epsilon$`)`

Algorithm 2: Multi-dimensional search from ISMAC($k, \mathbf{x}^{\mathrm{A}}$)

`MDSearch(`$\mathbf{x}, \mathbf{x}^{\mathrm{A}}, \mathbf{e}, k, c, D, L, U, i, \epsilon$`)`:

for $d \leftarrow 1$ **to** D **do**
  if $d \neq T[k, i-1]$ **then**
    $\delta \leftarrow \frac{W-c}{e_d}$
    $u = \min\{U, x_d + \delta\}$, $l = \max\{L, x_d - \delta\}$
    if $d \in \{T[k,1], \ldots, T[k, i-2]\}$ **then**
      if $x_d > x_d^{\mathrm{A}}$ **then** $l = \max\{L, x_d^{\mathrm{A}} - \delta\}$
      else $u = \min\{U, x_d^{\mathrm{A}} + \delta\}$
    $\mathbf{x}^{\mathrm{u}} \leftarrow \mathbf{x}$, $\mathbf{x}^{\mathrm{l}} \leftarrow \mathbf{x}$
    $x_d^{\mathrm{u}} \leftarrow u$, $x_d^{\mathrm{l}} \leftarrow l$
    if $f(\mathbf{x}^{\mathrm{u}}) \neq k$ **then** `RBSearch(`$x_d, u, \mathbf{x}, d, i, \epsilon$`)`
    if $f(\mathbf{x}^{\mathrm{l}}) \neq k$ **then** `RBSearch(`$l, x_d, \mathbf{x}, d, i, \epsilon$`)`

Algorithm 3: Recursive binary search on dimension d

`RBSearch(`$l, u, \mathbf{x}, d, i, \epsilon$`)`:

$\mathbf{x}^* \leftarrow \mathbf{x}$
if $u - l < \epsilon$ **then**
  $x_d^* \leftarrow u$
  $k \leftarrow f(\mathbf{x}^*)$, $c \leftarrow a(\mathbf{x}^*)$
  if $c < C[k]$ **then** `Update(`$\mathbf{x}^*, k, c, d, i$`)`
$\mathbf{x}^{\mathrm{u}} \leftarrow \mathbf{x}$, $\mathbf{x}^{\mathrm{l}} \leftarrow \mathbf{x}$, $\mathbf{x}^{\mathrm{m}} \leftarrow \mathbf{x}$
$x_d^{\mathrm{u}} \leftarrow u$, $x_d^{\mathrm{l}} \leftarrow l$, $x_d^{\mathrm{m}} \leftarrow \frac{u+l}{2}$
if $f(\mathbf{x}^{\mathrm{m}}) = f(\mathbf{x}^{\mathrm{l}})$ **then**
  `RBSearch(`$m, u, \mathbf{x}, d, i, \epsilon$`)`
else if $f(\mathbf{x}^{\mathrm{m}}) = f(\mathbf{x}^{\mathrm{u}})$ **then**
  `RBSearch(`$l, m, \mathbf{x}, d, i, \epsilon$`)`
else
  `RBSearch(`$l, m, \mathbf{x}, d, i, \epsilon$`)`
  `RBSearch(`$m, u, \mathbf{x}, d, i, \epsilon$`)`

Algorithm 4: Update ISMAC($k, \mathbf{x}^{\mathrm{A}}$)

$(\Psi, C, T, W) \leftarrow$`Update(`$\mathbf{x}^*, k, c, d, i$`)`:

$\Psi[k, :] \leftarrow \mathbf{x}^*$
$C[k] \leftarrow c$
$T[k, i] \leftarrow d$
$W \leftarrow \max\{C[1], \ldots, C[K]\}$

Theorem 4. *The asymptotic time complexity of our algorithm is* $\mathcal{O}(\frac{U-L}{\epsilon} DKI)$.

Proof. Follows from the correctness of the algorithm and the fact that the time complexity of `RBSearch` is $\mathcal{O}(\frac{u-l}{\epsilon})$. □

5 Experiments

We demonstrate the algorithm[5] on two real-world data sets, the 20-newsgroups[6] and the 10-Japanese female face[7]. On the newsgroups data set, the task of the adversary is to evade a text classifier by disguising a commercial spam as a message in other topics. On the face data set, the task of adversary is to deceive the classifier by disguising a suspect's face as an innocent. We employ LIBLINEAR [6] package to build target multi-class linear classifiers, which return labels of queried instances. The cost coefficients are set to $e_1 = \cdots = e_D = 1$ for both tasks. For the groundtruth solution, we directly solve the optimization problem with linear constraints (3) and (4) by using the models' parameters. We then measure the average empirical ϵ for $(K-1)$ classes, which is defined as $\widehat{\epsilon} = \frac{1}{K-1} \sum_{k \neq f(\mathbf{x}^{\mathrm{A}})} \left[\frac{C[k]}{\mathrm{MAC}(k, \mathbf{x}^{\mathrm{A}})} - 1 \right]$, where $C[k]$ is the adversarial cost of disguised instance of class k. Evidently, small $\widehat{\epsilon}$ indicates better approximation of IMAC.

5.1 Spam Disguising

The training data used to configure the newsletter classifier consists of $7,505$ documents, which are partitioned evenly across 20 different newsgroups. Each document is represented as a $61,188$-dimensional vector, where each component is the number of occurrences of a word. The accuracy of the classifier on training data is 100% for every class. We set the category "misc.forsale" as the adversarial class. That is, given a random document in "misc.forsale", the adversary attempts to disguise this document as from other category; e.g. "rec.sport.baseball". Parameters of the algorithm are $K = 20, L = 0, U = 100, I = 10, \epsilon = 1$. The adversary is restricted to query at most $10,000$ times. The adversarial cost of each class is depicted in Fig. 1 (left).

5.2 Face Camouflage

The training data contains 210 gray-scaled images of 7 facial expressions (each with 3 images) posed by 10 Japanese female subjects. Each image is represented by a 100-dimensional vector using principal components. The accuracy of the classifier on training data is 100% for every class. We randomly pick a subject as an imaginary suspect. Given a face image of the suspect, the adversary camouflage this face to make it be classified as other subjects. Parameters of the algorithm are $K = 10, L = -10^5, U = 10^5, I = 10, \epsilon = 1$. The adversary is restricted to query at most $10,000$ times. The adversarial cost of each class is depicted in Fig. 1 (right). Moreover, we visualize disguised faces in Fig. 2. Observe that many disguised faces are similar to the suspect's face by humans interpretation, yet they are deceptive for the classifier. This visualization directly demonstrates the effectiveness of our algorithm.

[5] A Matlab implementation is available at http://home.in.tum.de/∼xiaoh/pakdd2012-code.zip

[6] http://people.csail.mit.edu/jrennie/20Newsgroups/

[7] http://www.kasrl.org/jaffe.html

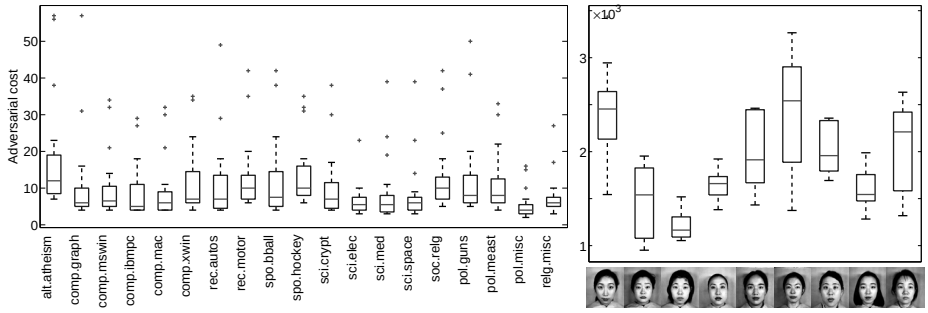

Fig. 1. Box plots for adversarial cost of disguised instance of each class. **(Left)** On the 20-newsgroups data set, we consider "misc.forsale" as the adversarial class. Note, that feature values of the instance are non-negative integers as they represent the number of words in the document. Therefore, the adversarial cost can be interpreted as the number of modified words in the disguised document comparing to the original document from "misc.forsale". The value of $\widehat{\epsilon}$ for 19 classes is 0.79. **(Right)** On the 10-Japanese female faces data set, we randomly select a subject as the suspect. The box plot shows that the adversarial cost of camouflage suspicious faces as other subjects. The value of $\widehat{\epsilon}$ for 9 classes is 0.51. A more illustrative result is depicted in Fig. 2.

It has not escaped our notice that an experienced adversary with certain domain knowledge can reduce the number of queries by careful selecting cost function and employing heuristics. Nonetheless, the goal of this paper is not to design real attacks but rather examine the correctness and effectiveness of our algorithm so as to understand vulnerabilities of classifiers.

6 Conclusions

Adversary and classifier are *Yin* and *Yang* of information security. We believe that understanding the vulnerability of classifiers is the only way to develop resistant classifiers in the future. In this paper, we showed that multi-class linear classifiers are vulnerable to the evasion attack and presented an algorithm for disguising the adversarial instance. Future work includes generalizing the evasion attack problem to the family of general multi-class classifier with nonlinear decision boundaries.

References

1. Ball, K.: Cube slicing in $\mathbb{R}^n$. Proc. American Mathematical Society 97(3), pp. 465–473 (1986)
2. Barbara, D., Jajodia, S.: Applications of data mining in computer security. Springer (2002)
3. Bratko, A., Filipič, B., Cormack, G., Lynam, T., Zupan, B.: Spam filtering using statistical data compression models. JMLR 7, 2673–2698 (2006)
4. Crammer, K., Singer, Y.: On the learnability and design of output codes for multiclass problems. Machine Learning 47(2), 201–233 (2002)
5. Dalvi, N., Domingos, P., et al.: Adversarial classification. In: Proc. 10th SIGKDD. pp. 99–108. ACM (2004)

Fig. 2. Disguised faces given by our algorithm to defeat a multi-class face recognition system. The original faces (with neutral expression) of 10 females are depicted in the first row, where the left most one is the imaginary suspect and the remaining 9 people are innocents. From the second row to sixth row, faces of the suspect with different facial expressions are fed to the algorithm (see the first column). The output disguised faces from the algorithm are visualized in the right hand image matrix. Each row corresponds to disguised faces of the input suspicious face on the left. Each column corresponds to an innocent.

6. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. JMLR 9, 1871–1874 (2008)
7. Fumera, G., Pillai, I., Roli, F.: Spam filtering based on the analysis of text information embedded into images. JMLR 7, 2699–2720 (2006)
8. Grünbaum, B.: Convex polytopes, vol. 221. Springer (2003)
9. Keerthi, S., Sundararajan, S., Chang, K., Hsieh, C., Lin, C.: A sequential dual method for large scale multi-class linear svms. In: Proc. 14th SIGKDD. pp. 408–416. ACM (2008)
10. Lowd, D., Meek, C.: Adversarial learning. In: Proc. 11th SIGKDD. pp. 641–647. ACM (2005)
11. Lowd, D., Meek, C.: Good word attacks on statistical spam filters. In: Proc. 2nd Conference on Email and Anti-Spam. pp. 125–132 (2005)
12. Maloof, M.: Machine learning and data mining for computer security: methods and applications. Springer (2006)
13. Nelson, B., Rubinstein, B.I.P., Huang, L., Joseph, A.D., hon Lau, S., Lee, S., Rao, S., Tran, A., Tygar, J.D.: Near-optimal evasion of convex-inducing classifiers. In: Proc. 13th AISTATS (2010)
14. Rockafellar, R.: Convex analysis, vol. 28. Princeton Univ Pr (1997)
15. Santaló, L.: Integral geometry and geometric probability. Cambridge Univ Pr (2004)