A Fundamental Performance Limitation for Adversarial Classification

Abed AlRahman Al Makdah, Vaibhav Katewa, and Fabio Pasqualetti

***Abstract*—Despite the widespread use of machine learning algorithms to solve problems of technological, economic, and social relevance, provable guarantees on the performance of these data-driven algorithms are critically lacking, especially when the data originates from unreliable sources and is transmitted over unprotected and easily accessible channels. In this letter, we take an important step to bridge this gap and formally show that, in a quest to optimize their accuracy, binary classification algorithms—including those based on machine-learning techniques—inevitably become more sensitive to adversarial manipulation of the data. Further, numerical evidence suggests that the accuracy-sensitivity tradeoff depends solely on the statistics of the data, and cannot be improved by tuning the algorithms or increasing their complexity.**

***Index Terms*—Pattern recognition and classification, machine learning, neural networks.**

I. Introduction

ARTIFICIAL intelligence and machine learning algorithms, including neural networks, are used widely in technological, social, and economic applications, such as computer vision, speech recognition, malware detection, and control design. For control applications, in particular, these data-driven algorithms are attracting increasingly more attention, as they promise to overcome the limitations of traditional model-based approaches, especially when the models are too complex to be useful, or too difficult to estimate or derive from first principles [1]–[3]. While these algorithms typically achieve high performance under nominal and well-modeled conditions, they are also very sensitive to small, targeted, and possibly malicious manipulation of the training and execution data [4]. A theoretical understanding of this unreliable behavior is still lacking, thus motivating the critical need for novel theories to deploy robust, reliable, and safe data-driven algorithms.

In this letter we formally study a fundamental tradeoff between the accuracy of a binary classification algorithm and its sensitivity to arbitrary manipulation of the data. In particular, we cast a binary classification problem into a hypothesis testing framework, parametrize classification algorithms – including those based on machine learning techniques – using their decision boundaries, and show that the accuracy of the algorithm can be maximized only at the expenses of its sensitivity. This tradeoff, which applies to general classification algorithms, depends on the statistics of the data, and cannot be improved by simply tuning the algorithm. Our theory explains how simple algorithms can outperform more complex ones when operating in adversarial environments.

Related Work: Recent work has shown that classification based on neural networks is vulnerable to adversarial perturbations [4], [5], and that these perturbations are universal and affect a large number of classification algorithms. While heuristic explanations of this phenomenon have been proposed, including adversarial learning [5]–[7], black-box [8], and gradient-based [5], [6], a fundamental analytical understanding of the limitations of classification algorithms under adversarial perturbations is critically lacking. We identify these limitations for a binary classification problem in a Bayesian setting. While in a simple setting, our analysis formally shows that a fundamental tradeoff exists between accuracy and sensitivity of any classification algorithm, independently of the complexity of the algorithm. The papers [9], [10] are also related to this letter, which derive methods to measure robustness of different classifiers against adversarial perturbations and obtain guarantees against bounded perturbations, as well as [7], which shows how adversarial training improves the classifier's performance against adversarial perturbations while deteriorating its performance under nominal conditions. Distributionally robust optimization has also been used to develop robust classifiers [11]. Yet, this theory does not formally explain the tradeoff highlighted in this letter. Our approach provides rigorous support to the empirical evidence obtained in these works.

Contribution: This letter features three main contributions. First, we propose metrics to quantify the accuracy of a classification algorithm and its sensitivity to arbitrary manipulation of the data. We prove that, under a set of mild technical assumptions, the accuracy of a classification algorithm can

Manuscript received March 1, 2019; revised May 14, 2019; accepted June 4, 2019. Date of publication June 10, 2019; date of current version June 24, 2019. This work was supported in part by ARO under Award ARO-71603NSYIP, and in part by ONR under Award ONR-N00014-19-1-2264. Recommended by Senior Editor G. Cherubini. *(Corresponding author: Abed AlRahman Al Makdah.)*

The authors are with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92521 USA, and also with the Department of Mechanical Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: aalmakdah@engr.ucr.edu; vkatewa@engr.ucr.edu; fabiopas@engr.ucr.edu).

Digital Object Identifier 10.1109/LCSYS.2019.2921989

only be maximized at the expenses of its sensitivity. Thus, a fundamental tradeoff exists between the performance of a classification algorithm in nominal and adversarial settings. While our results formally apply to binary classification problems, we conjecture that this fundamental tradeoff in fact applies to more general classification problems. Second, we show that a tradeoff between accuracy and sensitivity exists for different classes of classification algorithms, and that simpler algorithms can sometimes outperform more complex one in adversarial settings. Third, we numerically show that the accuracy versus sensitivity tradeoff depends solely on the statistics of the data, and cannot be arbitrarily improved by tuning the classification algorithm (varying classification boundaries) or increasing its complexity (number of boundaries), including using sophisticated adversarial learning techniques. Taken together, our results suggest that performance and robustness of data-driven algorithms are dictated by the properties of the data, and not by the sophistication or intelligence of the algorithm, a key insight that has critical implications for the deployment of provably-robust data-driven and learning-based control algorithms.

II. Problem Setup and Preliminary Notions

To reveal a fundamental tradeoff between the accuracy of a classification algorithm and its robustness against malicious data manipulation, we consider a binary classification problem where the objective is to decide whether a scalar observation $x \in \mathbb{R}$ belongs to one of the classes $\mathcal{H}_0$ and $\mathcal{H}_1$. We assume that the distribution of the observations satisfy

$$\mathcal{H}_0 : x \sim f_0(x;\theta_0), \text{ and } \mathcal{H}_1 : x \sim f_1(x;\theta_1), \qquad (1)$$

where $f_0(x;\theta_0)$ and $f_1(x;\theta_1)$ are arbitrary, yet known, probability density functions with parameters $\theta_0 \in \mathbb{R}^{m_0}$ and $\theta_1 \in \mathbb{R}^{m_1}$, respectively. We assume that the partial derivatives of f_k with respect to x and θ_k exist and are continuous over the domain of the distributions, for $k = 0, 1$. Let p_0 and p_1 denote the prior probabilities of the observations belonging to the classes $\mathcal{H}_0$ and $\mathcal{H}_1$, respectively. Different (machine learning) algorithms can be used to solve the above binary classification problem. Yet, because of the binary nature of the problem, any classification algorithm can be represented by a suitable partition of the real line, and it can be written as

$$\mathfrak{C}(x;y) = \begin{cases} \mathcal{H}_0, & x \in \mathcal{R}_0, \\ \mathcal{H}_1, & x \in \mathcal{R}_1, \end{cases} \qquad (2)$$

where[1] $y = [y_i]$ denotes a set of boundary points, with $y_0 \leq \cdots \leq y_{n+1}$, $y_0 = -\infty$, $y_{n+1} = \infty$, and

$$\mathcal{R}_0 = \{z : y_i < z < y_{i+1}, \text{ with } i = 0, 2, \ldots, n\},$$
$$\mathcal{R}_1 = \{z : y_i \leq z \leq y_{i+1}, \text{ with } i = 1, 3, \ldots, n-1\}.$$

We refer to (2) as general classifier. We measure the performance of a classification algorithm through its *accuracy*, that is, its probability of making a correct classification.

*Definition 1 (**Accuracy of a Classifier**):* The accuracy of the classification algorithm $\mathfrak{C}(x;y)$ is

$$\mathcal{A}(y;\theta) = p_0\mathbf{P}[x \in \mathcal{R}_0|\mathcal{H}_0] + p_1\mathbf{P}[x \in \mathcal{R}_1|\mathcal{H}_1], \qquad (3)$$

where $\theta = [\theta_0^\mathsf{T}\ \theta_1^\mathsf{T}]^\mathsf{T}$ contains the distribution parameters.

Using Equation (3) and the distributions in (1), we obtain

$$\mathcal{A}(y;\theta) = p_0\left(\sum_{l=1}^{n}(-1)^{l+1}\int_{-\infty}^{y_l} f_0(x;\theta_0)dx + 1\right)$$
$$+ p_1\left(\sum_{l=1}^{n}(-1)^{l}\int_{-\infty}^{y_l} f_1(x;\theta_1)dx\right). \qquad (4)$$

Clearly, the accuracy of a classification algorithm depends on the position of its boundaries, which can be selected to maximize the accuracy of the classification algorithm. To this aim, let $L(x)$ denote the Likelihood Ratio defined as

$$L(x) = \frac{p_1 f_1(x;\theta_1)}{p_0 f_0(x;\theta_0)}.$$

The Maximum Likelihood (ML) classifier is

$$\mathfrak{C}_{\mathrm{ML}}(x;\eta) = \begin{cases} \mathcal{H}_0, & L(x) < \eta, \\ \mathcal{H}_1, & L(x) \geq \eta, \end{cases} \qquad (5)$$

where the threshold $\eta > 0$ is a design parameter that determines the boundary points and, thus, the accuracy of the classifier. As a known result in statistical hypothesis testing [12], the accuracy of the ML classifier with $\eta = 1$ is the largest among all possible classifiers. The value and the number of boundary points of the ML classifier depend on the distributions $f_0(x;\theta_0)$ and $f_1(x;\theta_1)$, the threshold η, and the prior probabilities through the equation

$$p_1 f_1(x;\theta_1) - \eta p_0 f_0(x;\theta_0) = 0. \qquad (6)$$

Another important class of classifiers is the class of linear classifiers, which are less complex and often achieve a competitive performance compared to nonlinear classifiers (see [13] for more details). In our setting, a linear classifier consists of one decision boundary $y \in \mathbb{R}$, and is given by

$$\mathfrak{C}_{\mathrm{L}}(x;y) = \begin{cases} \mathcal{H}_0, & x < y, \\ \mathcal{H}_1, & x \geq y. \end{cases} \qquad (7)$$

Following Definition 1, the accuracy of $\mathfrak{C}_{\mathrm{L}}$ is

$$\mathcal{A}(y;\theta) = p_0\int_{-\infty}^{y} f_0(x;\theta_0)dx - p_1\int_{-\infty}^{y} f_1(x;\theta_1)dx + p_1. \qquad (8)$$

The optimal boundary y_{L}^* that maximizes $\mathcal{A}(y;\theta)$ is

$$y_{\mathrm{L}}^* = \arg\max_{y_i} \quad \mathcal{A}(y_i;\theta)$$
$$\text{s.t.} \quad y_i \text{ is a solution of (6) with } \eta = 1. \qquad (9)$$

While the boundaries are difficult to compute for general distributions, they can be computed explicitly when the observations are Gaussian (see below). Let $\mathcal{N}(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ be the p.d.f. of a normal random variable with

[1] For simplicity and without affecting generality, we assume that n is even. Further, an alternative configuration of the classifier (2) assigns $\mathcal{H}_0$ and $\mathcal{H}_1$ to $\mathcal{R}_1$ and $\mathcal{R}_0$, respectively. However, because accuracy and sensitivity of the two configurations can be obtained from each other, we consider only the configuration in (2) without affecting the generality of our analysis.

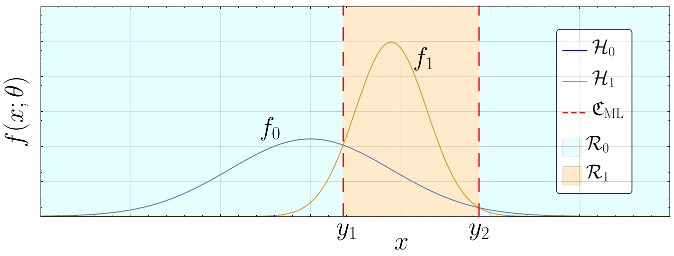

Fig. 1. The distributions of x under Gaussian hypotheses with $\mu_0 = 0$, $\sigma_0 = 9$, $\mu_1 = 9$, $\sigma_1 = 4$, and $p_0 = p_1 = 0.5$. The dashed red lines represent the decision boundaries of the $\mathfrak{C}_{\mathrm{ML}}(x; \eta = 1)$, which divide the space into $\mathcal{R}_0$ (represented by the blue region) and $\mathcal{R}_1$ (orange region).

mean μ and variance σ, and $Q(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} dx$ the c.d.f. of the standard normal distribution.

Remark 1 ***(ML and Linear Classifiers for Gaussian Distributions):*** For the Gaussian distributions $f_i(x; \theta_i) = \mathcal{N}(x; \mu_i, \sigma_i)$, $i = 0, 1$, the boundaries of ML classifier satisfy

$$ax^2 + bx + c = 0 \qquad \text{where,} \tag{10}$$

$$a = \frac{1}{2}\left(\frac{1}{\sigma_0^2} - \frac{1}{\sigma_1^2}\right), b = \left(\frac{\mu_1}{\sigma_1^2} - \frac{\mu_0}{\sigma_0^2}\right), \text{ and}$$

$$c = \log\left(\frac{\sigma_0}{\sigma_1}\right) + \log\left(\frac{p_1}{p_0}\right) + \frac{\mu_0^2}{2\sigma_0^2} - \frac{\mu_1^2}{2\sigma_1^2} - \log(\eta).$$

Equation (10) has at most two real solutions, implying that the ML classifier has at most two decision boundaries (see Fig. 1). The ML classifier with boundaries corresponding to the solutions of (10) with $\eta = 1$ has maximum accuracy [12]. The solution of (10) which maximizes the accuracy in (8) is the boundary for the optimal linear classifier.

In this letter, we consider adversarial manipulations of the observations in which an attacker is capable of adding deterministic or random perturbations to the observations in order to degrade the performance of the classifier. We model such manipulations as modification to the parameters of distributions in (1), i.e., the attacker can change the parameter θ. To characterize the robustness of a classifier to these adversarial manipulations of the observations, we define the following sensitivity metric, which captures the degradation of the classification accuracy following data manipulation.

Definition 2 ***(Sensitivity of a Classifier):*** The sensitivity of the classification algorithm[2] $\mathfrak{C}(x; y)$ is

$$\mathcal{S}(y; \theta) = \left\| \frac{\partial \mathcal{A}(y; \theta)}{\partial \theta} \right\|_\infty, \tag{11}$$

where θ contains the parameters of the distributions in (1), and $\mathcal{A}(y; \theta)$ denotes the accuracy of $\mathfrak{C}(x; y)$.

From Definition 2, a higher value of sensitivity implies that the adversary can affect the classifier's performance to a larger extent, whereas a lower sensitivity implies that the classifier is more robust to adversarial manipulation. Further, the $\infty-$norm captures the worst case in terms of the largest sensitivity with respect to the components of θ. Finally, the sensitivity vector $\frac{\partial \mathcal{A}(y;\theta)}{\partial \theta}$ can be used to determine a perturbation to θ that maximizes (locally) the degradation of the classifier.

Remark 2 ***(Comparison With Adversarial Classification):*** In adversarial classification, the attacker designs a perturbation for a given observation (e.g., an image) to induce misclassification [4], [7]. Such observation can be viewed as a realization of a multi-dimensional distribution. In contrast, we consider perturbations of the distribution, which affect all the realizations, and focus on the average reduced performance of the classifier over all realizations. Despite this difference, our sensitivity vector and its norm capture the direction and the extent of the worst-case perturbation, similar to the worst-case smallest perturbation in adversarial classification, allow us to obtain formal guarantees, and provide additional insight into the performance limitations of adversarial classification.

Remark 3 ***(Accuracy and Sensitivity of the ML Classifier for Gaussian Distributions):*** The accuracy and the sensitivity of the ML classifier are obtained by substituting the expression of the normal distributions $\mathcal{N}(x; \mu_i, \sigma_i)$ in (3) and (11):

$$\mathcal{A}(y; \theta) = p_0\Big(Q\Big(\frac{y_1 - \mu_0}{\sigma_0}\Big) - Q\Big(\frac{y_2 - \mu_0}{\sigma_0}\Big) + 1\Big) + p_1\Big(- Q\Big(\frac{y_1 - \mu_1}{\sigma_1}\Big) + Q\Big(\frac{y_2 - \mu_1}{\sigma_1}\Big)\Big) \text{ and,}$$

$$\mathcal{S}(y; \theta) = \left\| \begin{bmatrix} p_0\Big(f_0(y_2; \theta_0) - f_0(y_1; \theta_0)\Big) \\ p_0\Big(\frac{\mu_0 - y_1}{\sigma_0} f_0(y_1; \theta_0) - \frac{\mu_0 - y_2}{\sigma_0} f_0(y_2; \theta_0)\Big) \\ p_1\Big(f_1(y_1; \theta_1) - f_1(y_2; \theta_1)\Big) \\ p_1\Big(\frac{\mu_1 - y_2}{\sigma_1} f_1(y_2; \theta_1) - \frac{\mu_1 - y_1}{\sigma_1} f_1(y_1; \theta_1)\Big) \end{bmatrix} \right\|_\infty,$$

where $\theta_i = [\mu_i \ \sigma_i]^{\mathsf{T}}$ and $i = 0, 1$.

A classification algorithm should have high accuracy and low sensitivity, so as to exhibit robust performance against adversarial manipulation. Unfortunately, we show that accuracy and sensitivity are directly related, so that optimizing the accuracy of a classifier inevitably increases its sensitivity.

III. A Fundamental Tradeoff Between Accuracy and Sensitivity of Classification Algorithms

In this section, we characterize a tradeoff between accuracy and sensitivity of a classification algorithm for the binary classification problem in (1). We prove that, under some mild conditions, there exist a classifier that is less accurate than $\mathfrak{C}_{\mathrm{ML}}(x; 1)$, yet more robust to adversarial manipulation of the data. This shows that there exist a tradeoff between accuracy and sensitivity at the configuration of maximum accuracy.

Let $y^* = [y_1^* \ y_2^* \ \cdots \ y_n^*]^{\mathsf{T}}$ be the vector of the boundaries of $\mathfrak{C}_{\mathrm{ML}}(x; 1)$, which maximizes $\mathcal{A}(y; \theta)$. Let $\theta^{(i)}$ be the i^{th} component of θ. We make the following assumptions:

A1: The vector $\frac{\partial \mathcal{A}(y;\theta)}{\partial \theta}\Big|_{y^*}$ has a unique largest absolute element, located at index j.

A2: There exist at least one boundary y_i^* such that

$$\left(p_0 \frac{\partial}{\partial y_i} f_0(y_i; \theta_0)\Big|_{y_i^*} - p_1 \frac{\partial}{\partial y_i} f_1(y_i; \theta_1)\Big|_{y_i^*}\right) \frac{\partial y_i^*}{\partial \theta^{(j)}} \neq 0.$$

Assumption A1 is specific to our definition of sensitivity in (11), and is not required if $2-$norm is used (see Remark 5). Further, A2 is mild and typically satisfied in most problems.

[2] Definition 2 is also valid for the ML and the linear classifier.

Theorem 1 ***(Accuracy-Sensitivity Tradeoff for General Classifier (2)):*** Let y^* contain the boundaries of the classifier $\mathfrak{C}_{\mathrm{ML}}(x;1)$. Then, under Assumptions A1 and A2, it holds

$$\left.\frac{\partial\mathcal{S}(y;\theta)}{\partial y}\right|_{y^*}\neq 0. \tag{12}$$

Proof: Assumption A1 guarantees that $\mathcal{S}(y;\theta)$ is differentiable with respect to y at y^*. Let $g\big(y;\theta\big)\triangleq\frac{\partial\mathcal{A}(y;\theta)}{\partial y}$. Since y^* maximizes $\mathcal{A}(y;\theta)$, $g\big(y^*;\theta\big)=0$. Differentiating $g\big(y^*;\theta\big)$ with respect to $\theta^{(j)}$, and noting that y^* depends on θ, we get:

$$\frac{\mathsf{d}g\big(y^*;\theta\big)}{\mathsf{d}\theta^{(j)}}=\left.\frac{\partial g\big(y;\theta\big)}{\partial\theta^{(j)}}\right|_{y^*}+\left.\frac{\partial g\big(y;\theta\big)}{\partial y}\right|_{y^*}\frac{\partial y^*}{\partial\theta^{(j)}}=0,$$

$$\Rightarrow\frac{\partial}{\partial y}\left.\frac{\partial\mathcal{A}(y;\theta)}{\partial\theta^{(j)}}\right|_{y^*}=-\left.\frac{\partial^2\mathcal{A}(y;\theta)}{\partial y^2}\right|_{y^*}\frac{\partial y^*}{\partial\theta^{(j)}}, \tag{13}$$

where the last equation follows by substituting $g\big(y;\theta\big)=\frac{\partial\mathcal{A}(y;\theta)}{\partial y}$ and switching the order of partial differentiation. Using (11), it can be easily observed that the left side of (13) equals $\pm\left.\frac{\partial\mathcal{S}(y;\theta)}{\partial y}\right|_{y^*}$. Further, differentiating (4) twice, we get $\frac{\partial^2}{\partial y^2}\mathcal{A}(y;\theta)=\mathrm{diag}(w_1(y_1),\dots,w_n(y_n))$, where

$$w_i(y_i)=p_0(-1)^{i+1}\frac{\partial}{\partial y_i}f_0(y_i;\theta_0)+p_1(-1)^i\frac{\partial}{\partial y_i}f_1(y_i;\theta_1).$$

Assumption A2 guarantees that there exist a boundary y_i^* such that $w_i(y_i^*)\frac{\partial y_i^*}{\partial\theta^{(j)}}\neq 0$. The result follows from (13). ■

Theorem 1 implies that the sensitivity of the classifier $\mathfrak{C}(x;y)$ can be decreased by modifying the boundaries y^*. Yet, because $\mathfrak{C}(x;y^*)$ exhibits the largest classification accuracy among all classifiers, the reduction of sensitivity inevitably decreases the accuracy of classification. In other words, for any classification problem (1) satisfying Assumptions A1 and A2 and for any classification algorithm (2), there exists an arbitrarily small δ such that[3]

$$\mathcal{S}(y^*+\delta;\theta)<\mathcal{S}(y^*;\theta)\text{ and }\mathcal{A}(y^*+\delta;\theta)\leq\mathcal{A}(y^*;\theta).$$

Thus, a fundamental tradeoff exists between the accuracy of a classifier and its robustness to adversarial manipulation. Note that the result of Theorem 1 holds for all distributions that satisfy Assumptions A1 and A2. Further, we show next that such tradeoff also exists for linear and ML classifiers, and for multi-dimensional digit classifier based on a neural network (Section IV). This tradeoff is observed for a large class of problems, thereby highlighting its fundamental nature.

Corollary 1 ***(Accuracy-Sensitivity Tradeoff for the Linear Classifier (7)):*** Let y_{L}^* be the boundary given in (9) that maximizes the accuracy (in (8)) of the linear classifier $\mathfrak{C}_{\mathrm{L}}(x;y)$. Then, under Assumptions A1 and A2, it holds

$$\left.\frac{\partial\mathcal{S}(y;\theta)}{\partial y}\right|_{y_{\mathrm{L}}^*}\neq 0. \tag{14}$$

Proof: Since y_{L}^* corresponds to one of the boundaries contained in y^*, the proof follows from Theorem 1. ■

Next, we show that this tradeoff also exists for the Maximum Likelihood classifier. This fact does not follow trivially from Theorem 1, because the general classifier in the theorem has independent boundaries, while the boundaries of the ML classifier are dependent on one another via (6). We make the following mild technical assumption.

A3: The vectors $\left.\frac{\partial y(\eta,\theta)}{\partial\eta}\right|_{\eta=1}$ and $\left.\frac{\partial\mathcal{S}(y;\theta)}{\partial y}\right|_{y^*}$ are not orthogonal, where $y(\eta,\theta)$ contains the boundaries of $\mathfrak{C}_{\mathrm{ML}}(x;\eta)$.

Lemma 1 ***(Accuracy-Sensitivity Tradeoff for the ML Classifier (5)):*** Let $y(\eta,\theta)$ contain the boundaries of the classifier $\mathfrak{C}_{\mathrm{ML}}(x;\eta)$. Then, under Assumptions A1, A2 and A3, it holds

$$\left.\frac{\partial\mathcal{S}(y(\eta,\theta);\theta)}{\partial\eta}\right|_{\eta=1}\neq 0.$$

Proof: Let y^* contain the boundaries of the classifier $\mathfrak{C}_{\mathrm{ML}}(x;\eta=1)$. The derivative of $\mathcal{S}\big(y(\eta,\theta);\theta\big)$ with respect to η can be written as:

$$\left.\frac{\partial\mathcal{S}\big(y(\eta,\theta);\theta\big)}{\partial\eta}\right|_{\eta=1}=\left.\frac{\partial\mathcal{S}\big(y;\theta\big)}{\partial y^{\mathsf{T}}}\right|_{y^*}\left.\frac{\partial y(\eta,\theta)}{\partial\eta}\right|_{\eta=1}.$$

We conclude following Theorem 1 and Assumption A3. ■

In what follows we numerically show that a tradeoff between accuracy and sensitivity also exists when the classification boundaries are not selected to maximize the accuracy of the classifier. To this aim, first we compute the accuracy and sensitivity of the ML classifier $\mathfrak{C}_{\mathrm{ML}}(x;\eta)$, for different values of η. Notice that, by varying $0<\eta<\infty$, Equation (6) returns different classification boundaries and, thus, different classification algorithms. Similarly, we compute the accuracy and sensitivity of linear classifier $\mathfrak{C}_{\mathrm{L}}(x;y)$ by varying the single boundary y. Second, we numerically solve

$$\begin{aligned}\min_{y}\quad&\mathcal{S}(y;\theta)\\ \text{s.t.}\quad&\mathcal{A}(y;\theta)=\zeta,\end{aligned} \tag{15}$$

for different values of ζ ranging from 0.5 to $\mathcal{A}(y^*;\theta)$. Notice that the minimization problem (15) returns the classifier with lowest sensitivity and accuracy equal to ζ, and that the boundaries solving the minimization problem (15) may not satisfy (6). Further, for a given number of classification boundaries, the minimization problem (15) returns a fundamental tradeoff curve relating accuracy and sensitivity over the range of ζ, which is independent of the choice of classification algorithm. Finally, the minimization problem (15) is not convex, because of its nonlinear equality constraint.

Fig. 2(a) shows the accuracy-sensitivity tradeoff for the Gaussian hypothesis testing problem discussed in Remark 3. In this case, since the ML classifier has 2 boundaries, we also consider general classifiers with 2 boundaries. We observe that the general classifier exhibits the tradeoff at the maximum accuracy point (identified by the red dot) in accordance with Theorem 1. Several comments are in order. First, the ML and linear classifiers also exhibit tradeoff at their respective maximum accuracy points in accordance with Lemma 1 and Corollary 1. Second, the tradeoff for the ML classifier is not strict and there exist points where reducing accuracy increases sensitivity (green dot in the figure). On the other hand, the tradeoff for the general classifier is strict. This might be because the decision boundaries of the general classifier can be varied independently, whereas the boundaries of the ML

[3] The inequality for accuracy is strict for most distributions.

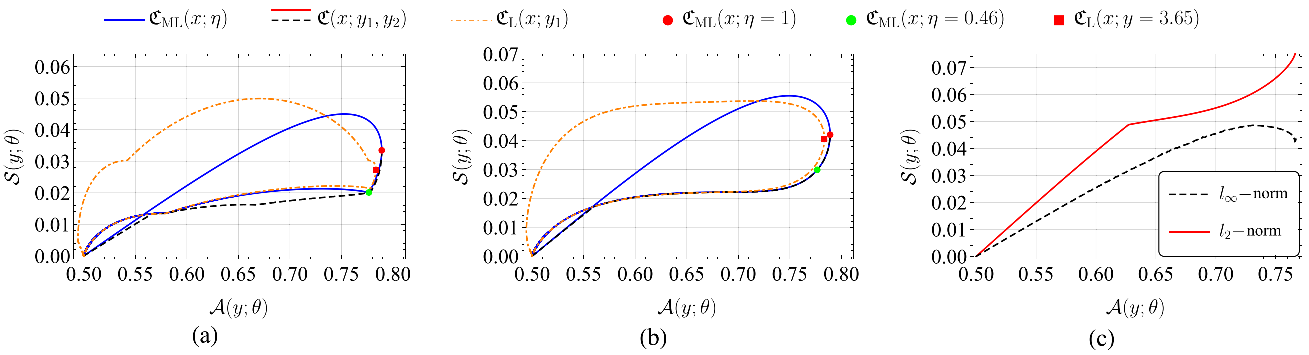

Fig. 2. Accuracy-sensitivity tradeoff curves for a general classifier with 2 boundaries (black dashed line), the ML classifier (blue line), and a linear classifier (orange dash-dotted line) corresponding to the Gaussian hypothesis testing problem. The parameters of the two distributions for (a)-(b) are $\mu_0 = 0$, $\sigma_0 = 9$, $\mu_1 = 9$, and $\sigma_1 = 4$, and for (c) are $\mu_0 = 0$, $\sigma_0 = 4$, $\mu_1 = 5$, and $\sigma_1 = 3$. The red dot represents $\mathfrak{C}_{\mathrm{ML}}(x; 1)$ (maximum accuracy point) and the green dot represents $\mathfrak{C}_{\mathrm{ML}}(x; 0.46)$. The red square represents $\mathfrak{C}_{\mathrm{L}}(x; y = 3.65)$, which is the linear classifier with maximum accuracy. The sensitivity in (a) and for the black dashed line in (c) is computed using Definition 2, while the sensitivity in (b) and for the red line in (c) is computed using (16).

classifier are related to each other since they are the solutions of (6). Thus, the general classifier provides more flexibility in choosing the boundaries, which induces lower sensitivity as compared to the ML classifier, and ultimately, results in a strict tradeoff. Similarly, the tradeoff for the linear classifier is not strict. Third, the tradeoff curve for the general classifier is below the tradeoff curves for the ML and linear classifiers, again, due to the aforementioned reason.[4] Fourth, the maximum accuracy of the linear classifier (corresponding to red square) is smaller than that of the ML classifier (corresponding to the red dot), but its sensitivity at the maximum accuracy configuration is also smaller than that of the ML classifier. This explains the observed phenomena that in some cases, linear models are more robust to adversarial attacks than nonlinear models (for example, neural networks) [14]. Finally, the curves are not smooth because of the ∞-norm in Definition 2.

Next, we present two remarks on using the 2-norm to define sensitivity and on the necessity of Assumption A1.

Remark 4 ***(Classification Sensitivity Using the 2–Norm):*** In Definition 2, the ∞-norm captures the largest change in accuracy with respect to a change in a single component of parameters vector θ. Instead, using the 2-norm to define the sensitivity of a classification algorithm leads to

$$\mathcal{S}(y;\theta) = \left\| \frac{\partial \mathcal{A}(y;\theta)}{\partial \theta} \right\|_2, \tag{16}$$

which captures the change in accuracy with respect to changes in all the components of θ. Fig. 2(b) shows the sensitivity versus accuracy tradeoff when sensitivity is defined using (16) instead of (11). For this case, a strict tradeoff is observed for all classifiers, although this may not be the case in general. Further, the tradeoff curves are smooth.

Remark 5 ***(Necessity of Assumption A1):*** Assumption A1 is required to ensure differentiability of the sensitivity in (11), and thus, it is required for Theorem 1. In contrast, the sensitivity defined in (16) is always differentiable, and A1 is not required in this case. We illustrate this in Fig. 2(c), where the vector $\frac{\partial \mathcal{A}(y^*;\theta)}{\partial \theta} = [0.043,\ 0.024,\ -0.043,\ 0.040]^{\mathsf{T}}$ has two elements with maximum absolute value, violating Assumption A1. We observe that a tradeoff at the maximum accuracy point (denoted by the red dot) does not exist in this case using (11), while it still exists using (16).

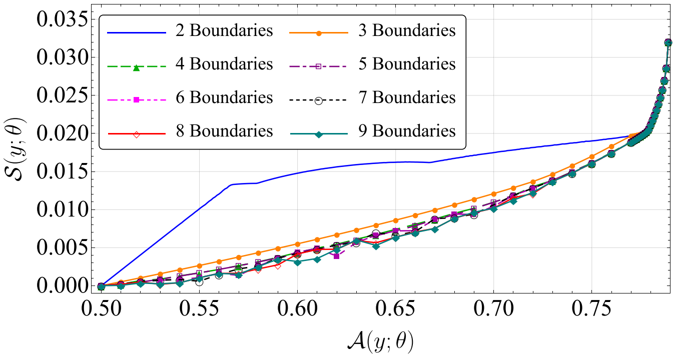

Fig. 3. Accuracy-sensitivity tradeoff curves for general classifiers with different number of boundaries for the Gaussian hypothesis testing problem. The parameters of the distributions are $\mu_0 = 0$, $\sigma_0 = 9$, $\mu_1 = 9$, $\sigma_1 = 4$.

Next, we numerically analyze the effect of the complexity (determined by the number of boundaries) of the general classifier on the tradeoff. Fig. 3 shows the tradeoff curves corresponding to ∞−norm sensitivity for different number of boundaries. Ideally, the tradeoff should improve as the number of boundaries increase. Interestingly, we observe that, for high values of accuracy (> 0.72), increasing the number of boundaries does not improve the tradeoff, and all curves for $9 \geq n \geq 4$ coincide. For low values of accuracy, we face numerical difficulties in obtaining the global minimum of (15), and therefore, we do not observe smooth and ordered points on the curve. However, we still observe that the curves are close to each other, and the tradeoff does not seem to improve beyond a certain number of boundaries. Based on this, we conjecture that there exists a fundamental tradeoff curve which cannot be improved by increasing the number of boundaries arbitrarily.

IV. Illustrative Examples

In this section we illustrate numerically the implications of Theorem 1. In particular, we consider two classification algorithms with different accuracy and sensitivity, and show how their performance degrades differently when the observations

[4] ML and linear classifiers are particular instances of the general classifier.

TABLE I
NUMERICAL RESULTS FOR BINARY CLASSIFICATION

Classifier	y_1	y_2	$\mathcal{S}(y;\theta)$	$\mathcal{A}(y;\theta)$	$\mathcal{A}_{\text{adv1}}$	$\mathcal{A}_{\text{adv2}}$
$\mathfrak{C}^1$	3.65	18.78	0.0334	0.7891	0.6857	0.6808
$\mathfrak{C}^2$	1.83	20.60	0.0201	0.7766	0.6947	0.6939

are corrupted by an adversary. This implies that, when robustness to adversarial manipulation of the observations is a concern, classification algorithms should be designed to simultaneously optimize accuracy and sensitivity, and should not operate at their point of maximum accuracy.

Consider the classification problem (1), and let

$$f_0(x,\theta_0) = \mathcal{N}(x;\mu_0,\sigma_0), \quad f_1(x,\theta_1) = \mathcal{N}(x;\mu_1,\sigma_1). \tag{17}$$

Let $\mathfrak{C}^1 = \mathfrak{C}_{\text{ML}}(x;1)$ and $\mathfrak{C}^2 = \mathfrak{C}_{\text{ML}}(x;0.46)$ be the classification algorithms identified by the red and green points in Fig. 2(a), respectively. Notice that, when the observations are not manipulated and follow the distributions (17), $\mathfrak{C}^1$ achieves higher accuracy and sensitivity than $\mathfrak{C}^2$. This is also the case when using definition (16), as illustrated in Fig. 2(b). While the nominal distributions (17) are used to design the classifiers $\mathfrak{C}^1$ and $\mathfrak{C}^2$, we consider an adversary that manipulates the observations so that their true distributions are

$$\begin{aligned} f_0(x,\theta_0) &= \mathcal{N}(x;\mu_0+\bar{\mu}_0,\sigma_0+\bar{\sigma}_0), \text{ and} \\ f_1(x,\theta_1) &= \mathcal{N}(x;\mu_1+\bar{\mu}_1,\sigma_1+\bar{\sigma}_1), \end{aligned} \tag{18}$$

where $\bar{\mu}_0$, $\bar{\mu}_1$, $\bar{\sigma}_0$, and $\bar{\sigma}_1$ are unknown parameters selected by the adversary to deteriorate the accuracy of the classifiers.

To evaluate the accuracy of $\mathfrak{C}^1$ and $\mathfrak{C}^2$, we generate 10000 observations obeying the modified distributions (18), and compute the accuracy of the classifiers as the ratio of the number of correct predictions to the total number of observations. We repeat this experiment 100 times, and then compute the average accuracy of the classifiers over all trials.

Table I summarizes the results of the classification problems with $\mathfrak{C}^1$ and $\mathfrak{C}^2$ on the altered observations. In particular, y_1 and y_2 are the decision boundaries of the classifiers, while $\mathcal{S}(y;\theta)$ and $\mathcal{A}(y;\theta)$ denote their nominal sensitivity and accuracy. Instead, $\mathcal{A}_{\text{adv1}}$ and $\mathcal{A}_{\text{adv2}}$ denote the average accuracy of the classifiers when, respectively, the adversarial parameters are $\bar{\mu}_1 = \bar{\mu}_0 = \bar{\sigma}_0 = 0$, $\bar{\sigma}_1 = 3$, and $\bar{\mu}_0 = 1$, $\bar{\sigma}_0 = 2$, $\bar{\mu}_1 = -2$, $\bar{\sigma}_1 = 1.5$. The results show that, although $\mathfrak{C}^1$ exhibits higher accuracy than $\mathfrak{C}^2$ when the observations follow the nominal distributions (17), $\mathfrak{C}^2$ outperforms $\mathfrak{C}^1$ in both adversarial scenarios, as supported by our analysis.

Next, we illustrate that the results of Theorem 1 can be observed for more complex and multidimensional classification problems. We consider the classification of hand-written digits (0-9) using a neural network (NN). We consider a NN with 6 layers, which uses cross entropy loss function, and we use the MNIST dataset [15] for its training. We add a regularization term to the loss function to increase the robustness of the NN against adversarial perturbations. We train 4 NNs using unperturbed images - NN1 without any regularization term, and NN2, NN3 and NN4 with increasing regularization weight coefficients. The adversarial images are computed using the framework of [4]. The results are reported in Table II, where $\mathcal{A}_{\text{nom}}$ and $\mathcal{A}_{\text{adv}}$ denote the accuracy of a NN under clean and adversarial images, respectively. We observe that a NN with larger robustness ($\mathcal{A}_{\text{adv}}$) exhibits lower accuracy ($\mathcal{A}_{\text{nom}}$), indicating the existence of an accuracy-sensitivity tradeoff.

TABLE II
NUMERICAL RESULTS FOR DIGIT CLASSIFICATION

Neural Networks	NN1	NN2	NN3	NN4
$\mathcal{A}_{\text{nom}}$	0.9828	0.9641	0.9170	0.8665
$\mathcal{A}_{\text{adv}}$	0.2462	0.2734	0.3189	0.3204

V. CONCLUSION AND FUTURE WORK

In this letter we show that a fundamental tradeoff exists between the accuracy of a binary classification algorithm and its sensitivity to adversarial manipulation of the data. Thus, accuracy can only be maximized at the expenses of the sensitivity to data manipulation, and this tradeoff cannot be arbitrarily improved by tuning the algorithm's parameters. Directions of future interest include the extension to M-ary testing problems, as well as the formal characterization of the relationships between the complexity of the classification algorithm and its accuracy versus sensitivity tradeoff.

REFERENCES

[1] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 2, pp. 253–279, May 2019.

[2] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst. Mag.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.

[3] P. Zhu, J. Isaacs, B. Fu, and S. Ferrari, "Deep learning feature extraction for target recognition and classification in underwater sonar images," in *Proc. IEEE Conf. Decis. Control*, Melbourne, VIC, Australia, Dec. 2017, pp. 2724–2731.

[4] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent.*, Banff, AB, Canada, Apr. 2014. [Online]. Available: https://arxiv.org/abs/1312.6199

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015. [Online]. Available: http://arxiv.org/abs/1412.6572

[6] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. Eur. Symp. Security Privacy*, Saarbrucken, Germany, Mar. 2016, pp. 372–387.

[7] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017. [Online]. Available: https://openreview.net/forum?id=BJm4T4Kgx

[8] D. Lowd and C. Meek, "Adversarial learning," in *Proc. Int. Conf. Knowl. Disc. Data Min.*, Chicago, IL, USA, Aug. 2005, pp. 641–647.

[9] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 2574–2582.

[10] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, May 2018. [Online]. Available: https://openreview.net/forum?id=Bys4ob-Rb

[11] A. Sinha, H. Namkoong, and J. Duchi, "Certifying some distributional robustness with principled adversarial training," in *Proc. Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, May 2018. [Online]. Available: https://openreview.net/forum?id=Hk6kPgZA-

[12] T. A. Schonhoff and A. A. Giordano, *Detection and Estimation Theory and Its Applications*. Upper Saddle River, NJ, USA: Pearson College Division, 2006.

[13] G.-X. Yuan, C.-H. Ho, and C.-J. Lin, "Recent advances of large-scale linear classification," *Proc. IEEE*, vol. 100, no. 9, pp. 2584–2603, Sep. 2012.

[14] A. Ghafouri, Y. Vorobeychik, and X. Koutsoukos, "Adversarial regression for detecting attacks in cyber-physical systems," in *Proc. Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, Jul. 2018, pp. 3769–3775.

[15] Y. LeCun, C. Cortes, and C. J. C. Burges. (1998). *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist