On Security and Sparsity of Linear Classifiers for Adversarial Settings

Ambra Demontis, Paolo Russu, Battista Biggio,
Giorgio Fumera, and Fabio Roli

Department of Electrical and Electronic Engineering,
University of Cagliari, Piazza d'Armi 09123, Cagliari, Italy
{ambra.demontis,paolo.russu,battista.biggio,fumera,roli}@diee.unica.it

Abstract. Machine-learning techniques are widely used in security-related applications, like spam and malware detection. However, in such settings, they have been shown to be vulnerable to *adversarial* attacks, including the deliberate manipulation of data at test time to evade detection. In this work, we focus on the vulnerability of linear classifiers to evasion attacks. This can be considered a relevant problem, as linear classifiers have been increasingly used in embedded systems and mobile devices for their low processing time and memory requirements. We exploit recent findings in robust optimization to investigate the link between regularization and security of linear classifiers, depending on the type of attack. We also analyze the relationship between the sparsity of feature weights, which is desirable for reducing processing cost, and the security of linear classifiers. We further propose a novel octagonal regularizer that allows us to achieve a proper trade-off between them. Finally, we empirically show how this regularizer can improve classifier security and sparsity in real-world application examples including spam and malware detection.

1 Introduction

Machine-learning techniques are becoming an essential tool in several application fields such as marketing, economy and medicine. They are increasingly being used also in security-related applications, like spam and malware detection, despite their vulnerability to *adversarial* attacks, *i.e.*, the *deliberate* manipulation of training or test data, to subvert their operation; *e.g.*, spam emails can be manipulated (at test time) to evade a trained anti-spam classifier [1–12].

In this work, we focus on the security of linear classifiers. These classifiers are particularly suited to mobile and embedded systems, as the latter usually demand for strict constraints on storage, processing time and power consumption. Nonetheless, linear classifiers are also a preferred choice as they provide easier-to-interpret decisions (with respect to nonlinear classification methods). For instance, the widely-used SpamAssassin anti-spam filter exploits a linear classifier [7, 5].[1] Work in the adversarial machine learning literature has already

---

[1] See also http://spamassassin.apache.org.

investigated the security of linear classifiers to evasion attacks [4, 7], suggesting the use of more evenly-distributed feature weights as a mean to improve their security. Such a solution is however based on heuristic criteria, and a clear understanding of the conditions under which it can be effective, or even optimal, is still lacking. Moreover, in mobile and embedded systems, *sparse* weights are more desirable than evenly-distributed ones, in terms of processing time, memory requirements, and interpretability of decisions.

In this work, we shed some light on the security of linear classifiers, leveraging recent findings from [13–15] that highlight the relationship between classifier regularization and robust optimization problems in which the input data is potentially corrupted by noise (see Sect. 2). This is particularly relevant in adversarial settings as the aforementioned ones, since evasion attacks can be essentially considered a form of noise affecting the non-manipulated, initial data (*e.g.*, malicious code before obfuscation). Connecting the work in [13–15] to adversarial machine learning aims to help understanding what the optimal regularizer is against different kinds of adversarial *noise* (attacks). We analyze the relationship between the sparsity of the weights of a linear classifier and its security in Sect. 3, where we also propose an octagonal-norm regularizer to better tune the trade-off arising between sparsity and security. In Sect. 4, we empirically evaluate our results on a handwritten digit recognition task, and on real-world application examples including spam filtering and detection of malicious software (malware) in PDF files. We conclude by discussing the main contributions and findings of our work in Sect. 5, while also sketching some promising research directions.

2   Background

In this section, we summarize the attacker model previously proposed in [8–11], and the link between regularization and robustness discussed in [13–15].

2.1   Attacker's Model

To rigorously analyze possible attacks against machine learning and devise principled countermeasures, a formal model of the attacker has been proposed in [6, 8–11], based on the definition of her goal (*e.g.*, evading detection at test time), knowledge of the classifier, and capability of manipulating the input data.

Attacker's Goal. Among the possible goals, here we focus on evasion attacks, where the goal is to modify a single malicious sample (*e.g.*, a spam email) to have it misclassified as legitimate (with the largest confidence) by the classifier [8].

Attacker's Knowledge. The attacker can have different levels of knowledge of the targeted classifier; she may have *limited* or *perfect* knowledge about the training data, the feature set, and the classification algorithm [8, 9]. In this work, we focus on *perfect-knowledge* (worst-case) attacks.

Attacker's Capability. In evasion attacks, the attacker is only able to modify malicious instances. Modifying an instance usually has some cost. Moreover, arbitrary modifications to evade the classifier may be ineffective, if the resulting

instance loses its malicious nature (*e.g.*, excessive obfuscation of a spam email could make it unreadable for humans). This can be formalized by an application-dependent constraint. As discussed in [16], two kinds of constraints have been mostly used when modeling real-world adversarial settings, leading one to define *sparse* (ℓ_1) and *dense* (ℓ_2) attacks. The ℓ_1-**norm** yields typically a sparse attack, as it represents the case when the cost depends on the number of modified features. For instance, when instances correspond to text (*e.g.*, the email's body) and each feature represents the occurrences of a given term in the text, the attacker usually aims to change as few words as possible. The ℓ_2-**norm** yields a dense attack, as it represents the case when the cost of modifying features is proportional to the distance between the original and modified sample in Euclidean space. For example, if instances are images, the attacker may prefer making small changes to many or even all pixels, rather than significantly modifying only few of them. This amounts to (slightly) blurring the image, instead of obtaining a salt-and-pepper noise effect (as the one produced by sparse attacks) [16].

Attack Strategy. It consists of the procedure for modifying samples, according to the attacker's goal, knowledge and capability, formalized as an optimization problem. Let us denote the legitimate and malicious class labels respectively with -1 and $+1$, and assume that the classifier's decision function is $f(\mathbf{x}) = \text{sign}\,(g(\mathbf{x}))$, where $g(\mathbf{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b \in \mathbb{R}$ is a linear discriminant function with feature weights $\boldsymbol{w} \in \mathbb{R}^{\mathsf{d}}$ and bias $b \in \mathbb{R}$, and $\mathbf{x}$ is the representation of an instance in a d-dimensional feature space. Given a malicious sample $\boldsymbol{x}_0$, the goal is to find the sample $\boldsymbol{x}^*$ that minimizes the classifier's discriminant function $g(\cdot)$ (*i.e.*, that is classified as legitimate with the highest possible confidence) subject to the constraint that $\boldsymbol{x}^*$ lies within a distance $d_{\max}$ from $\boldsymbol{x}_0$:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} g(\boldsymbol{x}) \tag{1}$$

$$\text{s.t.}\ \ d(\boldsymbol{x}, \boldsymbol{x}_0) \leq d_{\max}\,, \tag{2}$$

where the distance measure $d(\cdot, \cdot)$ is defined in terms of the cost of data manipulation (*e.g.*, the number of modified words in each spam) [1, 2, 8, 9, 12]. For sparse and dense attacks, $d(\cdot, \cdot)$ corresponds respectively to the ℓ_1 and ℓ_2 distance.

2.2   Robustness and Regularization

The goal of this section is to clarify the connection between regularization and input data uncertainty, leveraging on the recent findings in [13–15]. In particular, Xu *et al.* [13] have considered the following *robust* optimization problem:

$$\min_{\boldsymbol{w},b}\ \max_{\boldsymbol{u}_1,\dots,\boldsymbol{u}_{\mathsf{m}} \in \mathcal{U}}\ \sum_{i=1}^{\mathsf{m}} \left(1 - y_i(\boldsymbol{w}^\top(\boldsymbol{x}_i - \boldsymbol{u}_i) + b)\right)_+ \,, \tag{3}$$

where $(z)_+$ is equal to $z \in \mathbb{R}$ if $z > 0$ and 0 otherwise, $\boldsymbol{u}_1, ..., \boldsymbol{u}_{\mathsf{m}} \in \mathcal{U}$ define a set of bounded perturbations of the training data $\{\boldsymbol{x}_i, y_i\}_{i=1}^{\mathsf{m}} \in \mathbb{R}^{\mathsf{m}} \times \{-1, +1\}^{\mathsf{m}}$, and the so-called *uncertainty set* $\mathcal{U}$ is defined as $\mathcal{U} \stackrel{\Delta}{=} \big\{(\boldsymbol{u}_1, \dots, \boldsymbol{u}_{\mathsf{m}}) \,|\, \sum_{i=1}^{\mathsf{m}} \|\boldsymbol{u}_i\|^* \leq c\big\},$
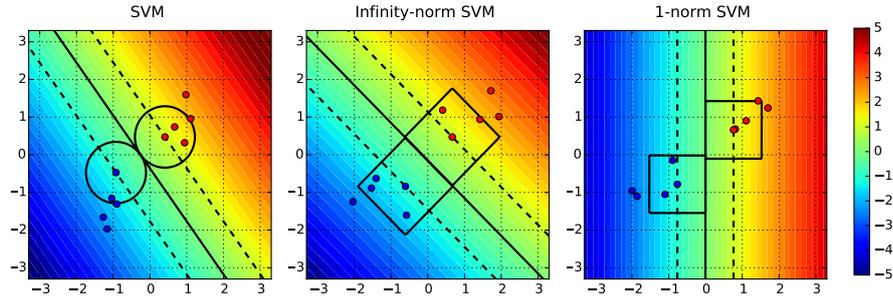
Fig. 1. Discriminant function $g(\boldsymbol{x})$ for SVM, Infinity-norm SVM, and 1-norm SVM (in colors). The decision boundary ($g(\boldsymbol{x}) = 0$) and margins ($g(\boldsymbol{x}) = \pm 1$) are respectively shown with black solid and dashed lines. Uncertainty sets are drawn over the support vectors to show how they determine the orientation of the decision boundary.

being $\|\cdot\|^*$ the dual norm of $\|\cdot\|$. Typical examples of uncertainty sets according to the above definition include ℓ_1 and ℓ_2 balls [13, 14].

Problem (3) basically corresponds to minimizing the hinge loss for a two-class classification problem under worst-case, bounded perturbations of the training samples $\boldsymbol{x}_i$, *i.e.*, a typical setting in robust optimization [13–15]. Under some mild assumptions easily verified in practice (including non-separability of the training data), the authors have shown that the above problem is equivalent to the following non-robust, regularized optimization problem (*cf.* Th. 3 in [13]):

$$\min_{\boldsymbol{w}, b} \; c\|\boldsymbol{w}\| + \sum_{i=1}^{m} \left(1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)\right)_+ . \qquad (4)$$

This means that, if the ℓ_2 norm is chosen as the dual norm characterizing the uncertainty set $\mathcal{U}$, then $\boldsymbol{w}$ is regularized with the ℓ_2 norm, and the above problem is equivalent to a standard Support Vector Machine (SVM) [17]. If input data uncertainty is modeled with the ℓ_1 norm, instead, the optimal regularizer would be the ℓ_∞ regularizer, and vice-versa.[2] This notion is clarified in Fig. 1, where we consider different norms to model input data uncertainty against the corresponding SVMs; *i.e.*, the standard SVM [17], the Infinity-norm SVM [18] and the 1-norm SVM [19] against ℓ_2, ℓ_1 and ℓ_∞-norm uncertainty models, respectively.

3 Security and Sparsity

We discuss here the main contributions of this work. The result discussed in the previous section, similar to that reported independently in [15], helps understanding the security properties of linear classifiers in adversarial settings, in terms of the relationship between security and sparsity. In fact, what discussed in

---

[2] Note that the ℓ_1 norm is the dual norm of the ℓ_∞ norm, and vice-versa, while the ℓ_2 norm is the dual norm of itself.

the previous section does not only confirm the intuition in [4, 7], *i.e.*, that more uniform feature weighting schemes should improve classifier security by enforcing the attacker to manipulate more feature values to evade detection. The result in [13–15] also clarifies the meaning of *uniformity* of the feature weights $\boldsymbol{w}$. If one considers an ℓ_1 (sparse) attacker, facing a higher cost when modifying more features, it turns out that the optimal regularizer is given by the ℓ_∞ norm of $\boldsymbol{w}$, which tends to yield more uniform weights. In particular, the solution provided by ℓ_∞ regularization (in the presence of a strongly-regularized classifier) tends to yield weights which, in absolute value, are all equal to a (small) maximum value. This also implies that ℓ_∞ regularization does not provide a *sparse* solution.

For this reason we propose a novel *octagonal* (8gon) regularizer,[3] given as a linear (convex) combination of ℓ_1 and ℓ_∞ regularization:

$$\|\boldsymbol{w}\|_{8\text{gon}} = (1 - \rho)\|\boldsymbol{w}\|_1 + \rho\|\boldsymbol{w}\|_\infty \tag{5}$$

where $\rho \in (0, 1)$ can be increased to trade sparsity for security.

Our work does not only aim to clarify the relationships among regularization, sparsity, and *adversarial* noise. We also aim to quantitatively assess the afore-mentioned trade-off on real-world application examples, to evaluate whether and to what the extent the choice of a proper regularizer may have a significant impact in practice. Thus, besides proposing a new regularizer and shedding light on uniform feature weighting and classifier security, the other main contribution of the present work is the experimental analysis reported in the next section, in which we consider both dense (ℓ_2) and sparse (ℓ_1) attacks, and evaluate their impact on SVMs using different regularizers. We further analyze the weight distribution of each classifier to provide a better understanding on how sparsity is related to classifier security under the considered evasion attacks.

4 Experimental Analysis

We first consider dense and sparse attacks in the context of handwritten digit recognition, to visually demonstrate their blurring and salt-and-pepper effect on images. We then consider two real-world application examples including spam and PDF malware detection, investigating the behavior of different regularization terms against (*sparse*) evasion attacks.

Handwritten Digit Classification. To visually show how evasion attacks work, we perform sparse and dense attacks on the MNIST digit data [21]. Each image is represented by a vector of 784 features, corresponding to its gray-level pixel values. As in [8], we simulate an adversarial classification problem where the digits 8 and 9 correspond to the legitimate and malicious class, respectively.

Spam Filtering. This is a well-known application subject to adversarial attacks. Most spam filters include an automatic text classifier that analyzes the email's

---

[3] Note that octagonal regularization has been previously proposed also in [20]. However, differently from our work, the authors have used a pairwise version of the infinity norm, for the purpose of selecting (correlated) groups of features.

(a) ℓ_2 (b) ℓ_∞ (c) ℓ_1 (d) elastic net   (e) octagonal
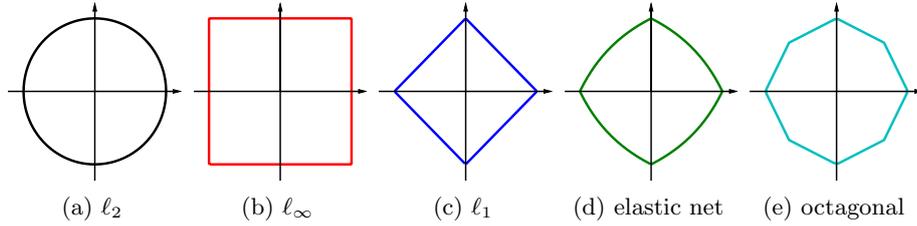
Fig. 2. Unit balls for different norms.

body text. In the simplest case Boolean features are used, each representing the presence or absence of a given term. For our experiments we use the TREC 2007 spam track data, consisting of about 25000 legitimate and 50000 spam emails [22]. We extract a dictionary of terms (features) from the first 5000 emails (in chronological order) using the same parsing mechanism of SpamAssassin, and then select the 200 most discriminant features according to the information gain criterion [23]. We simulate a well-known (*sparse*) evasion attack in which the attacker aims to modify only few terms. Adding or removing a term amounts to switching the value of the corresponding Boolean feature [3, 4, 8, 9, 12].

PDF Malware Detection. Another application that is often targeted by attackers is the detection of malware in PDF files. A PDF file can host different kinds of contents, like Flash and JavaScript. Such third-party applications can be exploited by attacker to execute arbitrary operations. We use a data set made up of about 5500 legitimate and 6000 malicious PDF files. We represent every file using the 114 features that are described in [24]. They consist of the number of occurrences of a predefined set of keywords, where every keyword represents an action performed by one of the objects that are contained into the PDF file (*e.g.*, opening another document that is stored inside the file). An attacker cannot trivially remove keywords from a PDF file without corrupting its functionality. Conversely, she can easily add new keywords by inserting new object's operations. For this reason, we simulate this attack by only considering feature increments (decrementing a feature value is not allowed). Accordingly, the most convenient strategy to mislead a malware detector (classifier) is thus to insert as many occurrences of a *given* keyword as possible, which is a sparse attack.[4]

We consider different versions of the SVM classifier obtained by combining the hinge loss with the different regularizers shown in Fig. 2.

2-norm SVM (SVM). This is the standard SVM learning algorithm [17]. It finds $\boldsymbol{w}$ and b by solving the following quadratic programming problem:

$$\min_{\boldsymbol{w}, b} \ \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{m}\left(1 - y_i g(\boldsymbol{x}_i)\right)_+ \ , \tag{6}$$

---

[4] Despite no upper bound on the number of injected keywords may be set, we set the maximum value for each keyword to the corresponding one observed during training.

where $g(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ denotes the SVM's linear discriminant function. Note that ℓ_2 regularization does not induce sparsity on $\boldsymbol{w}$.

Infinity-norm SVM (∞-norm). In this case, the ℓ_∞ regularizer bounds the weights' maximum absolute value as $\|\boldsymbol{w}\|_\infty = \max_{j=1,\dots,\mathsf{d}} |w_j|$ [20]:

$$\min_{\boldsymbol{w},b} \ \|\boldsymbol{w}\|_\infty + C \sum_{i=1}^{\mathsf{m}} (1 - y_i g(\boldsymbol{x}_i))_+ \ . \tag{7}$$

As the standard SVM, this classifier is not sparse; however, the above learning problem can be solved using a simple linear programming approach.

1-Norm SVM (1-norm). Its learning algorithm is defined as [19]:

$$\min_{\boldsymbol{w},b} \ \|\boldsymbol{w}\|_1 + C \sum_{i=1}^{\mathsf{m}} (1 - y_i g(\boldsymbol{x}_i))_+ \ . \tag{8}$$

The ℓ_1 regularizer induces sparsity, while retaining convexity and linearity.

Elastic-net SVM (el-net). We use here the elastic-net regularizer [25], combined with the hinge loss to obtain an SVM formulation with tunable sparsity:

$$\min_{\boldsymbol{w},b} \ (1-\lambda)\|\boldsymbol{w}\|_1 + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^{\mathsf{m}} (1 - y_i g(\boldsymbol{x}_i))_+ \ . \tag{9}$$

The level of sparsity can be tuned through the trade-off parameter $\lambda \in (0, 1)$.

Octagonal-norm SVM (8gon). This novel SVM is based on our octagonal-norm regularizer, combined with the hinge loss:

$$\min_{\boldsymbol{w},b} \ (1-\rho)\|\boldsymbol{w}\|_1 + \rho\|\boldsymbol{w}\|_\infty + C \sum_{i=1}^{\mathsf{m}} (1 - y_i g(\boldsymbol{x}_i))_+ \ . \tag{10}$$

The above optimization problem is linear, and can be solved using state-of-the-art solvers. The sparsity of $\boldsymbol{w}$ can be increased by decreasing the trade-off parameter $\rho \in (0, 1)$, at the expense of classifier security.

Sparsity and Security Measures. We evaluate the degree of sparsity S of a given linear classifier as the fraction of its weights that are equal to zero:

$$S = \frac{1}{\mathsf{d}} |\{w_j | w_j = 0, j = 1, \dots, \mathsf{d}\}| , \tag{11}$$

being $|\cdot|$ the cardinality of the set of null weights.

To evaluate security of linear classifiers, we define a measure E of *weight evenness*, similarly to [4, 7], based on the ratio of the ℓ_1 and ℓ_∞ norm:

$$E = \frac{\|\boldsymbol{w}\|_1}{\mathsf{d}\|\boldsymbol{w}\|_\infty} \ , \tag{12}$$

where dividing by the number of features d ensures that $E \in \left[\frac{1}{\mathsf{d}}, 1\right]$, with higher values denoting more evenly-distributed feature weights. In particular, if only a
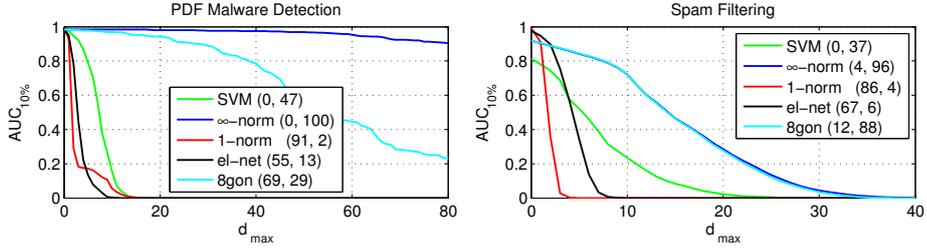
Fig. 3. Classifier performance under attack for PDF malware and spam data, measured in terms of $AUC_{10\%}$ against an increasing number d_{max} of modified features. For each classifier, we also report (S, E) percentage values (Eqs. 11-12) in the legend.

weight is not zero, then $E = \frac{1}{d}$; conversely, when all weights are equal to the maximum (in absolute value), $E = 1$.

Experimental Setup. We randomly select 500 legitimate and 500 malicious samples from each dataset, and equally subdivide them to create a training and a test set. We optimize the regularization parameter C of each SVM (along with λ and ρ for the Elastic-net and Octagonal SVMs, respectively) through 5-fold cross-validation, maximizing the following objective on the training data:

$$AUC + \alpha E + \beta S \qquad (13)$$

where AUC is the area under the ROC curve, and α and β are parameters defining the trade-off between security and sparsity. We set $\alpha = \beta = 0.1$ for the PDF and digit data, and $\alpha = 0.2$ and $\beta = 0.1$ for the spam data, to promote more secure solutions in the latter case. These parameters allow us to accept a marginal decrease in classifier security only if it corresponds to much sparser feature weights. After classifier training, we perform evasion attacks on all malicious test samples, and evaluate the corresponding performance as a function of the number of features modified by the attacker. We repeat this procedure five times, and report the average results on the original and modified test data.

Experimental Results. We consider first PDF malware and spam detection. In these applications, as mentioned before, only sparse evasion attacks make sense, as the attacker aims to minimize the number of modified features. In Fig. 3, we report the AUC at 10% false positive rate for the considered classifiers, against an increasing number of words/keywords changed by the attacker. This experiment shows that the most secure classifier under sparse evasion attacks is the Infinity-norm SVM, since its performance degrades more gracefully under attack. This is an expected result, given that, in this case, infinity-norm regularization corresponds to the dual norm of the attacker's cost/distance function. Notably, the Octagonal SVM yields reasonable security levels while achieving much sparser solutions, as expected (*cf.* the sparsity values S in the legend of Fig. 3). This experiment really clarifies how much the choice of a proper regularizer can be crucial in real-world adversarial applications.
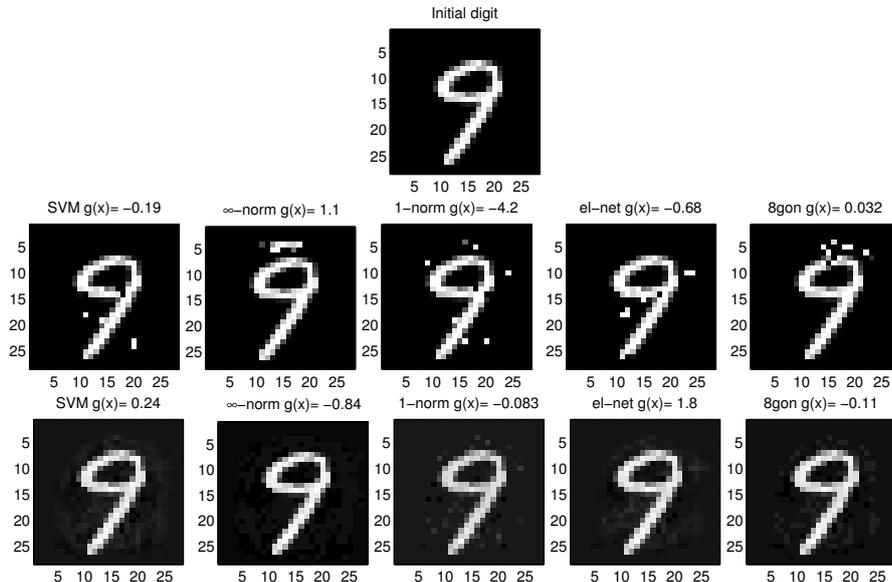
Fig. 4. Initial digit "9" (*first row*) and its versions modified to be misclassified as "8" (*second and third row*). Each column corresponds to a different classifier (from *left* to *right* in the second and third row): SVM, Infinity-norm SVM, 1-norm SVM, Elastic-net SVM, Octagonal SVM. *Second row*: sparse attacks (ℓ_1), with $d_{max} = 2000$. *Third row*: dense attacks (ℓ_2), with $d_{max} = 250$. Values of $g(\boldsymbol{x}) < 0$ denote a successful classifier evasion (*i.e.*, more vulnerable classifiers).

By looking at the values reported in Fig. 3, it may seem that the security measure E does not properly characterize classifier security under attack; *e.g.*, note how Octagonal SVM exhibits lower values of E despite being more secure than SVM on the PDF data. The underlying reason is that the attack implemented on the PDF data only considers feature increments, while E generically considers any kind of manipulation. Accordingly, one should define alternative security measures depending on specific kinds of data manipulation. However, the security measure E allows us to properly tune the trade-off between security and sparsity also in this case and, thus, this issue may be considered negligible.

Finally, to visually demonstrate the effect of sparse and dense evasion attacks, we report some results on the MNIST handwritten digits. In Fig. 4, we show the "9" digit image modified by the attacker to have it misclassified by the classifier as an "8". These modified digits are obtained by solving Problem (1)-(2) through a simple projected gradient-descent algorithm, as in [8].[5] Note how dense attacks only produce a slightly-blurred effect on the image, while sparse attacks create

---

[5] Note also that, in the case of sparse attacks, Problem (1)-(2) amounts to minimizing a linear function subject to linear constraints. It can be thus formulated as a linear programming problem, and solved with state-of-the-art linear solvers.

more evident visual artifacts. By comparing the values of $g(\boldsymbol{x})$ reported in Fig. 4, one may also note that this simple example confirms again that Infinity-norm and Octagonal SVM are more secure against sparse attacks, while SVM and Elastic-net SVM are more secure against dense attacks.

5   Conclusions and Future Work

In this work we have shed light on the theoretical and practical implications of sparsity and security in linear classifiers. We have shown on real-world adversarial applications that the choice of a proper regularizer is crucial. In fact, in the presence of sparse attacks, Infinity-norm SVMs can drastically outperform the security of standard SVMs. We believe that this is an important result, as (standard) SVMs are widely used in security tasks without taking the risk of adversarial attacks too much into consideration. Moreover, we propose a new octagonal regularizer that enables trading sparsity for a marginal loss of security under sparse evasion attacks. This is extremely useful in applications where sparsity and computational efficiency at test time are crucial. When dense attacks are instead deemed more likely, the standard SVM may be retained a good compromise. In that case, if sparsity is required, one may trade some level of security for sparsity using the Elastic-net SVM. Finally, we think that an interesting extension of our work may be to investigate the trade-off between sparsity and security also in the context of classifier poisoning (in which the attacker can contaminate the training data to mislead classifier learning) [9, 26, 27].

References

1. Dalvi, N., Domingos, P., Mausam, Sanghai, S., Verma, D.: Adversarial classification. In: 10th Int'l Conf. Knowl. Disc. Data Mining (KDD), Seattle, WA, USA, ACM (2004) 99–108
2. Lowd, D., Meek, C.: Adversarial learning. In: 11th Int'l Conf. Knowl. Disc. Data Mining (KDD), Chicago, IL, USA, ACM (2005) 641–647
3. Lowd, D., Meek, C.: Good word attacks on statistical spam filters. In: 2nd Conf. Email and Anti-Spam (CEAS), Mountain View, CA, USA (2005)
4. Kolcz, A., Teo, C.H.: Feature weighting for improved classifier robustness. In: 6th Conf. Email and Anti-Spam (CEAS), Mountain View, CA, USA (2009)
5. Nelson, B., Barreno, M., Chi, F.J., Joseph, A.D., Rubinstein, B.I.P., Saini, U., Sutton, C., Tygar, J.D., Xia, K.: Exploiting machine learning to subvert your spam filter. In: LEET '08, Berkeley, CA, USA, USENIX Association (2008) 1–9
6. Barreno, M., Nelson, B., Sears, R., Joseph, A.D., Tygar, J.D.: Can machine learning be secure? In: ASIACCS '06, New York, NY, USA, ACM (2006) 16–25
7. Biggio, B., Fumera, G., Roli, F.: Multiple classifier systems for robust classifier design in adversarial environments. Int'l J. Mach. Learn. Cyb. **1**(1) (2010) 27–41
8. Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., Roli, F.: Evasion attacks against machine learning at test time. In Blockeel, H. et al., eds.: ECML-PKDD, Part III, vol. 8190, LNCS, Springer (2013) 387–402
9. Biggio, B., Fumera, G., Roli, F.: Security evaluation of pattern classifiers under attack. IEEE Trans. on Knowl. and Data Eng. **26**(4) (April 2014) 984–996

10. Biggio, B., Fumera, G., Roli, F.: Pattern recognition systems under attack: Design issues and research challenges. Int'l J. Patt. Rec. Artif. Intell. **28**(7) (2014) 1460002

11. Huang, L., Joseph, A.D., Nelson, B., Rubinstein, B., Tygar, J.D.: Adversarial machine learning. In: 4th ACM Workshop Artif. Intell. and Sec. (AISec), Chicago, IL, USA (2011) 43–57

12. Zhang, F., Chan, P., Biggio, B., Yeung, D., Roli, F.: Adversarial feature selection against evasion attacks. IEEE Trans. Cyb. **46**(3) (2016) 766–777

13. Xu, H., Caramanis, C., Mannor, S.: Robustness and regularization of support vector machines. J. Mach. Learn. Res. **10** (July 2009) 1485–1510

14. Sra, S., Nowozin, S., Wright, S.J.: Optimization for Machine Learning. The MIT Press (2011)

15. Livni, R., Crammer, K., Globerson, A., Edmond, E.i., Safra, L.: A simple geometric interpretation of SVM using stochastic adversaries. In: JMLR W&CP - vol. 22 of AISTATS '12. (2012) 722–730

16. Wang, F., Liu, W., Chawla, S.: On sparse feature attacks in adversarial learning. In: IEEE Int'l Conf. on Data Mining (ICDM), IEEE (2014) 1013–1018

17. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20** (1995) 273–297

18. Bennett, K. P., Bredensteiner, E. J.: Duality and geometry in SVM classifiers. In: 17th ICML, Morgan Kaufmann Publishers Inc. (2000) 57–64

19. Zhu, J., Rosset, S., Tibshirani, R., Hastie, T. J.: 1-norm support vector machines. In Thrun, S., Saul, L., Schölkopf, B., eds.: NIPS 16, MIT Press (2004) 49–56.

20. Bondell, Reich: Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. Biometrics **64** (2007) 115–123

21. LeCun, Y. et al.: Comparison of learning algorithms for handwritten digit recognition. In: Int'l Conf. Artif. Neural Networks. (1995) 53–60

22. Cormack, G.V.: TREC 2007 spam track overview. In Voorhees, E.M., Buckland, L.P., eds.: TREC. Volume Special Publication 500-274., NIST (2007)

23. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34** (March 2002) 1–47

24. Maiorca, D., Giacinto, G., Corona, I.: A pattern recognition system for malicious PDF files detection. In Perner, P., ed.: Mach. Learn. and Data Mining in Patt. Rec. vol. 7376, LNCS, Springer Berlin Heidelberg (2012) 510–524

25. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, Series B **67**(2) (2005) 301–320

26. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against SVMs. In Langford, J., Pineau, J., eds.: 29th ICML, Omnipress (2012) 1807–1814

27. Xiao, H., Biggio, B., Brown, G., Fumera, G., Eckert, C., Roli, F.: Is feature selection secure against training data poisoning? In Bach, F., Blei, D., eds.: 32nd ICML, vol. 37, JMLR W&CP (2015) 1689–1698