Bounding an Attack's Complexity for a Simple Learning Model

Blaine Nelson Anthony D. Joseph
Computer Science Division
University of California, Berkeley
{nelsonb,adj}@cs.berkeley.edu

ABSTRACT

As machine learning becomes more prevalent as a systems and networking analysis and detection tool, it is becoming an attractive target for attackers who seek to manipulate the system. We examine a naive model for assessing the effectiveness of classifiers against threats poised by adversaries determined to subvert the learner by inserting data designed for this purpose. Based on this model, we analyze the attack in detail, develop bounds on the adversary's capability, and discuss the implications for the security of learning-based detection systems.

1. INTRODUCTION

Machine learning is becoming more prevalent in the systems and networking domain as an analysis and detection tool. This trend has emerged in several fields including spam filtering [8, 11], virus detection [13, 14], and adaptive intrusion detection [7]. However, deployments of learning techniques in security-sensitive systems and networking applications raises the question of whether or not these learners could be misled by a cleverly crafted attack. In such environments, there is a threat of malicious users subverting a learning mechanism in order to compromise or disrupt a service or an entire system.

Machine learning techniques are being deployed in diverse settings in system applications that present a variety of potential security vulnerabilities ranging from denial-of-service attacks to intrusions, or even privacy breaches. In [1], we laid out a broad set of characteristics of the security concerns we foresee as problems. This paper elaborates on the analytic approach presented as an example in that work and presents the details originally approached in [9]. We prove the bounds on the adversary and discuss the implications and limitations of our analysis.

This paper examines the effects of an adversary on a simple model of outlier detection. We begin with a discussion of related work in Section 2. In Section 3, we formalize the setting and model as well as the adversary's strategy. An analysis in Section 4 derives bounds on this adversary. Finally, in Section 5, we discuss the implications of this analysis and how it could be extended in future work.

2. RELATED WORK

The earliest theoretical work we know of that approaches learning in the presence of an adversary was done by Kearns and Li [4]. They worked in the context of Valiant's Probably Approximately Correct (PAC) learning framework [15], extending it to prove bounds for maliciously chosen errors in the training data. They showed that if the learner is to achieve an ϵ classification error, in general, the fraction of training points controlled by the adversary must be less than $\epsilon/(1+\epsilon)$.

Dalvi et al. examine the learn-adapt-relearn cycle from a game-theoretic point of view [2]. In their model, the learner has a cost for measuring each feature of the data and the adversary has a cost for changing each feature in attack points. If the adversary and learner have complete information about one another and we accept some other assumptions, they find an optimal strategy for the learner to defend against the adversary's adaptations.

Research has also begun to examine the vulnerability of learners to reverse engineering. Lowd and Meek introduce a novel learning problem for adversarial classifier reverse engineering in which an adversary conducts an attack that minimizes a cost function [5]. Under their framework, Lowd and Meek construct algorithms for reverse engineering linear classifiers. Moreover, they build an attack to reverse engineer spam filters [6].

3. PROBLEM SETTING

We present a scenario in which an adversary attempts to manipulate an outlier detector by strategically inserting data to progressively influence a learner until a malicious threat is no longer detected. In Section 3 of Barreno et al., this type of attack is referred to as a *causative targeted integrity* attack [1]. This category describes the nature of the attack — "causative" means the adversary is actively manipulating the learner, "targeted" indicates the adversary has a particular goal, and "integrity" means that the adversary wants outlier points to be misclassified.

We discuss an *outlier detection*[1] technique. Outlier detection is the task of identifying anomalous data and is a widely used paradigm in intrusion detection [7] and virus detection [13, 14]. Outlier detection estimates a classification boundary that partitions the space into

[1] Also called novelty or anomaly detection.

"normal" and "outlier" regions where the normal region is the smallest region such that data generated by the training distribution is likely to be contained in it. Outlier detection is used in settings where anomalous data is difficult to obtain or to characterize making outlier detection generally more robust to novel outliers than other classification techniques.

The simple outlier detection model analyzed in this paper estimates the support of the normal data by a multi-dimensional *hypersphere*. As depicted in Figure 1(a) every point in the hypersphere is classified as normal and those outside the hypersphere are classified as outliers. The radius of the hypersphere is fixed a priori and the training algorithm centers it at the mean of the training data. This training is simply maximum likelihood estimation of the location parameter of a multivariate Gaussian distribution with fixed variance. Moreover, outlier detection under this model is equivalent to thresholding the likelihood of the resulting Gaussian distribution. Our results can easily be generalized to an arbitrary fixed covariance matrix Σ by using the Mahalanobis distance $D_\Sigma(\mathbf{x}) = \sqrt{(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)}$, under which equidistant points form an ellipse around the centroid μ.

To make the hypersphere adaptive, the hypersphere is retrained on new data to reestimate its center. To prevent arbitrary data from being introduced, we employ a conservative retraining strategy that only admits new points to the training set if they are classified as normal; we say the classifier *bootstraps* itself. This learning framework is not meant to represent the state of the art in learning techniques. Gaussian maximum likelihood methods often estimate both the location and scale parameters and Gaussian mixture models use a combination of several Gaussians [3]. Some nonparametric techniques estimate hyperspheres in high dimensions, but typically have a dynamic radius and need not be mean-centered [12]. Finally, other outlier detection techniques such as one-class SVMs and thresholded kernel density estimators do not use hyperspheres at all [3, 10]. Nonetheless, we hope that the analysis of this simple technique will provide a foundation for analyzing more complex models.

3.1 Attack Strategy

The attack we analyze involves an adversary determined to alter our detector to include a specific point G by constructing data to shift the hypersphere toward the target as the hypersphere is retrained. We assume the goal G is initially correctly classified as an anomaly by our algorithm as shown in Figure 1(b). The adversary wants to change the state of our detector to misclassify the target as normal. Before the attack, the hypersphere is centered at $\bar{X}_0$ and it has a fixed radius R. The attack is iterated over the course of $T > 1$ training iterations. At the t-th iteration the mean of the hypersphere is denoted by $\bar{X}_t$.

To obtain a worst-case bound, we conservatively give the adversary complete control: the adversary knows the algorithm, its feature set, and its current state, and all points are attack points. At each iteration, the bootstrapping policy retrains on all points that were classified as normal at any previous iteration. Under this policy, the adversary's optimal strategy is straightforward — as depicted in Figure 1(b) the adversary places points at the location where the line between the mean and G intersects with the boundary. This reduces the attack to a single dimension along this line. Suppose that in the t-th iteration, the adversary strategically places α_t points at the t-th optimal location achieving optimal displacement of the mean toward the adversary's goal, G. The effort of the adversary is measured by M defined as $\sum_{t=1}^{T}\alpha_t$ — the total number of attack points.

3.2 Optimal Attack Displacement

To measure the progress of the adversary, we calculate the displacement of the hypersphere's mean from its initial location caused by a sequence $\{\alpha_t\}$ of attack points. Let M_t be defined as $\sum_{j=1}^{t}\alpha_j$, the cumulative mass at time t. Using these terms, the t-th mean, $\bar{X}_t$ can be defined recursively as

$$\bar{X}_t = \frac{M_{t-1}}{M_t}\bar{X}_{t-1} + \frac{\alpha_t}{M_t}\left(\bar{X}_{t-1} + R\right) \qquad (1)$$

In noting that $M_t = M_{t-1} + \alpha_t$, the t-th mean is a *convex combination* of the previous mean and the optimal attack location. Thus, as the cumulative mass M_{t-1} becomes large, α_t must increase in order to sustain the attack's progress. However, a large α_t causes a large M_t, which makes future progress more difficult. These opposing mechanisms require a trade-off between current progress and the difficulty of future progress.

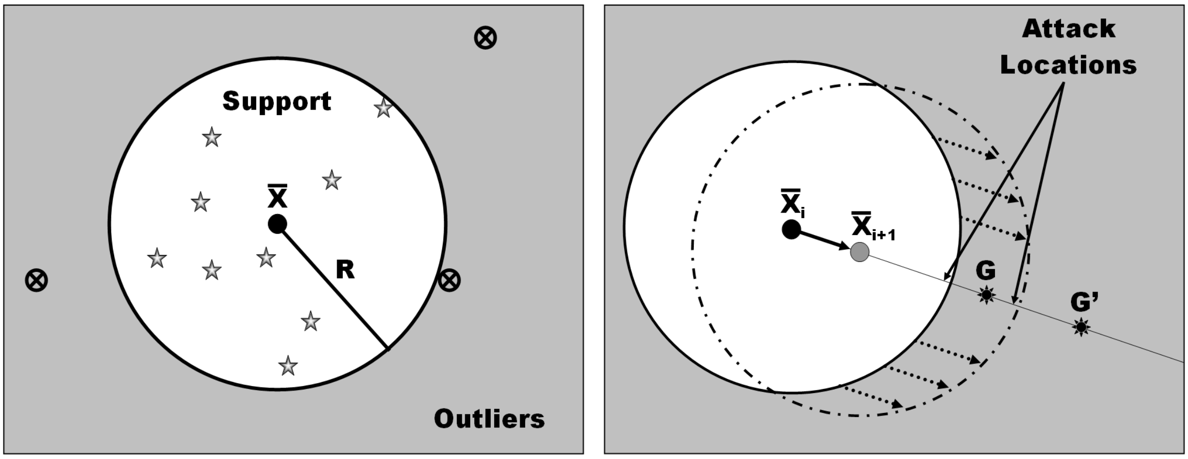
To measure the progress of the entire attack, we calculate the displacement over the course of the attack. For T iterations and a total of M attack points, the function $D_{R,T}(\{M_t\})$ denotes the *relative displacement* caused by the attack sequence — the total displacement relative to the radius of the hypersphere, $\frac{\bar{X}_T - \bar{X}_0}{R}$. Using Equation (1) and the fact that $M_t = M_{t-1} + \alpha_t$, this quantity can be expressed as

$$D_{R,T}(\{M_t\}) = T - \sum_{t=2}^{T}\frac{M_{t-1}}{M_t} \qquad (2)$$

where we constrain $M_1 = 1$ and $M_T = M$ [9]. The relative displacement quantifies the attack's effect in terms of the number of radii by which the hypersphere's mean is displaced. The goal of the adversary is to construct an attack sequence $\{M_t^*\}$ that maximizes Equation (2) with respect to constraints on the size and duration of the attack ($1 = M_1^* \leq M_2^* \leq \ldots \leq M_T^* = M$).

4. BOUNDING THE ADVERSARY

The goal of this analysis is to bound the effort required by the adversary to achieve his desired relative displacement D_R in an attack of duration T. In this section, we derive optimal attack strategies $\{M_t^*\}$ in several variants of the original adversarial setting described



(a) Hypersphere Outlier Detection

(b) Attack on a Hypersphere Outlier Detector

Figure 1: Depictions of the concept of hypersphere outlier detection. In Figure 1(a) a bounding hypersphere centered at $\bar{X}$ of fixed radius R is used to estimate the empirical *support* of a distribution excluding outliers. Samples from the "normal" distribution being modeled are indicated by $\star$ with three outliers indicated by $\otimes$. Figure 1(b) depicts how an adversary with knowledge of the state of the outlier detector can shift the outlier detector toward a first goal G. It could take several iterations of attacks to shift the hypersphere further to include the second goal G'.

in Section 3. These strategies yield an upper bound on the relative displacement achieved by the adversary. The monotonicity in M of these solutions allows us to invert these bounds yielding a lower bound on the minimal effort M^* required by the adversary to succeed.

4.1 Unconstrained Optimal Attack

In the case where the adversary is unconstrained in the duration of the attack, T, a maximum of Equation (2) emerges from a simple analogy. In Equation (1), we saw that the mean could be expressed recursively. Moreover, if we consider each attack point as a unit of mass placed along the line of attack, the mean is simply the center of mass of these points. Finally, the constraint that all attack points must be within the hypersphere corresponds to a balancing analogy: the current attack must "support" the center of mass of all previously placed attack points. As depicted in Figure 2, the attack corresponds to optimally stacking blocks of length $2R$ to extend the maximum distance beyond the edge of a table [9].

The optimal solution to the stacking blocks problem gives the strategy of placing a single attack point at each iteration: $\alpha_t^* = 1$. The optimal distance achieved by this attack corresponds to the harmonic series, $\mathrm{D}^*_{R,T}(M) = \sum_{t=1}^{M} \frac{1}{t}$, which has an upper bound of $\ln(M) + 1$. Since this upper bound is monotonically increasing in $M > 0$, we can invert it to bound M^*,

$$M^* \geq e^{D_R - 1} \tag{3}$$

where D_R is the desired relative displacement of the adversary. This bounds the number of attack points required for the adversary to achieve his goal in an unconstrained setting.

4.2 Optimal Constrained Attack

The adversary's strategy in Section 4.1 is optimal, but it ignores the constraint of expedience — the adversary wants to limit the duration of the attack to $T \ll M$ iterations. This variant of the attack is more plausible in situations where the adversary can generate large numbers of points per iteration and smaller durations are desirable. Moreover, the unconstrained attack is less plausible in scenarios where non-adversarial data would mitigate the adversary's progress.

As was shown in Section 4.1, the original problem is equivalent to the problem of optimally extending a stack of identical blocks over the edge of a table – a reduction to a solved problem. In this setting, constraining the attack to T iterations corresponds to stacking T blocks of length $2R$ with total mass M. However, the blocks have variable (integer-valued) masses. Unlike the unconstrained setting, we are unaware of an optimal integer solution to this problem. However, we can derive an upper bound on Equation (2) by relaxing the constraint that the masses be integers.

In the real-valued setting, analysis of the adversary's optimal strategy can be accomplished using the conventional machinery of optimization. The adversary's goal is to maximize Equation (2). The maximum of this function must occur at a *critical point*[2] or on its boundary. For our objective function in Equation (2), its partial derivatives exist and setting them to zero yields the following relation,

$$1 < t < T \quad M_t^2 = M_{t-1} \cdot M_{t+1} \tag{4}$$

[2] A critical point of a function f is a point x where all its partial derivatives are zero, i.e., $\nabla f(x) = 0$.

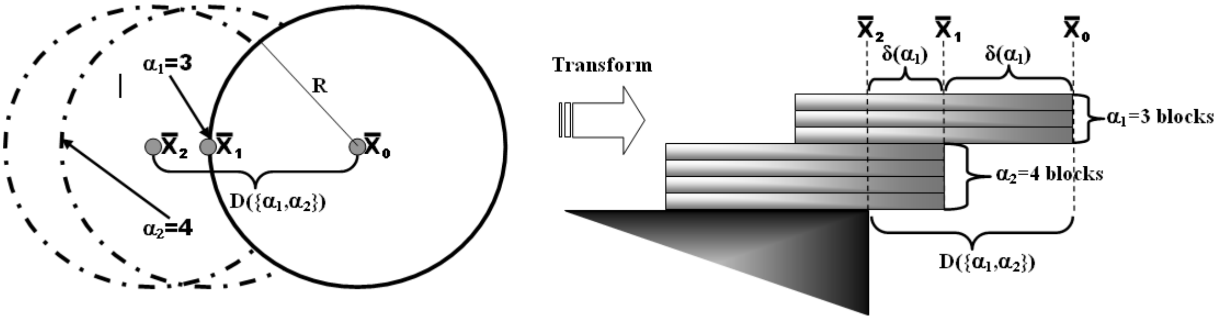


Figure 2: A figure depicting the physics analogy between the attack sequence $\{\alpha_1 = 3, \alpha_2 = 4\}$ and the concept of optimally stacking blocks on the edge of a table to extend beyond the edge of the table. In this analogy, blocks of length $2R$ with a starting edge at $\bar{X}_t$ are equivalent to placing an attack point at the t-th retraining iteration of a hypersphere with mean $\bar{X}_t$ and radius R. Vertical stacks can be interpreted as placing several points at time t and time flows down the blocks to the table. The overall effect of the attack is the displacement $\mathrm{D}(\{\alpha_1, \alpha_2\})$.

with $M_1 = 1$ and $M_T = M$. By considering the logarithm of the cumulative masses ℓM_t, these conditions become a linear difference equation whose solution yields the unique optimal strategy of $M_t^* = M^{\frac{-1}{T-1}}$. Finally, since the real-valued optima is an upper bound on the relative displacement achieved by any integer solution, we have

$$\mathrm{D}^*_{R,T}(M) \leq T - (T-1) \cdot M^{\frac{-1}{T-1}} \leq T \qquad (5)$$

For $M \geq 1$ and $T > 1$, Equation (5) is monotonically increasing in M. Given that the desired relative displacement to the goal is D_R, the bound in Equation (5) can be inverted to bound the minimal effort M^* required to achieve the goal. Since $D_R < T$, this bound is given by:

$$M^* \geq \left(\frac{T-1}{T-D_R}\right)^{T-1} \qquad (6)$$

Not surprisingly, this bound converges to the unconstrained bound given in Equation (3) as $T \to \infty$ and, in fact, dominates it since $D_R < T$.

4.3 Bounds with Initial Support

In the previous analyses, we provided lower bounds on the adversary's required effort when the adversary has complete control of the learner's training. The bounds in Equations (3) and (6) ensure a desirable exponential increase in effort M^* as the goal D_R increases. However, these bounds are poor for small D_R. In particular, for $D_R \leq 1$ the bounds are $M^* \geq \epsilon$ where $0 < \epsilon \leq 1$. In this range, these bounds only guarantee the attack must use a "single" attack point. Moreover, this an important range of D_R since the adversary's goal may be near the boundary.

The deficiency in our bounds is a result of the lack of the initial support for the hypersphere prior to the attack. Since no mass is supporting the initial hypersphere, placing points on the boundary will guarantee a relative displacement of 1 for the subsequent mean $\bar{X}_1$. Therefore, the adversary can reach the objective of $D_R \leq 1$ with any $M \geq 1$ in a single iteration.

By adding initial non-adversarial data to our model, we can compensate for this deficiency. We augment the model with N initial non-adversarial points that support the hypersphere before the attack (i.e., initial clean data). In this setting we pretend, without loss of generality, that the initial data was adversarial for some imaginary $\bar{X}_{-1}$ that lies along the line of the attack. Thus, in this imaginary scenario, the addition of the N points caused a relative displacement of 1 toward the goal resulting in the initial mean $\bar{X}_0$. We incorporate $M_0 = N$ as our new initial attack points and have a total of $M_T = M + N$. Thus, the real-valued analysis of Section 4.2 can be applied yielding an optimal policy of $M_t^* = N^{\frac{T-t}{T}}(M+N)^{\frac{t}{T}}$ and a bound on the relative displacement similar to Equation (5).

Again, the monotonicity in M allows us to invert this bound. For a desired relative displacement D_R, our new bound is given by

$$M^* \geq N\left(1 - \frac{D_R}{T}\right)^{-T} - N \qquad (7)$$

Moreover, this bound asymptotically converges to $M^* \geq N\left(e^{D_R} - 1\right)$ and dominates it for feasible D_R. The bound in Equation (7) ensures that even for small D_R, the adversary's effort is a multiple of N that increases exponentially in the desired displacement [9].

5. CONCLUSION

The bounds on the attacker derived in Section 4 explicitly demonstrate the feasibility of a *causative targeted integrity* attack against our simplified model but bound its effect. These attacks are characterized by a trade-off between minimizing the duration of the attack and the amount of adversarial data. Moreover, while longer attacks require less total data, their effect is more likely to be mitigated by normal traffic not incorporated in our model. Thus, under the assumptions of our model, effective attacks can be expected to exhibit an exponential behavior that could facilitate detection. It remains to be seen if these behaviors extend to more realistic learning models.

The bounds in Equations (3), (6), and (7) also illustrate the robustness of the fixed radius hypersphere under a *bootstrapping* policy. These bounds increase exponentially in D_R, the relative displacement required for success of the attack. Moreover the strong bound in Equation (7) shows that the adversary must use a multiple of N (the number of initial non-adversarial points) attack points where the multiple increases exponentially in D_R. This bound suggests the following general guidelines for robustness against an adversary.

Large N - A large initially clean corpus deters against attacks.

Small R - A tightly fit boundary can increase effort required by the adversary.

Large Displacement - Ensure that an adversary's goal should be far from the boundary.

Our model provides an analytic justification for these general rules, although it is unclear how dependent they are on our model's assumptions.

In future work we would like to extend such models beyond the simple framework presented in this model. The results obtained here were largely dependent on the *bootstrapping* policy to provide robustness. This policy restricts and biases our model against adaptation. One future research goal is to examine a broader range of rejection policies and to assess their consequences. We would also like to extend such analyses to a broader range of models beyond bounded hyperspheres.

Acknowledgments

We would like to thank Peter Bartlett, Doug Tygar, Michael Jordan, Marco Barreno, Russell Sears, and Brighten Godfrey for the discussions and input that made this work possible. This work was supported in part by DETER, which receives support from the Homeland Security Advanced Research Projects Agency and the National Science Foundation (NSF grant ANI-0335241) and in part by TRUST (the Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422) and the following organizations: Cisco, ESCHER, Hewlett Packard, IBM, Intel, Microsoft, ORNL, Qualcomm, Sun and Symantec. The views expressed here are solely those of the authors and do not necessarily reflect the views of the companies, funding agencies, or any agency of the U.S. government.

REFERENCES

[1] Barreno, M., Nelson, B., Sears, R., Joseph, A. D., and Tygar, J. Can machine learning be secure? In *ACM Symposium on Information, Computer, and Communications Security* (2006).

[2] Dalvi, N., Domingos, P., Mausam, Sanghai, S., and Verma, D. Adversarial classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Seattle, WA, 2004), ACM Press, pp. 99–108.

[3] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer, 2003.

[4] Kearns, M., and Li, M. Learning in the presence of malicious errors. *SIAM Journal on Computing 22* (1993), 807–837.

[5] Lowd, D., and Meek, C. Adversarial learning. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2005), pp. 641–647.

[6] Lowd, D., and Meek, C. Good word attacks on statistical spam filters. In *Proceedings of the Second Conference on Email and Anti-Spam (CEAS)* (2005).

[7] Mahoney, M. V., and Chan, P. K. Learning nonstationary models of normal network traffic for detecting novel attacks. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2002), ACM Press, pp. 376–385.

[8] Meyer, T. A., and Whateley, B. SpamBayes: Effective open-source, Bayesian based, email classification system. In *Conference on Email and Anti-Spam* (2004).

[9] Nelson, B. Designing, Implementing, and Analyzing a System for Virus Detection. Master's thesis, University of California at Berkeley, Dec. 2005.

[10] Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., and Williamson, R. Estimating the support of a high-dimensional distribution. Tech. Rep. 87, Microsoft Research, 1999.

[11] Segal, R., Crawford, J., Kephart, J., and Leiba, B. SpamGuru: An enterprise anti-spam filtering system. In *CEAS* (2004).

[12] Shawe-Taylor, J., and Cristianini, N. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[13] Stolfo, S. J., Hershkop, S., Wang, K., Nimeskern, O., and Hu, C. W. A behavior-based approach to secure email systems. In *Mathematical Methods, Models and Architectures for Computer Networks Security* (2003).

[14] Stolfo, S. J., Li, W. J., Hershkop, S., Wang, K., Hu, C. W., and Nimeskern, O. Detecting viral propagations using email behavior profiles. In *ACM TOIT* (2004).

[15] Valiant, L. G. A theory of the learnable. *Communications of the ACM 27*, 11 (Nov. 1984), 1134–1142.