Query-limited Black-box Attacks to Classifiers

Fnu Suya University of Virginia
Yuan Tian University of Virginia
David Evans University of Virginia
Paolo Papotti EURECOM

Abstract

In this paper, we study black-box attacks on machine learning classifiers where the adversary has a limited opportunity to interact with the model via queries. Queries to the machine learning model are expensive for the adversary, because each query poses some risk of detection, and attackers pay a service per query. Previous works in black-box attack did report the query number used in their attack procedure, however, none of these works explicitly set minimizing query number as a major objective. Specifically, we consider the problem of attacking machine learning classifiers subject to budget of feature modification cost with minimum number of queries where each query returns only a class and confidence score. We found that the number of queries can be reduced to around 30% of the random modification on average, and even less (< 10%) when feature modification cost budget is small.

1 Introduction

Recent works reveal the vulnerabilities of current machine learning models to carefully crafted adversarial examples [1, 2, 3, 4]. In many scenarios, complete model information is not available to the attacker and hence it is important to study black-box attacks, where the attackers do not have full knowledge of the model but only some way of interacting with it. In this work, we focus on black-box attacks where only query access to the model is available. We assume the query result can be returned in the form of confidence prediction score.

Since queries to the model is costly, attackers are motivated to minimize query number when interacting with the model. In the scenario of spam email detection system, query to the underlying classification model is in the form of emails and adversaries will not be able to afford large number of email queries [5]. Hence, our problem setting is: given a budget on feature modification cost, find an adversarial example with the minimal number of queries. This problem can be cast as a constrained optimization problem. Specifically, given a budget C on total feature modification cost, minimize the total number of queries in the process of searching adversarial examples. The problem can be mathematically formulated as:

$$\begin{aligned} &\min Q(\mathbf{x}) \\ &\text{s.t. } f(\mathbf{x}) \neq f(\mathbf{x}^A) \\ &\quad c(\mathbf{x}, \mathbf{x}^A) \leq C \end{aligned} \tag{1}$$

where $Q(\mathbf{x})$ denotes total number of queries consumed in searching for an adversarial example. $c(\mathbf{x}, \mathbf{x}^A) = ||\mathbf{x} - \mathbf{x}^A||_p$ denotes feature modification cost, where $\mathbf{x}^A$ is the original instance. In this paper, we apply L_1-norm as the application scenario is in text domain. $f(\mathbf{x})$ denotes the prediction label of instance $\mathbf{x}$.

Above formulation is highly intractable as we do not have a closed form expression for function $Q(\mathbf{x})$ and also, $f(\mathbf{x})$ is unknown as we assume black-box access to the machine learning model. Due to the high intractability of the resulting problem, we transform the original optimization form in a way that suits for a global optimization framework. Global optimization techniques works well for query based optimization problems, where query to the unknown objective function is expensive. It is the major advantage of global optimization to minimize (maximize) an unknown objective

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

function with less number of queries. In particular, we apply Bayesian optimization (BO) as the main approach for solving our optimization problem. Details can be found in section 3. Our empirical results show that BO based attack can find valid adversarial samples with limited number of queries. We summarize our contribution as follows: (1) we study a new formulation of minimizing query numbers in black-box attack setting; (2) we propose Bayesian optimization based (BO) black-box attack strategy, which reduces the total query number efficiently.

We provide background on Bayesian optimization (section 2) and how we use it to find a sequence of queries to minimize the number of interactions (section 3). Section 4 reports on our preliminary experiments using these techniques to generate spam messages that evade a black-box detector.

Related Work Prior works have studied black-box attacks on machine learning classifiers in two categories: substitute model attacks and numerical approximation method-based attacks.

First type of attack uses query responses obtained from the target model to train a substitute model, and then generates adversarial examples for that substitute model. Several results have shown that adversarial examples produced this way are transferable and often effective against the original model [6, 7, 8]. For example, Papernot et al. train a substitute model (locally) for attacking the target unknown black-box model [7]. The local model is trained using training data with labels obtained through querying the target model. As there exists transferability among different models [8, 9], it is highly likely to obtain instances that are adversarial to both local and the unknown target model. The drawback of the substitute model is it will suffer form the transfer loss as not all adversarial examples can transfer from one model to another model [10]. Also, the number of training instances needed to produce an effective substitute model may be very large.

Another line of work, introduced by [10], is to apply some numerical approximation to model gradient calculation to support known white-box attack strategies. The authors approximate the gradient information by symmetric difference quotient and further utilize the Carlini & Wagner attack [11] to generate adversarial samples. The drawback of this approach is in the high query number. In leveraging the Carlini & Wagner attack, gradient needs to be calculated in each step and single gradient estimation requires high number of function value evaluations resulted from the high dimensional feature space.

Previous papers in black-box attack scenario never explicitly consider minimizing total query number. One most related work is in [5], where the author considers spam email setting and sets a bound on the total number of queries and feature modification cost. The attacker then applies query strategy to find adversarial sample and if no adversarial example is found within given cost or query budget, just stop the process. However, this work only considers linear classifier. In contrast, our work considers classifiers whose boundary can be in any shape (including linear boundary).

2 Background on Bayesian Optimization

Bayesian optimization is a global optimization technique that handles optimization problem with unknown objective function. It works by querying the unknown function and aims to find optimal solution with minimum number of queries to the objective function. Detailed background information can be found in Appendix A

3 Minimize Query Numbers with Bayesian Optimization

As discussed in section 1, we face two major challenges of no closed form expression for function $Q(\mathbf{x})$ and an unknown constraint in $f(\mathbf{x})$, where only queries to $f(\mathbf{x})$ is allowed. Hence, optimization through query is required for our problem. We first handle the unknown constraint by following the previous approach [11, 1] and move the intractable classification label constraint into the objective function. As we we do not know $f(\mathbf{x})$, we transform the constraint of $f(\mathbf{x}) \neq f(\mathbf{x}^A)$ as minimizing the probability of $\mathbf{x}$ having same label with $\mathbf{x}^A$. In order to minimize the total number of queries, as outlined in the objective function of Eq. (1), we adopt a heuristic strategy for minimization. Namely, in each step of query, we utilize our query history to select the (currently) best point for solving the optimization problem above. Hence, specific to our problem, in each query step, we find the best point for minimizing Pr$[f(\mathbf{x}) == f(\mathbf{x}^A)]$ and consequently, the whole optimization process eventually minimizes function $Q(\mathbf{x})$ (i.e., total query number). Our query step will terminate once we have found a valid instance whose label is different from $\mathbf{x}^A$. The problem can be mathematically formulated as:

$$\begin{aligned} &\min \Pr[f(\mathbf{x}) == f(\mathbf{x}^A)] \\ &\text{s.t.} \quad c(\mathbf{x}, \mathbf{x}^A) \leq C \end{aligned} \tag{2}$$

To solve the problem in Eq. (2), we adopt the Bayesian optimization framework. As discussed in detail in Appendix A, Bayesian optimization suits for solving unknown function (in our case, $\Pr[f(\mathbf{x}) == f(\mathbf{x}^A)]$) minimization with less number of queries (in our case, $Q(\mathbf{x})$). Note that $c(\mathbf{x}, \mathbf{x}^A)$ is a function known to the adversary (i.e., L_1-norm constraint). We now have a Bayesian optimization problem with unknown objective and known constraint. We take Upper Confidence Bound (UCB) as the acquisition function (Acq$(\mathbf{x})$) and select the point that maximizes Acq$(\mathbf{x})$ with respect to the constraint $c(\mathbf{x}, \mathbf{x}^A) \leq C$ in each step. Details of UCB and acquisition function can be found in Appendix A.

We apply the DIRECT algorithm [12] to solve acquisition function maximization problem in Eq. (4) in Appendix A. DIRECT algorithm is a well-known algorithm for solving global optimization problems. To increase the robustness of the code when facing with an extremely small cost budget C, we applied DIRECT method with minor modifications: DIRECT method works by dividing a unit hypercube sequentially and evaluating function values in each of the sub hyperrectangle [12] and the initial point is center of the unit hypercube. Originally, each dimension value of this point was determined by the lower and upper bounds in that dimension. When C is very small and the initial center is too far away from initial point $\mathbf{x}^A$, it is very hard to find an instance within the feature cost budget (which will result in very long search time). Instead, we now take the initial point $\mathbf{x}^A$ as the center of the unit hypercube such that we can always find instances satisfy feature modification cost constraint. The outline for the Bayesian algorithm is shown in Algorithm 1. Details regarding Guassian process update can be found in [13] and are omitted here due to space limitation.

Algorithm 1 Bayesian Optimization Based Black-box Attack

Require: $\mathbf{x}^A, C, f(\mathbf{x}^A), N$
$\mathbf{x} = \mathbf{x}^A$
for $t = 1, 2, ..., N$ **do**
  Find $\mathbf{x}_t$ by solving problem $\mathbf{x}_t =$ argmax Acq$(\mathbf{x}|D_{1:t-1})$, s.t. $c(\mathbf{x}, \mathbf{x}^A) \leq C$
  Sample the objective function value: $y_t = \Pr(f(\mathbf{x}_t) == f(\mathbf{x}^A))$
  if $f(\mathbf{x}_t) \neq f(\mathbf{x}^A)$ **then**
    return $x^* = \mathbf{x}_t$;
  end if
  Augment the data $D_{1:t} = \{D_{1:t-1}, (\mathbf{x}_t, y_t)\}$ and update the Gaussian Process and Acq$(\mathbf{x})$.
end for
return $\mathbf{x}^* = \mathbf{x}^A$

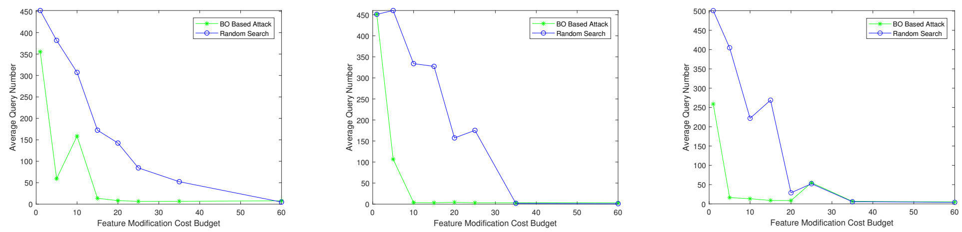
4 Evaluation

To evaluate the effectiveness of BO based black-box attacks, we conduct experiments on spam email dataset. The attacker's objective is to create a spam email (i.e., instance $\mathbf{x}^*$) that is misclassified by the unknown classifier while the L_1-norm distance (i.e., edit distance) to the original spam email $\mathbf{x}^A$ is within C. We show that BO based attack reduces query numbers significantly.

Spam Email Dataset The dataset [14] contains 4601 records and each record holds 57 attributes. Among the 57 features, 2 of them are integers (we discard these two attributes as we are currently dealing with continuous features). Every email is labeled as either spam or normal. We randomly choose 3500 of the instances to train three different classifiers (Probabilistic Linear SVM, Probabilistic RBF SVM, Artificial neural network (ANN)) and report the error rate on the remaining dataset. The original instance $\mathbf{x}^A$ is randomly selected from the spam emails.

Classifier Models We train both linear SVM and RBF kernel SVM, which achieve classification accuracy of 91% and 94% respectively. Details of transforming normal SVM into probabilistic SVM can be found in [15]. We also train an ANN model with classification accuracy of 94%.

Baseline In this paper, we compare our result with random search method, which will randomly generate values for each dimension and terminate the search process when the class label is changed.



(a) Linear SVM Classifier (b) RBF SVM Classifier (c) ANN Classifier

Figure 1: Average Query Number w.r.t Different Cost Budgets for Different Classifiers

Specifically, we take the cost budget C and generate random samples whose L_1-distance to $\mathbf{x}^A$ is in the range of $(C - \epsilon, C)$. We set $\epsilon = 0.05$. Our assumption here is, having larger distance to the original instance can maximize the chance of flipping into opponent class as boundary of the classifier is in normal shape.

For different classifiers, we compare query numbers of both algorithms (BO attack and random search) with respect to different C values. We took C as [1,5,10,15,20,25,35,60]. Note that, when C is extremely small, the chance of getting an adversarial example within the boundary is rare. Hence, we set some threshold values for both algorithms and once the iteration number exceeds the threshold, we take it as an indicator of non-existence of adversarial example. For BO attack, we set it as 50 and for random search, we set it as 500.

Result and Discussion We demonstrate our BO attack strategy uses far less amount of queries in finding valid adversarial examples. Details are shown in Figure 1, where 1a shows the average query number with respect to different feature modification cost budget C for linear probabilistic SVM model. Similarly, 1b, 1c represent results for probabilistic RBF SVM and ANN respectively.

In Figure 1a, BO attack takes [355,59,158,14,8,6,7,8] queries in response to C values in [1,5,10,15,20,25,35,60] and random search takes [451,381,307,172,142,85,53,5] queries. In Figure 1b, BO attack has [451,106,4,3,4,3,3,3] queries while random search has [451,460,334,327,157,175,2,1] queries. In Figure 1c, BO attack has [258,16,13,9,8,54,6,5] queries and random search takes [501,404,221,269,29,52,6,4] queries. In count of total query number, our BO based black-box attack finds valid adversarial example using small fraction of queries of random search, especially when the cost budget is small. Note that, the average query number shown here is a conservative estimation for the BO method, as we take all iterations of BO exceeding 50 as failure and set it to 500 for fair comparison with random search method. It is expected that our algorithm can reduce its query number by taking more Bayesian search steps and is therefore more practical. It is also observed that, when C is large, random search performs slightly better than BO based attack (in average, random search uses 2 or 3 queries less). As we are mostly concerned with smaller C values, our BO attack strategy is still more practical than random search.

We investigated possible reasons for random search outperforming BO attack when C is large: Bayesian optimization spends some additional few queries to make "mistakes" such that it can explore the whole space more comprehensively and as the total query number (with large C) is small, it can be outperformed by the random search method. We also checked the classification score of initial points $\mathbf{x}^A$s under these cases and found most of these $\mathbf{x}^A$s are close to the classification boundary. Hence, it also makes sense to have random search performing slightly better. It is our ongoing work to compare with other black-box attacking methods and test on data from different domains (e.g., image and text). We also note that, our BO approach can work for both targeted and untargeted attack. For untargeted attack, our current formulation works and for targeted attack, we simply set the objective function as maximizing Pr$[f(\mathbf{x}) == y^*]$, where y^* is the target class.

5 Conclusion

Our proposed black-box attack strategy considers the problem of generating adversarial example with minimum number of queries, which, to the best of our knowledge, was not addressed by previous literature. We then empirically verified that the approach is a promising method for devising a black-box attack with less number of (costly) queries.

References

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[3] Daniel Lowd and Christopher Meek. Adversarial learning. In *Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005.

[4] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

[5] Bo Li and Yevgeniy Vorobeychik. Feature cross-substitution in adversarial classification. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[6] Weiwei Hu and Ying Tan. Generating adversarial malware examples for black-box attacks based on GAN. *arXiv preprint arXiv:1702.05983*, 2017.

[7] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *ACM Asia Conference on Computer and Communications Security (AsiaCCS)*, 2017.

[8] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[9] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. *arXiv preprint arXiv:1610.08401*, 2016.

[10] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *10th ACM Workshop on Artificial Intelligence and Security (AISec)*, 2017.

[11] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.

[12] Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[13] Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT Press, 2006.

[14] M. Lichman. UCI machine learning repository, 2013.

[15] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999.

[16] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[17] José Miguel Hernández-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. In *International Conference on Machine Learning (ICML)*, 2015.

[18] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[19] Kenji Kawaguchi, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Bayesian optimization with exponential convergence. In *Advances in Neural Information Processing Systems*, pages 2809–2817, 2015.

[20] Adam D Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(Oct):2879–2904, 2011.

A Bayesian Optimization Background

Bayesian optimization is a derivative free strategy for global optimization of black-box functions [16, 17, 18]. The Bayesian optimization problem can be formulated as:

$$\begin{aligned} &\min g(\mathbf{x}) \\ &\text{s.t. } h(\mathbf{x}) \leq 0. \end{aligned} \tag{3}$$

Where $g(\mathbf{x})$ is an unknown function and $h(\mathbf{x})$ can either be known or unknown. In our formulation, $h(\mathbf{x}) = c(\mathbf{x}) - C$ is a known function. Unlike traditional optimization algorithm, BO method does not depend on gradient or hessian information, instead it works by querying function value of a point in each step of the interactive optimization process [16]. And as queries to $g(\mathbf{x})$ is assumed to be costly, BO algorithm minimizes total number of queries spent in the whole search process for the problem above. Step by step explanations of BO method are shown below.

Since the objective function is unknown, a *prior* over functions is assumed to be known, e.g., Gaussian prior [13] is a common attempt to model what we know about the function [16, 18, 17]. With the defined priors and current observations, the *posterior probability* of next function value can be defined. And with the posterior probability distribution, an *acquisition function* is then defined to capture an *exploration-exploitation* trade-off in determining the next query point. Points with larger Acquisition function values are more likely to have smaller $g(\mathbf{x})$ values. Thus, we prefer points with larger acquisition function values. As the point in each step is selected to maximize the current acquisition function, the whole optimization process heuristically minimizes number of interactions needed for searching the optimal solution. Convergence rate of Bayesian optimization can be referred to [19, 20].

Exploration prefers locations (i.e., points) where the uncertainty is high, while exploitation prefers locations where the objective function value is high (or low) in maximization (or minimization) problem. The acquisition function is updated along with the update of posterior probability. In this paper, we apply upper confidence bound (UCB) selection criterion in selecting the specific acquisition function type. As we assume the unknown function value follows Gaussian distribution, we obtain the closed form expression of the acquisition function (UCB) for point $\mathbf{x}$ as $\text{Acq}(\mathbf{x}) = \mu(\mathbf{x}) + \kappa\sigma(\mathbf{x})$, where $\sigma(\mathbf{x}), \mu(\mathbf{x})$ are variance and mean functions at point $\mathbf{x}$ and κ is a constant. We refer readers to [16] for more details regarding different types of acquisition functions and closed form expression for $\mu(\mathbf{x}), \sigma(\mathbf{x})$. The following optimization problem is solved to obtain the current best point $\mathbf{x}_t$ in step t.

$$\begin{aligned} &\max \text{Acq}(\mathbf{x}) \\ &\text{s.t. } c(\mathbf{x}, \mathbf{x}^A) \leq C. \end{aligned} \tag{4}$$

Once the query result $f(\mathbf{x}_t)$ of the point $\mathbf{x}_t$ is returned, the BO framework updates its belief about the unknown function distribution and the whole procedure iterates until termination condition is satisfied.