5-2016

Learning adversary behavior in security games: A PAC model perspective

Arunesh SINHA
Singapore Management University, aruneshs@smu.edu.sg

Debarun KAR

Milind TAMBE

Learning Adversary Behavior in Security Games: A PAC Model Perspective

Arunesh Sinha, Debarun Kar, Milind Tambe
University of Southern California
{aruneshs, dkar, tambe}@usc.edu

ABSTRACT

Recent applications of Stackelberg Security Games (SSG), from wildlife crime to urban crime, have employed machine learning tools to learn and predict adversary behavior using available data about defender-adversary interactions. Given these recent developments, this paper commits to an approach of directly learning the response function of the adversary. Using the PAC model, this paper lays a firm theoretical foundation for learning in SSGs and provides utility guarantees when the learned adversary model is used to plan the defender's strategy. The paper also aims to answer practical questions such as how much more data is needed to improve an adversary model's accuracy. Additionally, we explain a recently observed phenomenon that prediction accuracy of learned adversary behavior is not enough to discover the utility maximizing defender strategy. We provide four main contributions: (1) a PAC model of learning adversary response functions in SSGs; (2) PAC-model analysis of the learning of key, existing bounded rationality models in SSGs; (3) an entirely new approach to adversary modeling based on a non-parametric class of response functions with PAC-model analysis and (4) identification of conditions under which computing the best defender strategy against the learned adversary behavior is indeed the optimal strategy. Finally, we conduct experiments with real-world data from a national park in Uganda, showing the benefit of our new adversary modeling approach and verification of our PAC model predictions.

1. INTRODUCTION

Stackelberg Security Games (SSGs) are arguably the best example of the application of the Stackelberg game model in the real world. Indeed, numerous successful deployed applications [30] (LAX airport, US air marshal) and extensive research on related topics [5, 18, 6, 7] provide evidence about the generality of the SSG framework. More recently, new application domains of SSGs, from wildlife crime to urban crime, are accompanied by significant amounts of past data of recorded defender strategies and adversary reactions. This has enabled the learning of adversary behavior from such data [36, 35]. Also, analysis of these datasets and human subject experiment studies [26] have revealed that modeling bounded rationality of the adversary enables the defender to further optimize her allocation of limited security resources. Thus, learning the adversary's bounded rational behavior and computing defender strategy based on the learned model has become an important area of research in SSGs.

However, without a theoretical foundation for this learning and strategic planning problem, many issues that arise in practice cannot be explained or addressed. For example, it has been recently observed that in spite of good prediction accuracy of the learned models of adversary behavior, the performance of the defender strategy that is computed against this learned adversary model is poor [12]. A formal study could also answer several other important questions that arise in practice, for example, (1) How many samples would be required to learn a "reasonable" model of adversary behavior in a given SSG? (2) What utility bound can be provided when deploying the best defender strategy that is computed against the learned adversary model?

Motivated by the learning of adversary behavior from data in recent applications [13, 36], we adopt the framework in which the defender first learns the *response function* of the adversary (adversary behavior) and then optimizes against the learned response. This paper is the first theoretical study of the adversary bounded rational behavior learning problem and the optimality guarantees (utility bounds) when computing the best defender strategies against such learned behaviors[1]. Indeed, unlike past theoretical work on learning in SSGs (see related work) where reasoning about adversary response happens through payoff and rationality, we treat the response of the bounded rational adversary as the object to be learned.

Our *first contribution* is using the Probably Approximately Correct (PAC) model [16, 2] to analyze the learning problem at hand. A PAC analysis yields sample complexity, i.e., the number of samples required to achieve a given level of learning guarantee. Hence, the PAC analysis allows us to address the question of required quantity of samples raised earlier. While PAC analysis is fairly standard for classifiers and real valued functions (i.e., regression) it is not an out-of-the-box approach. In particular, PAC-model analysis of SSGs brings to the table significant new challenges. To begin with, given that we are learning adversary response functions, we must deal with the output being a probability distribution over the adversary's actions, i.e., these response functions are vector-valued. We appeal to the framework of Haussler [14] to study the PAC learnability of *vector-valued* response functions. For SSGs, we first pose the learning problem in terms of maximizing the likelihood of seeing the attack data, but without restricting the formulation to any particular class of response functions. This general PAC framework for learning adversary behavior in SSGs enables the rest of the analysis in this paper.

Our *second contribution* is an analysis of the Subjective Utility Quantal Response (SUQR) model of bounded rationality adver-

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

[1]Our theory results apply to Stackelberg games also. See Appendix.

sary behavior used in SSGs, which posits a class of parametrized response functions with a given number of parameters (and corresponding features). SUQR is the best known model of bounded rationality in SSGs, resulting in multiple deployed applications [13, 11]. In analyzing SUQR, we advance the state-of-the-art in the mathematical techniques involved in PAC analysis of vector-valued function spaces. In particular, we provide a technique to obtain sharper sample complexity for SUQR than simply directly applying Haussler's original techniques. We decompose the given SUQR function space into two (or more) parts, performing PAC analysis of each part and finally combining the results to obtain the sample complexity result (which scales as $T \log T$ with T targets) for SUQR (see details in Section 5).

Our *third contribution* includes an entirely new behavioral model specified by the *non-parametric Lipschitz* (NPL) class of response functions for SSGs, where the only restriction on NPL functions is Lipschitzness. The NPL approach makes very few assumptions about the response function, enabling the learning of a multitude of behaviors albeit at the cost of higher sample complexity. As NPL has never been explored in learning bounded rationality models in SSGs, we provide a *novel learning technique* for NPL. We also compute the sample complexity for NPL. Further, we observe in our experiments that the power to capture a large variety of behaviors enables NPL to perform better than SUQR with the real-world data from Queen Elizabeth National Park (QENP) in Uganda.

Our *fourth contribution* is to convert the PAC learning guarantee into a bound on the utility derived by the defender when planning her strategy based on the learned adversary behavior model. In the process, we make explicit the assumptions required from the dataset of adversary's attacks in response to deployed defender mixed strategies in order to discover the optimal (w.r.t. utility) defender strategy. These assumptions help explain a puzzling phenomenon observed in recent literature on learning in SSGs [12]—in particular that learned adversary behaviors provide good prediction accuracy, but the best defender strategy computed against such learned behavior may not perform well in practice. We explain this as a mismatch between the adversary model guarantee required for computing the best defender strategy and the guarantee provided by the (PAC based) learning methods. The key is that the dataset for learning must not simply record a large number of attacks against few defender strategies but rather contain the attacker's response against a variety of defender mixed strategies. We discuss the details of our assumptions and its implications for the strategic choice of defender's actions in Section 7.

We also conduct experiments with real-world poaching data from the QENP in Uganda (obtained from [25]) and data collected from human subject experiments. The experimental results support our theoretical conclusions about the number of samples required for different learning techniques. Showing the value of our new NPL approach, the NPL approach outperforms all existing approaches in predicting the poaching activity in QENP. Finally, our work opens up a number of exciting research directions, such as studying learning of behavioral models in active learning setting and real-world application of non-parametric models. [2]

2. RELATED WORK

Learning and planning in SSGs with rational adversaries has been studied in two recent papers [8, 3], and in Stackelberg games by Letchford et al. [20] and Marecki et al. [22]. All these papers study the learning problem under an active learning framework, where the defender can choose the strategy to deploy within the learning process. Also, all these papers study the setting with perfectly rational adversaries. Our work differs as we study bounded rational adversaries in a passive learning scenario (i.e., with given data) and once the model is learned we analyze the guarantees of planning against the learned model. Also, our focus on SSGs differentiates us from recent work on PAC learnability in co-operative games [4], in which the authors study PAC learnability of the value function of coalitions with perfectly rational players. Also, our work is orthogonal to adversarial learning [33], which studies game theoretic models of an adversary attacking a learning algorithm.

PAC learning model has a very rich and extensive body of work [2]. The PAC model provides a theoretical underpinning for most standard machine learning techniques. We use the PAC framework of Haussler [14]. For the parametric case, we derive sharp sample complexity bounds based on covering numbers using our techniques rather than bounding it using the standard technique of pseudo-dimension [28] or fat shattering dimension [2]. For the NPL case we use results from [32] along with our technique of bounding the mixed strategy space of the defender; these results [32] have also been used in the study of Lipschitz classifiers [21] but we differ as our hypothesis functions are real vector-valued.

3. SSG PRELIMINARIES

This section introduces the background and preliminary notations for SSGs. A summary of notations used in this paper is presented in Table 1. An SSG is a two player Stackelberg game between a defender (leader) and an adversary (follower) [27]. The defender wishes to protect T targets with a limited number of security resources K ($K \ll T$). For ease of presentation, we restrict ourselves to the scenario with no scheduling constraints (see Korzhyk et al. [18]). The defender's pure strategy is to allocate each resource to a target. A defender's mixed-strategy $\tilde{x}$ ($\forall j \in \mathcal{P}.\ \tilde{x}_j \in [0,1], \sum_{j=1}^{\mathcal{P}} \tilde{x}_j = 1$) is then defined as a probability distribution over the set of all possible pure strategies $\mathcal{P}$. An equivalent description (see Korzhyk et al. [18]) of these mixed strategies are coverage probabilities over the set of targets: x ($\forall i \in T.\ x_i \in [0,1], \sum_{i=1}^{T} x_i \leq K$). We refer to this latter description as the mixed strategy of the defender.

A pure strategy of the adversary is defined as attacking a single target. The adversary's mixed strategy is then a categorical distribution over the set of targets. Thus, it can be expressed as parameters q_i ($i \in T$) of a categorical distribution such that $0 \leq q_i \leq 1$ and $\sum_i q_i = 1$. The adversary's response to the defender's mixed strategy is given by a function $q : X \rightarrow Q$, where Q is the space of all mixed strategies of the adversary. The matrix U specifies the payoffs of the defender, and her expected utility is $x^T U q(x)$ when she plays a mixed strategy $x \in X$.

Bounded Rationality Models: We discuss the SUQR model and its representation for the analysis in this paper below. Building on prior work on quantal response [24], SUQR [26] states that given n actions, a human player plays action i with probability $q_i \propto e^{w \cdot v}$, where v denotes a vector of feature values for choice i and w denotes the weight parameters for these features. The model is equivalent to conditional logistic regression [23]. The features are specific to the domain, e.g., in case of SSG applications, the set of features include the coverage probability x_i, the reward R_i and penalty P_i of target i. Since, other than the coverage x, remaining features are fixed for each target in real world data, we assume a target-specific feature c_i (which may be a linear combination of re-

[2]Due to lack of space, most proofs are in the online Appendix: http://teamcore.usc.edu/papers/2016/PAC.Appendix.pdf

Notation	Meaning
T, K	Number of targets, defender resources
$d_{l_p}(o, o')$	l_p distance between points o, o'
$\bar{d}_{l_p}(o, o')$	Average l_p distance: $= d_{l_p}(o, o')/n$
X	Instance space (defender mixed strategies)
Y	Outcome space (attacked target)
A	Decision space
$h \in \mathcal{H}$	$h : X \to A$ is the hypothesis function
$\mathcal{N}(\epsilon, \mathcal{H}, d)$	ϵ-cover of set $\mathcal{H}$ using distance d
$\mathcal{C}(\epsilon, \mathcal{H}, d)$	capacity of $\mathcal{H}$ using distance d
$r_h(p), \hat{r}_h(\vec{z})$	true risk, empirical risk of hypothesis h
$d_{L^1(P,d)}(f, g)$	L_1 distance between functions f, g
$q^p(x)$	parameters of true attack distribution
$q^h(x)$	parameters of attack distr. predicted by h

Table 1: Notations

wards and penalties) and analyze the following *generalized*[3] form of SUQR with parameters w_1 and c_i's: $q_i(x) \propto e^{w_1 x_i + c_i}$. As $\sum_{i=1}^{T} q_i(x) = 1$, we have:

$$q_i(x) = \frac{e^{w_1 x_i + c_i}}{\sum_{j=1}^{T} e^{w_1 x_j + c_j}}.$$

Equivalent Alternate Representation: For ease of mathematical proofs, using standard techniques in logistic regression, we take $q_T \propto e^0$, and hence, $q_i \propto e^{w_1(x_i - x_T) + (c_i - c_T)}$. To shorten notation, let $c_{iT} = c_i - c_T$, $x_{iT} = x_i - x_T$. By multiplying the numerator and denominator by $e^{w_1 x_T + c_T}$, it can be verified that $q_i(x) = \frac{e^{w_1 x_{iT} + c_{iT}}}{e^0 + \sum_{j=1}^{T-1} e^{w_1 x_{jT} + c_{jT}}} = \frac{e^{w_1 x_i + c_i}}{\sum_{j=1}^{T} e^{w_1 x_j + c_j}}$.

4. LEARNING FRAMEWORK FOR SSG

First, we introduce some notations: given two n-dimensional points o and o', the l_p distance d_{l_p} between the two points is: $d_{l_p}(o, o') = ||o - o'||_p = (\sum_{i=1}^{n} |o_i - o'_i|^p)^{1/p}$. In particular, $d_{l_\infty}(o, o') = ||o - o'||_\infty = \max_i |o_i - o'_i|$. Also, $\bar{d}_{l_p} = d_{l_p}/n$. KL denotes the Kullback-Leibler divergence [19].

We use the learning framework of Haussler [14], which includes an *instance space* X and *outcome space* Y. In our context, X is same as the space of defender mixed strategies $x \in X$. *Outcome space* Y is defined as the space of all possible categorical choices over a set of T targets (i.e., choice of target to attack) for the adversary. Let t_i denote the attacker's choice to attack the i^{th} target. More formally $\mathrm{t}_i = \langle \mathrm{t}_i^1, \ldots, \mathrm{t}_i^T \rangle$, where $\mathrm{t}_i^j = 1$ for $j = i$ and otherwise 0. Thus, $Y = \{\mathrm{t}_1, \ldots, \mathrm{t}_T\}$. We will use y to denote any general element of Y. To give an example, given three targets T_1, T_2 and T_3, $Y = \{\mathrm{t}_1, \mathrm{t}_2, \mathrm{t}_3\} = \{\langle 1, 0, 0\rangle, \langle 0, 1, 0\rangle, \langle 0, 0, 1\rangle\}$, where t_1 denotes $\langle 1, 0, 0\rangle$, i.e., it denotes that T_1 was attacked while T_2 and T_3 were not attacked, and so on. The training data are samples drawn from $Z = X \times Y$ using an unknown probability distribution, say given by density $p(x, y)$. Each training data point (x, y) denotes the adversary's response $y \in Y$ (e.g., t1 or attack on target 1) to a particular defender mixed strategy $x \in X$. The density p also determines the true attacker behavior $q^p(x)$ which stands for the conditional probabilities of the attacker attacking a target given x so that $q^p(x) = \langle q_1^p(x), \ldots, q_T^p(x) \rangle$, where $q_i^p(x) = p(\mathrm{t}_i | x)$.

Haussler [14] also defines a *decision space* A, a *space of hypothesis* (functions) $\mathcal{H}$ with elements $h : X \to A$ and a *loss function* $l : Y \times A \to \mathbb{R}$. The hypothesis h outputs values in A that enables computing (probabilistic) predictions of the actual outcome. The loss function l captures the loss when the real outcome is $y \in Y$ and the prediction of possible outcomes happens using $a \in A$.

Example 1: Generalized SUQR For the parametric representation of generalized SUQR in the previous section and considering our 3-target example above, $\mathcal{H}$ contains vector valued functions with $(T - 1) = 2$ components that form the exponents of the numerator of prediction probabilities q_1 and q_2. $\mathcal{H}$ contains two components, since the third component q_3 is proportional to e^0 as discussed above. That is, $\mathcal{H}$ contains functions of the form: $\langle w_1(x_1 - x_3) + c_{13}, w_1(x_2 - x_3) + c_{23} \rangle$; $\forall x \in X$. Also, A is the range of the functions in $\mathcal{H}$, i.e., $A \subset \mathbb{R}^2$. Then, given $h(x) = \langle a_1, a_2 \rangle$, the prediction probabilities $q_1^h(x), q_2^h(x), q_3^h(x)$ are given by $q_i^h(x) = \frac{e^{a_i}}{1 + e^{a_1} + e^{a_2}}$ (assume $a_3 = 0$).

PAC learnability: The learning algorithm aims to learn a $h \in \mathcal{H}$ that minimizes the *true risk* of using the hypothesis h. The *true risk* $r_h(p)$ of a particular hypothesis (predictor) h, given density function $p(x, y)$ over $Z = X \times Y$, is the expected loss of predicting $h(x)$ when the true outcome is y:

$$r_h(p) = \int p(x, y) l(y, h(x))\, dx\, dy$$

Of course, as p is unknown the true risk cannot be computed. However, given (enough) samples from p, the true risk can be estimated by the *empirical risk*. The *empirical risk* $\hat{r}_h(\vec{z})$, where $\vec{z}$ is a sequence of m training samples from Z, is defined as: $\hat{r}_h(\vec{z}) = 1/m \sum_{i=1}^{m} l(y_i, h(x_i))$. Let h^* be the hypothesis that minimizes the true risk, i.e., $r_{h^*}(p) = \inf\{r_h(p) \mid h \in \mathcal{H}\}$ and let $\hat{h}^*$ be the hypothesis that minimizes the empirical risk, i.e., $\hat{r}_{\hat{h}^*}(\vec{z}) = \inf\{\hat{r}_h(\vec{z}) \mid h \in \mathcal{H}\}$. The following is the well-known PAC learning result [2] for any *empirical risk minimizing* (ERM) algorithm $\mathcal{A}$ yielding hypothesis $\mathcal{A}(\vec{z})$:

$$\begin{gathered}\text{If } Pr(\forall h \in \mathcal{H}. |\hat{r}_h(\vec{z}) - r_h(p)| < \alpha/3) > 1 - \delta/2 \\ \text{and } Pr(|\hat{r}_{\mathcal{A}(\vec{z})}(\vec{z}) - \hat{r}_{\hat{h}^*}(\vec{z})| < \alpha/3) > 1 - \delta/2 \\ \text{then } Pr(|r_{\mathcal{A}(\vec{z})}(p) - r_{h^*}(p)| < \alpha) > 1 - \delta\end{gathered}$$

The final result states that output hypothesis $\mathcal{A}(\vec{z})$ has true risk α-close to the lowest true risk in $\mathcal{H}$ attained by h^* with high probability $1 - \delta$ over the choice of training samples. The first pre-condition states that it must be the case that for all $h \in \mathcal{H}$ the difference between empirical risk and true risk is $\frac{\alpha}{3}$-close with high probability $1 - \frac{\delta}{2}$. The second pre-condition states that the output $\mathcal{A}(\vec{z})$ of the ERM algorithm $\mathcal{A}$ should have empirical risk $\frac{\alpha}{3}$-close to the lowest empirical risk of $\hat{h}^*$ with high probability $1 - \frac{\delta}{2}$. A hypothesis class $\mathcal{H}$ is called (α, δ)-PAC learnable if there exists an ERM algorithm $\mathcal{A}$ such that $\mathcal{H}$ and $\mathcal{A}$ satisfy the two pre-conditions. In this work, our empirical risk minimizing algorithms find $\hat{h}^*$ exactly (upto precision of convex solvers, see Section 6), thus, satisfying the second pre-condition; hence, we will focus more on the first pre-condition. As the empirical risk estimate gets better with increasing samples, a minimum number of samples are required to ensure that the first pre-condition holds (see Theorem 1). Hence we can relate (α, δ)-PAC learnability to the number of samples.

Modeling security games: Having given an example for generalized SUQR, we systematically model learning of adversary behavior in SSGs using the PAC framework for any hypothesis class $\mathcal{H}$. We assume certain properties of functions $h \in \mathcal{H}$ that we present below. First, the vector valued function $h \in \mathcal{H}$ takes the form

$$h(x) = \langle h_1(x), \ldots, h_{T-1}(x) \rangle.$$

Thus, A is the product space $A_1 \times \ldots, A_{T-1}$. Each $h_i(x)$ is assumed to take values between $[-\frac{M}{2}, \frac{M}{2}]$, where $M >> 1$,

[3]This general form is harder to analyze than the *standard* SUQR form in which the exponent function (function of x_i, R_i, P_i) for all q_i is same: $w_1 x_i + w_2 R_i + w_3 P_i$. For completeness, we derive the results for the standard SUQR form in the Appendix.

which implies $A_i = [-\frac{M}{2}, \frac{M}{2}]$. The prediction probabilities induced by any h is $q^h(x) = \langle q_1^h(x), \ldots, q_T^h(x)\rangle$, where $q_i^h(x) = \frac{e^{h_i(x)}}{1+\sum_i e^{h_i(x)}}$ (assume $h_T(x) = 0$). Next, we specify two classes of functions that we analyze in later sections. We choose these two functions classes because (1) the first function class represents the widely used SUQR model in literature [26, 35] and (2) the second function class is very flexible as it capture a wide range of functions and only imposes minimal Lipschitzness constraints to ensure that the functions are well behaved (e.g., continuous).

Parametric $\mathcal{H}$: In this approach we model generalized SUQR. Generalizing from Example 1, the functions $h \in \mathcal{H}$ take a parametric form where each component function is $h_i(x) = w_1 x_{iT} + c_{iT}$.

Non-parametric Lipschitz (NPL) $\mathcal{H}$: Here, the only restriction we impose on functions $h \in \mathcal{H}$ is that each component function h_i is L-Lipschitz where $L \leq \hat{K}$, for given and fixed constant $\hat{K}$. We show later (Lemma 7) that this implies that q^h is Lipschitz also.

Next, given the stochastic nature of the adversary's attacks, we use a loss function (same for parametric and NPL) such that minimizing the empirical risk is equivalent to maximizing the likelihood of seeing the attack data. The loss function $l : Y \times A \to \mathbb{R}$ for actual outcome t_i is defined as:

$$l(\mathrm{t}_i, a) = -\log \frac{e^{a_i}}{1 + \sum_{j=1}^{T-1} e^{a_j}} \tag{1}$$

It can be readily inferred that minimizing the empirical risk (recall $\hat{r}_h(\vec{z}) = 1/m \sum_{i=1}^m l(y_i, h(x_i))$) is equivalent to maximizing the log likelihood of the training data.

5. SAMPLE COMPLEXITY

In this section we derive the sample complexity for the parametric and NPL case, which provides an indication about the amount of data required to learn the adversary behavior. First, we present a general result about sample complexity bounds for any $\mathcal{H}$, given our loss l. This result relies on sample complexity results in [14]. The bound depends on the capacity $\mathcal{C}$ of $\mathcal{H}$, which we define after the theorem. The bound also assumes an ERM algorithm which we present for our models in Section 6.

THEOREM 1. *Assume that the hypothesis space $\mathcal{H}$ is permissible*[4]*. Let the data be generated by m independent draws from $X \times Y$ according to p. Then, assuming existence of an ERM algorithm and given our loss l defined in Eq. 1, the least m required to ensure (α, δ)-PAC learnability is (recall $\bar{d_{l_1}}$ is average l_1 distance)*

$$\frac{576M^2}{\alpha^2}\Big(\log\frac{1}{\delta} + \log\big(8\mathcal{C}(\frac{\alpha}{96T}, \mathcal{H}, \bar{d_{l_1}})\big)\Big)$$

PROOF SKETCH. Haussler [14] present a result of the above form using a general distance metric defined on the space A for any loss function l: $\rho(a,b) = \max_{y\in Y} |l(y,a) - l(y,b)|$. The main effort in this proof is relating ρ to $\bar{d_{l_1}}$ for our choice of the loss function l given by Equation 1. We are able to show that $\rho(a,b) \leq 2T\bar{d_{l_1}}(a,b)$. Our result then follows from this relation. □

The above sample complexity result is stated in terms of the capacity $\mathcal{C}(\alpha/96T, \mathcal{H}, \bar{d_{l_1}})$. Thus, in order to obtain the sample complexity of the generalized SUQR and NPL function spaces we need to compute the capacity of these function spaces. Therefore, in the rest of this section we will focus on computing capacity $\mathcal{C}(\alpha/96T, \mathcal{H}, \bar{d_{l_1}})$ for both the generalized SUQR and NPL hypothesis space. First, we need to define capacity $\mathcal{C}$ of functions spaces, for which we start by defining the covering number $\mathcal{N}$ of function spaces. Let d be a pseudo metric for the set $\mathcal{H}$. For any $\epsilon > 0$, an ϵ-cover for $\mathcal{H}$ is a finite set $\mathcal{F} \subseteq \mathcal{H}$ such that for any $h \in \mathcal{H}$ there is a $f \in \mathcal{F}$ with $d(f,h) \leq \epsilon$, i.e., any element in $\mathcal{H}$ is at least ϵ-close to some element of the cover $\mathcal{F}$. The *covering number* $\mathcal{N}(\epsilon, \mathcal{H}, d)$ denotes the size of the smallest ϵ-cover for set $\mathcal{H}$ (for the pseudo metric d). We now proceed to define a pseudo metric $d_{L^1(P,d)}$ on $\mathcal{H}$ with respect to any probability measure P on X and any given pseudo-metric d on A. This pseudo-metric is the expected (over P) distance (with d) between the output of f and g.

$$d_{L^1(P,d)}(f,g) = \int_X d(f(x), g(x))\, dP(x) \quad \forall f, g \in \mathcal{H}$$

Then, $\mathcal{N}(\epsilon, \mathcal{H}, d_{L^1(P,d)})$ is the covering number for $\mathcal{H}$ for the pseudo metric $d_{L^1(P,d)}$. However, to be more general, the capacity of function spaces provides a "distribution-free" notion of covering number. The *capacity* $\mathcal{C}(\epsilon, \mathcal{H}, d)$ is:

$$\mathcal{C}(\epsilon, \mathcal{H}, d) = \sup_P \{\mathcal{N}(\epsilon, \mathcal{H}, d_{L^1(P,d)})\}$$

Capacity of vector valued function: The function spaces (both parametric and NPL) we consider are vector valued. Haussler [14] provides an useful technique to bound the capacity for vector valued function space $\mathcal{H}$ in terms of the capacity of each of the component real valued function space. Given k functions spaces $\mathcal{H}_1, \ldots, \mathcal{H}_k$ with functions from X to A_i, he define the *free product* function space $\times_i \mathcal{H}_i$ with functions from X to $A = A_1 \times \ldots A_k$ as $\times_i \mathcal{H}_i = \{\langle h_1, \ldots, h_k\rangle \mid h_i \in \mathcal{H}_i\}$, where $\langle h_1, \ldots, h_k\rangle(x) = \langle h_1(x), \ldots, h_k(x)\rangle$. He shows that:

$$\mathcal{C}(\epsilon, \times_i \mathcal{H}_i, \bar{d_{l_1}}) < \prod_{i=1}^{k} \mathcal{C}(\epsilon, \mathcal{H}_i, d_{l_1}) \tag{2}$$

Unfortunately, a straightforward application of the above result does not give as tight bounds for capacity in the parametric case as the novel direct sum decomposition of function spaces approach we use in the sub-section. Even for the NPL case where the above result is used we still need to compute $\mathcal{C}(\epsilon, \mathcal{H}_i, d_{l_1})$ for each component function space $\mathcal{H}_i$.

5.1 Parametric case: Generalized SUQR

Recall that the hypothesis function h has $T-1$ component functions $w_1 x_{iT} + c_{iT}$. However, the same weight w_1 in all component functions implies that $\mathcal{H}$ is not a free product of component function spaces, hence we cannot use Equation 2 directly. However, if we consider the space of functions, say $\mathcal{H}'$, in which the i^{th} component function space $\mathcal{H}'_i$ is given by $w_i x_{iT} + c_{iT}$ (note w_i can be different for each i) then we can use Equation 2 to bound $\mathcal{C}(\epsilon, \mathcal{H}', \bar{d_{l_1}})$. Also, the fact that $\mathcal{H} \subset \mathcal{H}'$ allows upper bounding $\mathcal{C}(\epsilon, \mathcal{H}, \bar{d_{l_1}})$ by $\mathcal{C}(\epsilon, \mathcal{H}', \bar{d_{l_1}})$. But, this approach results in a weaker $T\log(\frac{T}{\alpha}\log\frac{T}{\alpha})$ bound (detailed derivation using this approach is in Appendix) than the technique we use below. We obtain a $T\log(\frac{T}{\alpha})$ result below in Theorem 2.

We propose a *novel approach that decomposes $\mathcal{H}$ into a direct sum of two functions spaces* (defined below), each of which capture the simpler functions $w_1 x_{iT}$ and c_{iT} respectively. We provide a general result about such decomposition which allows us to bound $\mathcal{C}(\epsilon, \mathcal{H}, \bar{d_{l_1}})$. We start with the following definition.

DEFINITION 1. *Direct-sum semi-free product of function spaces $\mathcal{G} \subset \times_i \mathcal{G}_i$ and $\times_i \mathcal{F}_i$ is defined as $\mathcal{G} \oplus \times_i \mathcal{F}_i = \{\langle g_1 + f_1, \ldots, g_{T-1} + f_{T-1}\rangle \mid \langle g_1, \ldots, g_{T-1}\rangle \in \mathcal{G}, \langle f_1, \ldots, f_{T-1}\rangle \in \times_i \mathcal{F}_i\}$.*

[4]As noted in Haussler: "This is a measurability condition defined in Pollard (1984) which need not concern us in practice."

Applying the above definition for our case, $\mathcal{G}_i$ contains functions of the form wx_{iT} (w taking different values for different $g_i \in \mathcal{G}_i$). A function $\langle g_1, \ldots, g_{T-1}\rangle \in \times_i \mathcal{G}_i$ can have different weights for each component g_i, and thus we consider the subset $\mathcal{G} = \{\langle g_1, \ldots, g_{T-1}\rangle \mid \langle g_1, \ldots, g_{T-1}\rangle \in \times_i \mathcal{G}_i$, same coefficient w for all $g_i\}$. $\mathcal{F}_i$ contains constant valued functions of the form c_{iT} (c_{iT} different for different functions $f_i \in \mathcal{F}_i$). Then, $\mathcal{H} = \mathcal{G} \oplus \times_i \mathcal{F}_i$. Next, we prove a general result about direct-sum semi-free products:

LEMMA 1. *If $\mathcal{H}$ is the direct-sum semi-free product $\mathcal{G} \oplus \times_i \mathcal{F}_i$*

$$\mathcal{C}(\epsilon, \mathcal{H}, \bar{d_{l_1}}) < \mathcal{C}(\epsilon/2, \mathcal{G}, \bar{d_{l_1}}) \prod_{i=1}^{T-1} \mathcal{C}(\epsilon/2, \mathcal{F}_i, d_{l_1})$$

PROOF. Fix any probability distribution over X, say P. For brevity, we write k instead of $T-1$. Consider an $\epsilon/2$-cover U_i for each $\mathcal{F}_i$; also let V be an $\epsilon/2$-cover for $\mathcal{G}$. We claim that $V \oplus \times_i U_i$ is an ϵ-cover for $\mathcal{G} \oplus \times_i \mathcal{F}_i$. Take any function $h = \langle g_1 + f_1, \ldots g_k + f_k\rangle$. Find functions $f_i' \in U_i$ such that $d_{L^1(P,d_{l_1})}(f_i, f_i') < \epsilon/2$. Similarly, find function $g' = \langle g_1', \ldots g_k'\rangle \in V$ such that $d_{L^1(P,\bar{d_{l_1}})}(g, g') < \epsilon/2$ where $g = \langle g_1, \ldots g_k\rangle$. Let $h' = \langle g_1' + f_1', \ldots g_k' + f_k'\rangle$. Then,

$$\begin{aligned}
&d_{L^1(P,\bar{d_{l_1}})}(h, h') \\
&= \int_X \frac{1}{k} \sum_{i=1}^{k} d_{l_1}(g_i(x) + f_i(x), g_i'(x) + f_i'(x))\, dP(x) \\
&\leq \int_X \frac{1}{k} \sum_{i=1}^{k} d_{l_1}(g_i(x), g_i'(x)) + d_{l_1}(f_i(x), f_i'(x))\, dP(x) \\
&= d_{L^1(P,\bar{d_{l_1}})}(g, g') + \frac{1}{k} \sum_{i=1}^{k} d_{L^1(P,d_{l_1})}(f_i, f_i') \\
&< \epsilon/2 + \epsilon/2 = \epsilon
\end{aligned}$$

Thus, the size of ϵ-cover for $\mathcal{G} \oplus \times_i \mathcal{F}_i$ is bounded by $|V| \prod_i |U_i|$.

$$\begin{aligned}
&\mathcal{N}(\epsilon, \mathcal{G} \oplus \times_i \mathcal{F}_i, d_{L^1(P,\bar{d_{l_1}})}) < |V| \prod_i |U_i| \\
&\quad = \mathcal{N}(\epsilon/2, \mathcal{G}, d_{L^1(P,\bar{d_{l_1}})}) \prod_{i=1}^{k} \mathcal{N}(\epsilon/2, \mathcal{F}_i, d_{L^1(P,d_{l_1})})
\end{aligned}$$

Taking sup over probability distribution P on both sides of the above inequality we get our desired result about capacity. □

Next, we need to bound the capacity of $\mathcal{G}$ and $\mathcal{F}_i$ for our case. We assume the range of all these functions (g_i, f_i) to be $[-\frac{M}{4}, \frac{M}{4}]$ (so that their sum h_i lies in $[-\frac{M}{2}, \frac{M}{2}]$). We can obtain sharp bounds on the capacities of $\mathcal{G}$ and $\mathcal{F}_i$ decomposed from $\mathcal{H}$ in order to obtain sharp bounds on the overall capacity.

LEMMA 2. *$\mathcal{C}(\epsilon, \mathcal{G}, \bar{d_{l_1}}) \leq M/4\epsilon$ and $\mathcal{C}(\epsilon, \mathcal{F}_i, d_{l_1}) \leq M/4\epsilon$.*

PROOF SKETCH. Note that $x_{iT} = x_i - x_T$ lies between $[-1, 1]$. Then, for any two $g, g' \in \mathcal{G}$ we can prove following result: $d_{L^1(P,\bar{d_{l_1}})}(g, g') \leq |(w - w')|$ (details in Appendix). Also, note that since the range of any $g = w(x_i - x_T)$ is $[-\frac{M}{4}, \frac{M}{4}]$ and given $x_i - x_T$ lies between $[-1, 1]$, we can claim that w lies between $[-\frac{M}{4}, \frac{M}{4}]$. Thus, as the distance between functions is bounded by the difference in weights, it is enough to divide the $M/2$ range of the weights into intervals of size 2ϵ and consider functions at the boundaries. Hence the ϵ-cover has at most $M/4\epsilon$ functions.

The proof for constant valued functions $\mathcal{F}_i$ is similar, since the distance between two functions in this space is the absolute difference in the constants, which lie in $[-\frac{M}{4}, \frac{M}{4}]$. □

Then, plugging the result of Lemma 2 (substituting $\epsilon/2$ for ϵ) into Lemma 1 we obtain $\mathcal{C}(\epsilon, \mathcal{H}, \bar{d_{l_1}}) < (M/2\epsilon)^T$. Having bounded $\mathcal{C}(\epsilon, \mathcal{H}, \bar{d_{l_1}})$, we use Theorem 1 to obtain:

THEOREM 2. *The generalized SUQR parametric hypothesis class $\mathcal{H}$ is (α, δ)-PAC learnable with sample complexity*[5]

$$O\Big(\big(\frac{1}{\alpha^2}\big)\big(\log(\frac{1}{\delta}) + T\log(\frac{T}{\alpha})\big)\Big)$$

The above result shows a modest $T \log T$ growth of sample complexity with increasing targets, suggesting the parametric approach can avoid overfitting limited data with increasing number of targets; however, the simplicity of the functions captured by this approach (compared to NPL) results in lower accuracy with increasing data, as shown later in our experiments on real-world data.

5.2 Non-Parametric Lipschitz case

Recall that $\mathcal{H}$ for the NPL case is defined such that each component function h_i is L-Lipschitz where $L \leq \hat{K}$. Consider the functions spaces $\mathcal{H}_i$ consisting of real valued L-Lipschitz functions where $L \leq \hat{K}$. Then, $\mathcal{H} = \times_i \mathcal{H}_i$. Then, using Equation 2: $\mathcal{C}(\epsilon, \mathcal{H}, \bar{d_{l_1}}) \leq \prod_{i=1}^{T-1} \mathcal{C}(\epsilon, \mathcal{H}_i, d_{l_1})$.

Next, our task is to bound $\mathcal{C}(\epsilon, \mathcal{H}_i, d_{l_1})$. Consider the sup-distance metric between real valued functions: $d_{l_\infty}(h_i, h_i') = \sup_X |h_i(x) - h_i'(x)|$ for $h_i, h_i' \in \mathcal{H}_i$. Note that d_{l_∞} is independent of any probability distribution P, and for all functions h_i, h_i' and any P, $d_{L^1(P,d_{l_1})}(h_i, h_i') \leq d_{l_\infty}(h_i, h_i')$. Thus, we can infer [14] that for all P, $\mathcal{N}(\epsilon, \mathcal{H}_i, d_{L^1(P,d_{l_1})}) \leq \mathcal{N}(\epsilon, \mathcal{H}_i, d_{l_\infty})$ and then taking sup over P (recall $\mathcal{C}(\epsilon, \mathcal{H}_i, d_{l_1}) = \sup_P \{\mathcal{N}(\epsilon, \mathcal{H}_i, d_{L^1(P,d_{l_1})})\}$) we get

$$\mathcal{C}(\epsilon, \mathcal{H}_i, d_{l_1}) \leq \mathcal{N}(\epsilon, \mathcal{H}_i, d_{l_\infty}) \tag{3}$$

We bound $\mathcal{N}(\epsilon, \mathcal{H}_i, d_{l_\infty})$ in terms of covering number for X (recall $X = \{x \mid x \in [0, 1]^T, \sum_i x_i \leq K\}$) using results from [32].

LEMMA 3. $\mathcal{N}(\epsilon, \mathcal{H}_i, d_{l_\infty}) \leq \Big(2\Big\lceil \frac{M}{\epsilon} \Big\rceil + 1\Big) \cdot 2^{\mathcal{N}(\frac{\epsilon}{2\hat{K}}, X, d_{l_\infty})}$

To use the above result, we still need to bound $\mathcal{N}(\epsilon, X, d_{l_\infty})$. We do so by combining two remarkable results about Eulerian number $\left\langle {T \atop k} \right\rangle$ [17] (k has to be integral).

- Laplace [10] [29] discovered that the volume of $X_k = \{x | x \in [0, 1]^T, k - 1 \leq \sum_i x_i \leq k\}$ is $\left\langle {T \atop k} \right\rangle / T!$. Thus, if $X_K = \cup_{k=1}^{K} X_k$, then $vol(X_K) = \sum_{k=1}^{K} vol(X_k) = \sum_{k=1}^{K} \left\langle {T \atop k} \right\rangle / T!$.
- Also, it is known [31] that $\left\langle {T \atop k} \right\rangle / T! = F_T(k) - F_T(k-1)$, where $F_T(x)$ is the CDF of the probability distribution of $S_T = U_1 + \ldots + U_T$ and each U_i is a uniform random variable on $[0, 1)$.

Combining these results, $vol(X_{K+1}) = F_T(K+1)$. The volume of a l_∞ ball of radius ϵ (l_∞ ball is a hypercube) is $(2\epsilon)^T$ [34]. Then, the number of balls that fit tightly (aligned with the axes) and completely inside X_{K+1} is bounded by $F_T(K+1)/(2\epsilon)^T$. Since $\epsilon << 1$, these balls cover $X_K = X$ completely and the tight packing ensures that the center of the balls forms an ϵ-cover for X. Then, bounding $F_T(K+1)$ using Bernstein's inequality about concentration of random variables we get:

LEMMA 4. *For $K + 1 \leq 0.5T$ (recall $K << T$)*

$$\mathcal{N}(\epsilon, X, d_{l_\infty}) \leq e^{\frac{-3T(0.5-(K+1)/T)^2}{1-(K+1)/T}} / (2\epsilon)^T$$

Plugging the above result into Lemma 3 and then using that in Equation 3, we bound $\mathcal{C}(\epsilon, \mathcal{H}_i, d_{l_1})$. Finally, Equation 2 gives a bound on $\mathcal{C}(\epsilon, \mathcal{H}, \bar{d_{l_1}})$ that we use in Theorem 1 to obtain

[5] In the Appendix, we show that for standard SUQR (simpler than our generalized SUQR) the sample size is $O\big(\frac{1}{\alpha^2}(\log\frac{1}{\delta} + \log\frac{T}{\alpha})\big)$

THEOREM 3. *The non-parametric hypothesis class $\mathcal{H}$ is a (α, δ)-PAC learnable with sample complexity*

$$O\Big(\big(\frac{1}{\alpha^2}\big)\big(\log(\frac{1}{\delta}) + (\frac{T^{T+1}}{\alpha^T})\big)\Big)$$

The above result shows that the sample complexity for NPL grows fast with T suggesting that NPL may not be the right approach to use when the number of targets is large.

6. LEARNING ALGORITHM

As stated earlier, our loss function was designed so that the learning algorithm (empirical risk minimizer in PAC framework) was the same as maximizing log likelihood of data. Indeed, for SUQR, the standard MLE approach can be used to learn the parameters (weights) and has been used in literature [26]. However, for NPL, which has no parameters, maximizing likelihood only provides $h(x)$ for those mixed strategies x that are in the training data.

Hence we present a *novel two step learning algorithm for the NPL case*. In the first step, we estimate the most likely value for $h_i(x)$ (for each i) for each x in the training data, ensuring that for any pair x, x' in the training data, $|h_i(x) - h_i(x')| \leq \hat{K}||x - x'||_1$. In the second step, we construct the function h_i with the least Lipschitz constant subject to the constraint that h_i takes the values for the training data output by the first step.

More formally, assume the training data has s unique values for x in the training set and let these values be $x^1, \ldots, x^s$. Further, let there be n_j distinct data points against x^j, i.e., n_j attacks against mixed strategy x^j. Denote by $n_{j,i}$ the number of attacks at each target i when x^j was used. Let h_{ij} be the variable that stands for the estimate of value $h_i(x^j)$; $i \in \{1, \ldots, T\}$, $j \in \{1, \ldots, s\}$. Fix $h_{Tj} = 0$ for all j. Then, probability of attack on target i against mixed strategy x^j is given by $q_{ij} = \frac{e^{h_{ij}}}{\sum_i e^{h_{ij}}}$. Thus, the log likelihood of the training data is $\sum_{j=1}^{s}\sum_{i=1}^{T} n_{j,i} \log q_{ij}$. Let $Lip(\hat{K})$ denote the set of L-Lipschitz functions with $L \leq \hat{K}$. Using our assumption that $h_i \in Lip(\hat{K})$, the following optimization problem provides the most likely h_{ij} :

$$\begin{aligned} \max_{h_{ij}} \quad & \sum_{j=1}^{s}\sum_{i=1}^{T} n_{j,i} \log \frac{e^{h_{ij}}}{\sum_i e^{h_{ij}}} \\ \text{subject to} \quad & \forall i, j, j',\ |h_{ij} - h_{ij'}| \leq \hat{K}||x^j - x^{j'}||_1 \\ & \forall i, j,\ -M/2 \leq h_{ij} \leq M/2 \end{aligned}$$

Given solution h^*_{ij} to the above problem, we wish to construct the solution h_i such that its Lipschitz constant (given by K_{h_i}) is the lowest possible subject to h_i taking the value h^*_{ij} for x^j. Such a construction provides the most smoothly varying solution given the training data, i.e., we do not assume any more sharp changes in the adversary response than what the training data provides.

$$\min_{h_i \in Lip(\hat{K})} K_{h_i} \ \text{ subject to } \ \forall i, j.\, h_i(x^j) = h^*_{ij} \quad (\mathsf{MinLip})$$

The above optimization is impractical to solve computationally as uncountably many constraints are required to relate K_{h_i} to h_i, Fortunately, we obtain an analytical solution:

LEMMA 5. *The following is a solution for problem* MinLip

$$h_i(x) = \min_j \{h^*_{ij} + K^*_i||x - x^j||_1\}$$

where $K^*_i = \max_{j,j':j\neq j'} \frac{|h^*_{ij} - h^*_{ij'}|}{||x^j - x^{j'}||_1}$

PROOF SKETCH. Observe that due to the definition of K^* any solution to MinLip will have Lipschitz constant $\geq K^*$. Thus, it suffices to show that the Lipschitz constant of h_i is K^*, to prove that h_i is a solution of MinLip, which we show in the Appendix. □

Note that for any point x^j is the training data we have $h_i(x^j) = h^*_{ij}$. Then the value of $h_i(x)$ for a x not in the training set and close to x^j is quite likely be the $h_i(x^j)$ plus the scaled distance $K^*_i||x - x^j||_1$ showing the value of x is influenced by nearby training points.

7. UTILITY BOUNDS

Next, we bound the difference between the optimal utility and the utility derived from planning using the learned h. The utility bound is same for the parametric and NPL case. Recall that the defender receives the utility $xUq^p(x)$ when playing strategy x. We need to bound the difference between the true distribution $q^p(x)$ and the predicted distribution $q^h(x)$ of attacks in order to start analyzing bounds on utility. Thus, we transform the PAC learning guarantee about the risk of output h to a bound on $||q^p(x) - q^h(x)||_1$. As the PAC guarantee only bounds the risk between predicted h and the best hypothesis h^* in $\mathcal{H}$, in order to relate the true distribution q^p and predicted distribution q^h, the lemma below assumes a bounded KL divergence between the distribution of the best hypothesis q^{h^*} and the true distribution q^p.

LEMMA 6. *Assume $E[KL(q^p(x) \,||\, q^{h^*}(x))] \leq \epsilon^*$. Given an ERM $\mathcal{A}$ with output $h = \mathcal{A}(\vec{z})$ and guarantee $Pr(|r_h(p) - r_{h^*}(p)| < \alpha) > 1 - \delta$, with prob. $\geq 1 - \delta$ over training samples $\vec{z}$ we have*

$$Pr(||q^p(x) - q^h(x)||_1 \leq \sqrt{2}\Delta) \geq 1 - \Delta$$

where $\Delta = (\alpha + \epsilon^*)^{1/3}$ and x is sampled using density p.

Utility bound: Next, we provide an utility bound, given the above guarantee about learned h. Let the optimal strategy computed using the learned adversary model h be $\tilde{x}$, i.e., $\tilde{x}^T U q^h(\tilde{x}) \geq x'^T U q^h(x')$ for all x'. Let the true optimal defender mixed strategy be x^* (optimal w.r.t. true attack distribution $q^p(x)$), so that the maximum defender utility is $x^{*T} U q^p(x^*)$. Let $B(x, \epsilon)$ denote the l_1 ball of radius ϵ around x. We make the following assumptions:

1. h_i is $\hat{K}$-Lipschitz $\forall i$ and q^p is K-Lipschitz in l_1 norm.
2. $\exists$ small ϵ such that $Pr(x \in B(x^*, \epsilon)) > \Delta$ over choice of x using p.
3. $\exists$ small ϵ such that $Pr(x \in B(\tilde{x}, \epsilon)) > \Delta$ over choice of x using p.

While the first assumption is mild, the last two assumptions for small ϵ mean that the points x^* and $\tilde{x}$ must not lie in low density regions of the distribution p used to sample the data points. In other words, there should be many defender mixed strategies in the data of defender-adversary interaction that lie near x^* and $\tilde{x}$. We discuss the assumptions in details after the technical results below. Given these assumptions, we need Lemma 7 that relates assumption (1) to Lipschitzness of q^h in order to obtain the utility bound.

LEMMA 7. *If h_i is $\hat{K}$-Lipschitz then $\forall x, x' \in X$. $||q^h(x) - q^h(x')||_1 \leq 3\hat{K}||x - x'||_1$, i.e., $q^h(x)$ is $3\hat{K}$-Lipschitz.*

Then, we can prove the following:

THEOREM 4. *Given above assumptions and the results of Lemma 6 and 7, with prob. $\geq 1 - \delta$ over the training samples the expected utility $\tilde{x}^T U q^p(\tilde{x})$ for the learned h is at least*

$$x^{*T} U q^p(x^*) - (K + 1)\epsilon - 2\sqrt{2}\Delta - 6\hat{K}\epsilon$$

Discussion of assumptions: A puzzling phenomenon observed in recent work on learning in SSGs is that good prediction accuracy of the learned adversary behavior is not a reliable indicator of the defender's performance in practice [12]. The additional assumptions, over and above the PAC learning guarantee, that are made to bound the utility deviation from the optimal utility point towards the possibility of such occurrences. Recall that the second assumption requires the existence of many defender mixed strategies in the dataset near the utility optimal strategy x^*. Of course x^* is not known apriori, hence in order to guarantee utility close to the highest possible utility the dataset must contain defender mixed strategies from all regions of the mixed strategy space; or at-least if it is known that some regions of the mixed strategies dominate other parts in terms of utility then it is enough to have mixed strategies from these regions. Thus, following our assumption, better utility can be achieved by collecting attack data against a variety of mixed strategies rather than many attacks against few mixed strategies.

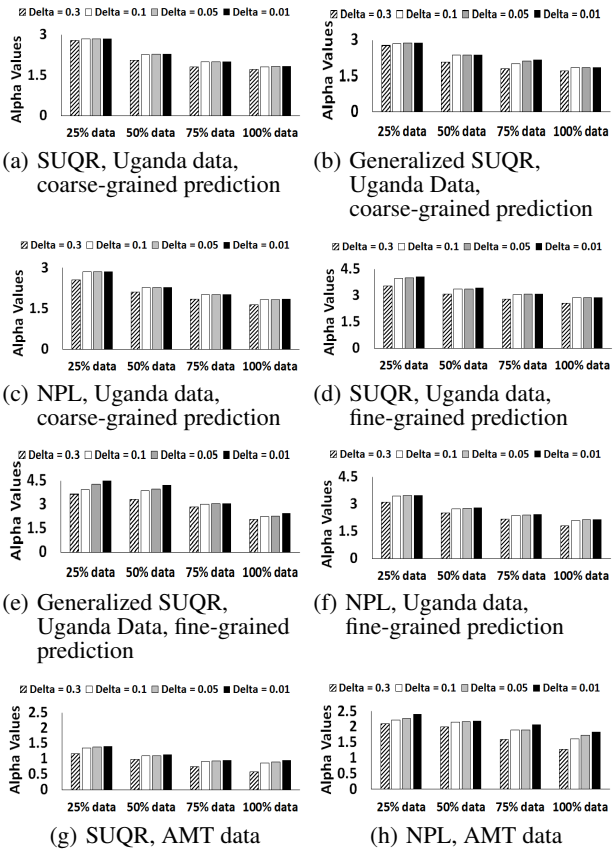
Going further, we illustrate with a somewhat extreme example where violating our assumptions can lead to this undesirable phenomenon. Consider the extreme example where probability distribution p (recall data points are sampled using p) puts all probability mass on x_0, where the the utility for x_0 is much lower than x^*. Hence, the dataset will contain only one defender mixed strategy x_0 (with many attacks against it). Due to Lipschitzness (assumption 1), the large utility difference between x_0 and x^* implies that x_0 is not close to x^* which in turn violates assumption 2. This example provides a very good PAC guarantee since there is no requirement for the learning algorithm to predict accurately for any other mixed strategies (which occur with zero probability) in order to have good prediction accuracy. The learning technique needs to predict well only for x_0 to achieve a low α, δ. As a result the defender strategy computed against the learned adversary model may not be utility maximizing because of the poor predictions for all defender mixed strategies other than the low utility yielding x_0. More generally, good prediction accuracy can be achieved by good predictions only for the mixed strategies that occur with high probability.

Indeed, in general, the prediction accuracy in the PAC model (and any applied machine learning approach) is not a reliable indicator of good prediction over the entire space of defender mixed strategies unless, following our assumption 2, the dataset has attacks against strategies from all parts of the mixed strategy space. However, in past work [1, 9] researchers have focused on gathering a lot of attack data but on limited number of defender strategies. Our analysis provides a principled explanation of the issues that arise from composing the machine learning module with the defender strategy optimization module in the learning and planning framework used for SSGs. Further, we believe that our analysis provides guidance to overcome these issues and discover the defender's utility maximizing strategy.

8. EXPERIMENTAL RESULTS

We show experimental results on two datasets: (i) real-world poaching data from QENP [25]; (ii) data from human subjects experiments on Amazon Mechanical Turk (AMT) [15], to estimate prediction errors and the amount of data required to reduce the error for both the parametric and NPL learning settings. Also, we compare the NPL approach with both the standard as well as the generalized SUQR approach and show that: (i) the NPL approach, while computationally slow, outperforms the standard SUQR model for Uganda data; and (ii) the performance of generalized SUQR is in between NPL and standard SUQR.

For each dataset, we conduct four experiments with 25%, 50%, 75% and 100% of the original data. We create 100 train-test splits



(a) SUQR, Uganda data, coarse-grained prediction

(b) Generalized SUQR, Uganda Data, coarse-grained prediction

(c) NPL, Uganda data, coarse-grained prediction

(d) SUQR, Uganda data, fine-grained prediction

(e) Generalized SUQR, Uganda Data, fine-grained prediction

(f) NPL, Uganda data, fine-grained prediction

(g) SUQR, AMT data

(h) NPL, AMT data

Figure 1: Results on Uganda and AMT datasets for both the parametric and NPL learning settings.

in each of the four experiments per dataset. For each train-test split we compute the average prediction error α (average difference between the log-likelihoods of the attacks in the test data using predicted and actual attack probabilities). We report the α value at the $1-\delta$ percentile of the 100 α values, e.g., reported $\alpha = 2.09$ for $\delta = 0.1$ means that 90 of the 100 test splits have $\alpha < 2.09$.

8.1 Real-World Poaching data

We first present results of our experiments with real-world poaching data. The dataset obtained contained information about features such as ranger patrols and animal densities, which are used as features in our SUQR model, and the poachers' attacks, with 40,611 total observations recorded by rangers at various locations in the park. The park area was discretized into 2423 grid cells, with each grid cell corresponding to a 1km^2 area within the park. After discretization, each observation fell within one of 2423 target cells and the animal densities and the number of poaching attacks within each target cell were then aggregated. The dataset contained 655 poachers' attacks in response to the defender's strategy for 2012 at QENP. Although the data is reliable because the rangers recorded the latitudes and longitudes of the location for each observation using a GPS device, it is important to note that this data set is extremely noisy because of: (i) Missing observations: all the poaching events are not recorded because the limited number of rangers cannot patrol all the areas in this park all the time; (ii) Uncertain feature values: the animal density feature is also based on incomplete observations of animals; (iii) Uncertain defender strategy: the

actual defender mixed strategy is unknown; we estimate the mixed strategies based on the provided patrol data.

In this paper, we provide two types of prediction in our experiments: (i) fine-grained and (ii) coarse-grained. First, to provide a baseline for our error measures, we use the same coarse-grained prediction approach as reported by Nguyen et al. [25], in which the authors only predict whether a target will be attacked or not. The results for coarse-grained predictions with our performance metric (α values for different δ) are shown in Figs. 1(a), 1(b) and 1(c). Next, in the fine-grained prediction approach we predict the *actual* number of attacks on each target in our test set; these results are shown in Figs. 1(d), 1(e) and 1(f). In [25], the authors used a particular metric for prediction performance called the area under the ROC curve (AUC), which we will discuss later in the section.

From our fine-grained and coarse-grained prediction approaches, we make several important observations. First, we observe that α decreases with increasing sample size at a rate proportional to $\frac{1}{\sqrt{m}}$, where m is the number of samples. For example, in Fig. 1(a), the α values corresponding to $\delta = 0.01$ (black bar) for the 25%, 50%, 75% and 100% data are 2.81, 2.38, 2.18 and 1.8 respectively, which fits $\frac{1}{\sqrt{m}}$with a goodness-of-fit, i.e., r^2=0.97. This observation supports the relationship between α and m shown in Theorem 3 and can be used to approximately infer the number of samples required to reduce the prediction error to a certain value. For example, assuming we collect same number of samples (=655) per year, to reduce α from 1.8 to 1.64, we would require two more years of data. Note here that α is in log-scale and hence the decrease is significant. It is also worth noting here that, for a random classifier, we observe a value of $\alpha = 6.3$ for $\delta = 0.01$ while performing coarse grained predictions with 25% data. This is more than $\alpha = 2.81$ for $\delta = 0.01$ obtained for our standardized SUQR model while performing coarse grained predictions with 25% data. As α is in the log-scale, the increase in error is actually more than two-fold.

Our second observation is that α values for fine-grained predictions (e.g., 2.9 for $\delta = 0.01$ and 100% data for the standardized SUQR model in Fig. 1(d)) are understandably higher than the corresponding values for coarse-grained predictions (1.8 for $\delta = 0.01$ and 100% data for SUQR in Fig. 1(a)) because in the fine-grained case we predict the exact number of attacks.

Third, we observe that the performance of the generalized SUQR model (e.g., 2.47 for $\delta = 0.01$ and 100% data in Fig. 1(e)) is better in most cases than that of the standardized SUQR approach (2.9 for $\delta = 0.01$ and 100% data in Fig. 1(d)), but worse than the NPL approach (2.15 for $\delta = 0.01$ and 100% data in Fig. 1(f)).

Finally, we observe that our NPL model performs better than its parametric counterparts in predicting future poaching attacks for the fine-grained case (see example in previous paragraph), indicating that the true adversary behavior model may indeed be more complicated than what can be captured by SUQR.

Relation to previous work: In earlier work, Nguyen et al. [25] uses the area under curve (of a ROC curve) metric to demonstrate the performance of their approaches. The AUC value of 0.73 reported in their paper is an alternate view of our α, δ metric for the coarse grained prediction approach. While there has been alternate analysis in terms of measuring prediction performances with the AUC metric in earlier papers, in our paper we have shown new trends and insights with the α, δ metric through analysis from the PAC model perspective, which is missing in earlier work. We show: (i) sample complexity results and the relationship between increasing number of samples and the reduction in prediction error for each of our models; (ii) the differences in errors while learning a vector valued response function (fine-grained prediction) as opposed to classifying targets as attacked or not (coarse-grained prediction); and (iii) comparison of the performance of our new NPL model with other parametric approaches in terms of both fine-grained and coarse-grained predictions and its effectiveness on real-world poaching data which was not shown in previous work.

Uganda Parametric	Uganda NPL	AMT Parametric	AMT NPL
0.7188	121.24	0.91	123.4

Table 2: Runtime results (in secs.) for one train-test split

8.2 AMT Data

Here we show fine-grained prediction results on real-world AMT data obtained from [15] to demonstrate the performance of both our approaches on somewhat cleaner data. This dataset is cleaner than the Uganda data because: (i) all attacks are observed, and (ii) animal densities and deployed defender strategies are known. The dataset consisted of 16 unique mixed strategies. There were an average of 40 attack data points per mixed strategy. Each attack had been conducted by an unique individual recruited on AMT. We used attack data corresponding to 11 randomly chosen mixed strategies for training and data for the remaining mixed strategies for testing. Results are shown in Figs. 1(g) and 1(h). We observe that: (i) α values in this case are lower as compared to the Uganda data as the AMT data is cleaner; and (ii) the NPL model's performance on this dataset is poor as compared to SUQR due to, (a) low number of samples in AMT data, and (b) real-world poacher behavior may be more complicated than that of AMT participants and hence SUQR in this case was able to better capture AMT participants' behavior with limited number of samples.[6]

Runtime: While running on Matlab R2015a on an Intel Core i7-5500 CPU@2.40Ghz, 8GB RAM machine with a 64-bit Windows 10, on average, the NPL computation takes longer than the parametric setting, as shown in Table 2.

9. CONCLUSION

Over the last couple of years, a lot of work has used learning methods to learn bounded rationality adversary behavioral models, but there has been no formal study of the learning process and its implication on the defender's performance. The lack of formal analysis also means that many practical questions go unanswered. We have advanced the state of the art in learning of adversary behaviors in SSGs, in terms of their analysis and implications of such learned behaviors on defender's performance.

While we used the PAC framework, it is not an out of the box approach. We needed to come up with innovative techniques to obtain sharp bounds for our case. Furthermore, we also provide a new non-parametric approach which showed promising results with real-world data. Finally, we provided a principled explanation of why prediction accuracy is not enough to guarantee good defender performance by revealing that the PAC learning guarantee is not the guarantee required for discovering the optimal defender strategy. Finally, we hope this work leads to more theoretical work in learning of adversary models and the use of non-parametric models in the real world.

10. ACKNOWLEDGMENT

This research is supported by MURI grant W911NF-11-1-0332. We wish to acknowledge the authors of [25], specifically Andrew Plumptre, for providing us with the Uganda data.

[6]In the Appendix we show that, on simulated data, α does indeed approach zero and NPL outperforms SUQR with enough samples.

REFERENCES

[1] Y. D. Abbasi, M. Short, A. Sinha, N. Sintov, C. Zhang, and M. Tambe. Human adversaries in opportunistic crime security games: Evaluating competing bounded rationality models. In *Conference on Advances in Cognitive Systems*, 2015.

[2] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[3] M.-F. Balcan, A. Blum, N. Haghtalab, and A. D. Procaccia. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, EC '15, 2015.

[4] M.-F. Balcan, A. D. Procaccia, and Y. Zick. Learning cooperative games. In *Proceedings of the International Joint Conference on Artificial Intelligence*, IJCAI '15, 2015.

[5] N. Basilico, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 57–64. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[6] J. Blocki, N. Christin, A. Datta, A. Procaccia, and A. Sinha. Audit games. In *Proceedings of the 23rd international joint conference on Artificial Intelligence (IJCAI'13)*, 2013.

[7] J. Blocki, N. Christin, A. Datta, A. Procaccia, and A. Sinha. Audit games with multiple defender resources. In *Proceedings of the 29th AAAI conference on Artificial Intelligence (AAAI'15)*, 2015.

[8] A. Blum, N. Haghtalab, and A. D. Procaccia. Learning optimal commitment to overcome insecurity. In *Advances in Neural Information Processing Systems*, pages 1826–1834, 2014.

[9] J. Cui and R. S. John. Empirical comparisons of descriptive multi-objective adversary models in stackelberg security games. In *Decision and Game Theory for Security*, pages 309–318. Springer, 2014.

[10] M. de Laplace. Oeuvres complètes. *Oeuvres complètes*, 7:257, 1886.

[11] F. Fang, T. H. Nguyen, B. An, M. Tambe, R. Pickles, W. Y. Lam, and G. R. Clements. Towards addressing challenges in green security games in the wild (extended abstract). In *Workshop of Behavioral, Economic and Computational Intelligence for Security (BECIS) held at IJCAI*, 2015.

[12] B. Ford, T. Nguyen, M. Tambe, N. Sintov, and F. D. Fave. Beware the soothsayer: From attack prediction accuracy to predictive reliability in security games. In *Conference on Decision and Game Theory for Security*, 2015.

[13] W. Haskell, D. Kar, F. Fang, M. Tambe, S. Cheung, and L. E. Denicola. Robust protection of fisheries with compass. In *Innovative Applications of Artificial Intelligence (IAAI)*, 2014.

[14] D. Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, Sept. 1992.

[15] D. Kar, F. Fang, F. D. Fave, N. Sintov, and M. Tambe. A Game of Thrones: When human behavior models compete in repeated stackelberg security games. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2015.

[16] M. J. Kearns and U. V. Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

[17] D. E. Knuth. *The art of computer programming: sorting and searching*, volume 3. Pearson Education, 1998.

[18] D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *In Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 805–810, 2010.

[19] S. Kullback. *Information Theory and Statistics*. John Wiley and Sons, 1959.

[20] J. Letchford, V. Conitzer, and K. Munagala. Learning and approximating the optimal strategy to commit to. In *Algorithmic Game Theory*, pages 250–262. Springer, 2009.

[21] U. v. Luxburg and O. Bousquet. Distance–based classification with lipschitz functions. *The Journal of Machine Learning Research*, 5:669–695, 2004.

[22] J. Marecki, G. Tesauro, and R. Segal. Playing repeated stackelberg games with unknown opponents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 821–828, 2012.

[23] D. McFadden. Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics*, pages 105–142, 1973.

[24] D. McFadden. Quantal choice analysis: A survey. *Annals of Economic and Social Measurement*, 5(4):363–390, 1976.

[25] T. H. Nguyen, F. M. D. Fave, D. Kar, A. S. Lakshminarayanan, A. Yadav, M. Tambe, N. Agmon, A. J. Plumptre, M. Driciru, F. Wanyama, and A. Rwetsiba. Making the most of our regrets: Regret-based solutions to handle payoff uncertainty and elicitation in green security games. In *Conference on Decision and Game Theory for Security*, 2015.

[26] T. H. Nguyen, R. Yang, A. Azaria, S. Kraus, and M. Tambe. Analyzing the effectiveness of adversary modeling in security games. In *Conf. on Artificial Intelligence (AAAI)*, 2013.

[27] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus. Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS, pages 895–902, 2008.

[28] D. Pollard. *Convergence of stochastic processes*. David Pollard, 1984.

[29] R. Stanley. Eulerian partitions of a unit hypercube. *Higher Combinatorics (M. Aigner, ed.), Reidel, Dordrecht/Boston*, 49, 1977.

[30] M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.

[31] S. Tanny. A probabilistic interpretation of eulerian numbers. *Duke Mathematical Journal*, 40(4):717–722, 1973.

[32] V. Tikhomirov and A. Kolmogorov. ε-entropy and ε-capacity of sets in functional spaces. In *Selected Works of AN Kolmogorov*, pages 86–170. Springer, 1993.

[33] Y. Vorobeychik and B. Li. Optimal randomized classification in adversarial settings. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 485–492. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[34] X. Wang. Volumes of generalized unit balls. *Mathematics Magazine*, pages 390–395, 2005.

[35] R. Yang, B. Ford, M. Tambe, and A. Lemieux. Adaptive resource allocation for wildlife protection against illegal poachers. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, 2014.

[36] C. Zhang, A. Sinha, and M. Tambe. Keeping pace with criminals: Designing patrol allocation against adaptive opportunistic criminals. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, 2015.