Adversarial AI

Yevgeniy Vorobeychik
Electrical Engineering and Computer Science
Vanderbilt University
yevgeniy.vorobeychik@vanderbilt.edu

Abstract

In recent years AI research has had an increasing role in models and algorithms for security problems. Game theoretic models of security, and Stackelberg security games in particular, have received special attention, in part because these models and associated tools have seen actual deployment in homeland security and sustainability applications. Stackelberg security games have two prototypical features: 1) a collection of potential assets which require protection, and 2) a sequential structure, where a defender first allocates protection resources, and the attacker then responds with an optimal attack. I see the latter feature as the major conceptual breakthrough, allowing very broad application of the idea beyond physical security settings. In particular, I describe three research problems which on the surface look nothing like prototypical security games: adversarial machine learning, privacy-preserving data sharing, and vaccine design. I describe how the second conceptual aspect of security games offers a natural modeling paradigm for these. This, in turn, has two important benefits: first, it offers a new perspective on these problems, and second, facilitates fundamental algorithmic contributions for these domains.

1 Introduction

In recent years AI research has had an increasing role in models and algorithms for security problems. AI interest in adversarial problems is not new: there have been decades of research in adversarial AI settings, such as games (e.g., checkers, chess, and poker) and robust algorithms (such as robust learning and optimization). Unlike the classical adversarial AI, where research is primarily algorithmic, application of AI to security makes modeling a crucial part of the endeavor.

The prototypical model in security and AI is a two-player game between a *defender* and an *attacker*. The defender is charged with protecting some *assets* (commonly called *targets*), which the attacker wishes to attack. Both players typically have either resource constraints (e.g., a bound on the number of targets that can be protected/attacked), or incur a protection/attack cost, respectively. The utility functions then reflect the value of assets, when safely protected (the defender gains) or successfully attacked (the attacker gains). In my view, a significant conceptual milestone was to model this interaction between the defender and attacker as a *Stackelberg game*, in which the defender first chooses which assets to protect (in a possibly randomized fashion), and the attacker subsequently responds, *taking into account the defender's strategy* or, more generally, his posterior belief about it [Paruchuri *et al.*, 2008]. This modeling milestone had two significant ramifications: first, it facilitated deployment of AI solutions to security problems in the field, and second—and this is particularly salient for my research—it opened the door for transferring modeling concepts from security games to a broad range of domains.

In this companion paper, I briefly describe three example problems of broad interest in which I successfully applied security game modeling concepts.

2 Adversarial Machine Learning

The practical success of machine learning has led to its widespread integration within security applications, such as spam filtering and fraud and intrusion detection. Conceptually, machine learning works when sufficient training data is collected so that the resulting model makes accurate predictions on new data generated from the same, or sufficiently similar distribution. Focusing on classification, adversarial learning problems feature a benign and a malicious class (in the simplest case), with the goal of predicting whether a particular instance, characterized by a feature vector $x \in X \subseteq \mathbb{R}^d$, is malicious. Take spam email filtering as a natural example. We see an email, which we translate into a feature vector x (commonly, features are binary and indicate the presence of specific words in an email), and apply a classifier $h(x)$ to determine whether it's spam (say, labeled as $+1$) or not (labeled as -1). If we decide it's spam, the email is filtered (e.g., into a spam folder); otherwise it is delivered to the user.

Now, notice that a spam email, which we see through the feature space as x, actually represents an action of an individual—the *spammer*—who chose to send that particular email to, say, sell Viagra. If this email is subsequently filtered, the spammer can be expected to edit the email template, choosing a new email, x', that now bypasses the filter. If all spammers were to do this, classifier-based spam filters would rapidly become useless. In practice, such *drift* happens some-

what gradually, and spam filters are regularly "retrained" to keep up with the shifting spam landscape.

The scenario I just described is called an *evasion* attack. Clearly, evasion attacks can be devastating if the consequences of false negatives are high. This problem motivates the following two related research questions: 1) how do we model evasion attacks, and 2) how do we develop classification methods which are robust to evasion. Natural models of evasion attacks, including those typically proposed in the literature, fix the classifier, $h(x)$, an *ideal* instance, x, and consider an attacker who minimizes a distance (measured by some weighted l_p norm) between an attack instance x' and the ideal instance x, subject to the constraint that $h(x') = -1$ (that is, x' is classified as benign; passed on to the user in the spam filtering example) [Lowd and Meek, 2005]. As stated, this attack model isn't itself useful for robust learning (the attacker essentially always wins), but a natural variation is: impose a budget constraint so that the attacker will do nothing if the distance between x' and x is too large [Li and Vorobeychik, 2014]. Many variations on this theme are possible without changing the general idea. Whatever the specifics, consider a classifier parametrized by a weight vector w, and let $Q(w; x)$ be a function which returns an x', the response of the adversary with ideal instance x to classifier w.

With the framing of security games and this adversarial framework in mind, a Stackelberg security game is a good conceptual fit: the defender here is the learner, controlling w (the defender's action space), while the attacker solves the optimization problem (computes a best response to w), which we encapsulate in the function $Q(w; x)$ (the attacker's action space is X). The utility function of the defender (learner) is, typically, the empirical risk (as an approximation of expected loss), while the attacker's utility is a combination of distance to ideal instance x and either the value of being classified as benign, or some function of distance to the classification boundary. At this point, one may wonder how the learner would possibly know the attacker's ideal instance, x. The answer is, by looking at training data: ideal instances correspond to the malicious training instances.[1]

We proposed the first method for solving such games optimally for a l_1-regularized linear SVM with a very general class of adversarial models, including important models which cannot be captured by a simple norm (for example, allowing us to capture attackers who can substitute words, such as synonyms, without detracting from the purpose of the email) [Li and Vorobeychik, 2014]. The idea behind the algorithm was that if the feature space X is finite, one can formulate the learner's optimization problem as a mixed-integer linear program. This is impractical for a large feature space, so we leverage constraint generation to iteratively add attacks (x'), until the process converges. If the attacker is solving an optimization problem (to optimality), convergence implies an optimal solution for the learner. Our experimental evaluation showed that this approach is significantly more robust to evasion than alternatives (including several previous adversarial learning methods), with only a small sacrifice in accuracy if no evasion attacks occur. In a related effort, we proposed a principled approach for embedding randomization in decisions about whether or not to act on a feature vector x, where action can involve filtering or forensic analysis [Li and Vorobeychik, 2015]. The idea, again, was to model the evasion attacks as expected utility maximizers, with utility combining evasion and modification of an ideal instance, and the defender choosing a probability distribution over feature-conditional decision functions.

[1] One can think of this as a form of *revealed preferences*.

3 Privacy-Preserving Data Sharing

Now that data storage is relatively cheap, there is an increasing tendency to collect information, and share it with the analytics teams. Data sharing takes many forms: for example, NIH and NSF have explicit policies that promote the sharing of research data by requiring data management plans. Numerous data sharing platforms have emerged, for example, to further facilitate such activities as the sharing of genomic data for both validation and further exploration. Indeed, within clinical contexts there is commonly a separation between those who generate data (e.g., through clinical trials, or by managing the EMR systems) and those who analyze it. Data is also often shared to facilitate background checks (intellius.com), and to increase transparency of democratic institutions (e.g., property assessment data, voter registration data). Moreover, detailed online and economic activity data is now routinely collected by social media, credit card companies, search engines, and crediting bureaus, and is aggregated and subsequently resold by data brokers. Through these formal and informal transactions involving the sharing of data, privacy stands out as a major concern. If you are a healthcare provider, you are subject to explicit privacy regulations, in the form of HIPAA in the US, and the Data Protection Directive of the EU. Both of these regulations recognize the importance of sharing data, and offer guidelines to balance the value of shared data and privacy risk. Of particular salience to my research is that such guidelines allow the use of risk assessment to govern data sharing. HIPAA, for example, suggests as one of the criteria for data to be considered "safe" to share (from a regulatory perspective) that "*the risk is very small* that the information could be used, alone or in combination with other reasonably available information, by an *anticipated recipient* to identify an individual who is a subject of the information".

As an illustration, consider the following problem: we wish to share a record (corresponding to an individual), which we represent as a vector $x \in \mathbb{R}^n$, with x_i a particular field, such as name, SSN, address, ZIP code, gender, ICD9 (diagnostic) code, and so on. A typical goal is to transform x into another attribute vector y so that it is difficult to recover the identifying attributes (e.g., name) of x from y. Clearly, one would strip all direct (or nearly direct) identifiers, such as name, SSN, and exact address. The famous attack by Sweeney in which she re-identified an anonymous hospital record as belonging to a Massachusetts governor using a combination of indirect attributes (such as admission date and ZIP code) made clear that this isn't sufficient [Sweeney, 1997]. In particular, some of the information, such as demographics (for example, a combination of gender, race, sex, and ZIP code), can sometimes be used in conjunction with side

channel information to uniquely identify an individual in a record which is superficially anonymized [Sweeney, 2002]. A common source of such side channels are public resources such as voter registration records, which are fully identified, and often contain detailed address and demographic information. However, demographic information is often critical to release, since it is often correlated with medically relevant information, such as diagnosis and prognosis of a patient; removing it altogether therefore significantly restricts the scope of analytics. Indeed, such overprotection can at times lead to elevated health risk [Fredrikson *et al.*, 2014]. A common approach, therefore, is *generalization*, by which an attribute, such as a ZIP code, is replaced with a set containing it, such as the first 3 digits of the zipcode [Sweeney, 2002]. Indeed, one can consider a generalization hierarchy, by which a particular attribute can be generalized to increasingly coarse categories (or larger sets). In our notation, y_i can then represent the level of generalization of attribute i. There are two extreme cases of interest: first, when $y_i = x_i$, that is, the attribute is shared as is, and second, when y_i is set to the highest level in the generalization hierarchy, which is equivalent to not sharing this attribute at all (since y_i indicates that the value x_i belongs to a set of all possible values of attribute i).

The tradeoff between the value of shared data and the associated privacy risk can be naturally captured by defining a data sharing utility $v(y; x)$ associated with the shared attribute vector y, and the risk, $R(y)$. If these are normalized to be comparable, then the data sharer would aim to maximize $u(y; x) = v(y; x) - R(y)$. The question is: how do you determine risk, $R(y)$? In our work [Wan *et al.*, 2015], we view risk as determined by two factors: 1) the decision of the data recipient whether to attempt re-identification of y (i.e., recovering the individual's name), which we simply term *attack*, and 2) the likelihood of success, conditional on such an attack. Let $Q(y) \in \{0, 1\}$ encode the attacker's decision whether to attack the record ($Q(y) = 1$) or not ($Q(y) = 0$), and let $p(y)$ the conditional probability of success, given attack. Then $R(y) = LQ(y)p(y)$ for some constant loss L incurred if re-identification is successful. Notice here that the recipient's decision whether to attack is a function of y. This is a consequence of the fact that y is what is actually shared with the recipient, who doesn't need to decide anything before having the data in hand. If we now posit that the recipient's decision is one of optimizing net benefit of a re-identification attack, net loss (stemming from costs of an attack, as well as any penalties that may be imposed should the attack be discovered), modeling this setting using the concept of a Stackelberg security game is natural. The simplest way to formalize the recipient's decision within this framework is to have them maximize the utility function $Q(y) \in \arg\max_{a\in\{0,1\}} a(Vp(y) - c)$, where V and c are the attacker's benefit from a successful re-identification, and cost incurred, respectively. In this model, it's direct that $Q(y) = 1$ iff $p(y) > c/V$ (ignoring ties). There are several interesting consequences of this model. The first is that sometimes data is shared even if the publisher is certain that there is an attack; this would happen if either L or $p(y)$ are sufficiently small. The second is that this model can be directly augmented to include a hard constraint on risk $R(y)$; for example, we can actually constrain $R(y) = 0$, that is, that an economically motivated attacker, as specified above, will never attack (we call this the "no-attack" scenario). Remarkably, through a case study we found that both of these variants *outperform* what is perhaps the most common option offered by HIPAA, termed *Safe Harbor*, which amounts to an enumerated list of attribute generalization guidelines. Especially intriguing is the fact that the "no-attack", while obviously lower risk than Safe Harbor within our modeling framework, actually attains a higher data sharing utility on average: in a nutshell, Safe Harbor will both over- and under-protect records, depending on context.

4 Vaccine Design

Vaccination has a tremendous public health impact, preventing severe individual sickness, and in many cases epidemic outbreaks. However, a number of infectious diseases defy effective vaccination, influenza and HIV being, perhaps, the most prominent in that class. Vaccination works by eliciting an antibody which binds a viral protein, ideally at an active site (that is, a site which is critical for viral survival and replication). Design of successful vaccines entails a myriad of challenges. The one I am particularly interested in here is the evolutionary challenge involved in *antibody design*, or the design of an antibody that would be targeted by a vaccine: both flu and HIV can rapidly escape antibody binding through a series of mutations.

Both the antibody, and the viral target, are proteins. While binding is determined by the 3D structure of these proteins, such structure is itself determined entirely by the sequence of amino acids comprising the protein. The key challenge is that structure is extremely challenging to determine computationally from sequence alone. Nevertheless, numerous computational tools have emerged that compute protein structure structure, and determine binding properties of pairs of proteins (or pairs of molecules, more generally). These tools typically work by using some form of stochastic local search to minimize a total energy score over possible structures. For my purposes, let's consider binding prediction as a black box that operates on a pair of protein sequences, a and v, the former denoting an antibody sequence, while the latter the viral protein sequence (which in our case was the HIV gp120 envelope protein, a common target of antibody binding). Henceforth, I use a to informally denote both a particular antibody (including its structural characteristics), as well as formally to denote a sequence (vector) of amino acids.

A basic antibody design problem can be formulated as the following feasibility problem: find an antibody a such that $B(a, v) \leq \theta$, where v is a *specific* target viral protein, $B(a, v)$ is the binding energy, and θ is a binding threshold (i.e., energy below this threshold implies that a binds to v). Commonly, this is posed as simply minimizing binding energy: $\min_a B(a, v)$ by the use of local search. Now, suppose that we have identified an antibody a which binds to the target virus v, is that the end of the line? I argue that it is not: what is missing is the potential of viral mutation to escape binding to the antibody.[2] In particular, let $v' = Q(a; v)$ be a muta-

[2]The way this is often captured is by considering a panel of com-

tion to the *native* virus protein v which escapes binding a. Mutations are generally stochastic, but there are a number of effective constraints on these, including: a) survival (if too many mutations are required, the virus is unlikely to survive long enough to escape), and b) fitness (native type must have a fitness advantage). To capture both of these, we introduced a simple model of viral response based on the following optimization problem: minimize the number of mutations from the native protein to escape the antibody a. We can formalize this as the following optimization problem for the virus: $\min_{v':B(a,v')\leq\theta} \|v-v'\|_0$, where the l_0 norm counts the number of mutations from the native virus v, and the constraint $B(a,v') \leq \theta$ ensures that v' has escaped binding.

A pivotal feature of the virus escape problem is that it depends on the antibody a which is being escaped. What implication does this have for antibody design? At this point, it seems that the conceptual framework from Stackelberg security games is a natural fit: the defender's goal is to design an antibody sequence a which is difficult for virus to escape. We can formalize this by capturing said difficulty as the number of mutations required to escape, which gives rise to the following optimization problem for the antibody [Panda and Vorobeychik, 2015]: $\max_a \min_{v':B(a,v')\leq\theta} \|v - v'\|_0$. Formalizing the problem this way enables, in principle, the application of AI algorithms to this particular combinatorial optimization problem. Of course, this is still not trivial: both the antibody design space, and viral escape space, are enormous. Just to be concrete, a typical antibody/virus protein is on the order of 100 amino acids long (at least). Since there are 20 amino acids that can be in each protein position, the joint search space is 200^{20} (which does not even include the possibility of insertions/deletions at particular positions). Even if we focus solely on the binding sites (which requires us to identify a baseline 3D binding structure, such as the VRC01/gp120 complex [Li *et al.*, 2011], which is what we used in our work), the resulting space is around 100^{20}: substantial savings, to be sure, but hardly a tractable search space. Our approach was to make use of stochastic local search, relying on single-point mutations for the virus which maximally increase binding energy, and considering local random deviations from a known broadly binding antibody (VRC01). To improve scalability further, we made us of machine learning techniques to predict binding. While such techniques are not typically very accurate, they allow us to rapidly scan the search space for promising candidates, and we only used the computationally expensive protein modeling tools (in our case, Rosetta) in the relatively few promising cases. After putting the computational elements together, we found antibodies which were significantly more difficult for the viral protein to escape through mutations than the VRC01 antibody which we used as the baseline (and which is one of the most broadly binding antibodies known).

mon viral subtypes, and seeking an antibody which binds to many of these. This task is generally viewed as quite computationally challenging, and only considers a fixed panel of viruses, ignoring rare variants that may become common as a result of vaccination.

5 Conclusion

Adversarial machine learning, privacy-preserving data sharing, and vaccine design, are three problems that, on the surface, seem entirely distinct. What I hope to have demonstrated is that these problems nevertheless can be usefully captured within a single high-level modeling framework of Stackelberg security games. This conceptual framework, thus, is surprisingly general, potentially enabling us to port AI algorithmic approaches designed for this class of models to a broad variety of problems of considerable social relevance.

6 Acknowledgments

The author gratefully acknowledges support from NSF (IIS-1526860), ONR (N00014- 15-1-2621), ARO (W911NF-16-1- 0069), NIH (R01- HG006844), and Sandia National Labs.

References

[Fredrikson *et al.*, 2014] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *USENIX Security Symposium*, pages 17–32, 2014.

[Li and Vorobeychik, 2014] Bo Li and Yevgeniy Vorobeychik. Feature cross-substitution in adversarial classification. In *Neural Information Processing Systems*, pages 2087–2095, 2014.

[Li and Vorobeychik, 2015] Bo Li and Yevgeniy Vorobeychik. Scalable optimization of randomized operational decisions in adversarial classification settings. In *Conference on Artificial Intelligence and Statistics*, 2015.

[Li *et al.*, 2011] Y. Li, S. O'Dell, L.M. Walker, X. Wu, J. Guenaga, Y. Feng, S.D. Schmidt, K. McKee, M.K. Louder, J.E. Ledgerwood, B.S. Graham, B.F. Haynes, D.R. Burton, R.T. Wyatt, and J.R. Mascola. Mechanism of neutralization by the broadly neutralizing HIV-1 monoclonal antibody VRC01. *Journal of Virology*, 85(17):8954–8967, 2011.

[Lowd and Meek, 2005] Daniel Lowd and Christopher Meek. Adversarial learning. In *International Conference on Knowledge Discovery in Data Mining*, pages 641–647. ACM, 2005.

[Panda and Vorobeychik, 2015] Swetasudha Panda and Yevgeniy Vorobeychik. Stackelberg games for vaccine design. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 1391–1399, 2015.

[Paruchuri *et al.*, 2008] Praveen Paruchuri, Jonathan P. Pearce, Janusz Marecki, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 895–902, 2008.

[Sweeney, 1997] L. Sweeney. Weaving technology and policy together to maintain confidentiality. *J Law Med Ethics*, 25(2-3):98–110, 1997.

[Sweeney, 2002] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *Int J Uncertainty Fuzz*, 10(5):571–588, 2002.

[Wan *et al.*, 2015] Zhiyu Wan, Yevgeniy Vorobeychik, Weiyi Xia, Ellen Wright Clayton, Murat Kantarcioglu, Ranjit Ganta, Raymond Heatherly, and Bradley A. Malin. A game theoretic framework for analyzing re-identification risk. *PLoS ONE*, 10(3):e0120592, 2015.