Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science

Semestral Project

Adversarial Machine Learning for Detecting Malicious Behavior in Network Security

Bc. Michal Najman

Supervised by
Mgr. Viliam Lisý, MSc., Ph.D.

Submitted in February, 2019

Contents

1 Monte Carlo Descent

(The feature map Θ is omitted for clarity in all equations.)

Assume D_θ is a distribution over $\mathcal{H}$ parametrised by $\theta \in \Theta$. The defender risk is then given by:

$$R(D_\theta) = \mathop{\mathbb{E}}_{h \sim D_\theta} R(h)$$

where $R(h)$ is the expected risk of the defender deploying a classifier h.

$$R(h) = \mathop{\mathbb{E}}_{(\pi,c)} \ell(h(\pi), c) = \mathop{\mathbb{E}}_{\pi}[\ell_h^{\mathsf{B}}(\pi)|\mathsf{B}]p(\mathsf{B}) + \mathop{\mathbb{E}}_{\pi}[\ell_h^{\mathsf{M}}(\pi)|\mathsf{M}]p(\mathsf{M})$$

Since the attacker's obfuscation function $g(Q^{\mathsf{cr}})$ maps a critical query set to a single obfuscated query profile (assuming Strong Stack. Eq.) and we defined $p(\pi|\mathsf{M}) = \sum_{Q^{\mathsf{cr}}:g^*(Q^{\mathsf{cr}})=\pi} p(Q^{\mathsf{cr}})$, the risk generated by class M is given by:

$$\mathop{\mathbb{E}}_{\pi}[\ell_h^{\mathsf{M}}(\pi)|\mathsf{M}] = \mathop{\mathbb{E}}_{Q^{\mathsf{cr}}}[\ell_h^{\mathsf{M}}(g^*(Q^{\mathsf{cr}}))|\mathsf{M}]$$

where g^* is the optimal obfuscation function, i.e. the defender's best response to h.

The nature of the task implies the priori class probabilities are not known. Therefore, we choose to redefine the task to comply with the Neyman-Pearson Problem. This means, instead of minimising $R(D_\theta)$, we solve the following problem:

$$\min_{\theta} \mathop{\mathbb{E}}_{Q^{\mathsf{cr}}}[\ell_h^{\mathsf{M}}(g^*(Q^{\mathsf{cr}}))|\mathsf{M}] \qquad \text{s.t.} \quad \mathop{\mathbb{E}}_{\pi}[\ell_h^{\mathsf{B}}(\pi)|\mathsf{B}] \leq \tau_{max}$$

Employing Langrange multiplicators, we arrive at:

$$\min_{\lambda,\theta} \mathop{\mathbb{E}}_{Q^{\mathsf{cr}}}[\ell_h^{\mathsf{M}}(g^*(Q^{\mathsf{cr}}))|\mathsf{M}] + \lambda(\mathop{\mathbb{E}}_{\pi}[\ell_h^{\mathsf{B}}(\pi)|\mathsf{B}] - \tau_{max})$$

Notice that $\lambda = \frac{p(\mathsf{B})}{p(\mathsf{M})}$. This is useful in choosing a set of possible lambdas prior to optimisation.

1.1 Attacker

As already shown, the attacker optimisation task reduces to:

$$g^*(Q^{\mathsf{cr}}) \in \min_{\pi \in S(Q^{\mathsf{cr}})} \Theta_A(\pi, Q^{\mathsf{cr}}) \qquad \text{s.t.} \quad h(\pi) = \mathsf{B}$$

1.2 Inspiration by Stackgrad

Building on the basis of Amin et al. (STACKGRAD), we first fix λ and compute correspondingly the prior class probabilities. This reduces the task back to the general definition with now estimated prior class probabilities. The problem can now be solved with gradient descent. The gradient of the criterion is given by:

$$\nabla_\theta R_\lambda(_\theta) = \mathop{\mathbb{E}}_{h,c,\pi}[\ \ell(h(\pi),c)\cdot\frac{\nabla_\theta p(h|\theta)}{p(h|\theta)}|\theta]$$

Notice the gradient is defined as an expectation over random variables h, c, π. Since its full analytical form is intractable, we can estimate it by sampling triple. First we sample h according to $p(h|\theta)$ and c according to $p(c)$. If a benign class is sampled, then $\pi$is taken from the training dataset of benign samples T^{B}. But if a malicious class is sampled, we randomly select Q^{cr} and map it to $\pi = g^*(Q^{\mathsf{cr}})$, i.e. we perform the defender's optimisation. The gradient can be estimated from m sampling cycles based on the current $\theta^(t)$, each yielding $\gamma^(i)$:

$$\gamma^{(i)} = \ell(h^{(i)}(\pi^{(i)}), c^{(i)})\cdot\frac{\nabla_\theta p(h^{(i)}|\theta^{(t)})}{p(h^{(i)}|\theta^{(t)})}$$

For now, it is unclear how to cope with an unbounded, continuous family of classifiers $\mathcal{H}$ as in their paper, Amin et al. work with discrete set $\mathcal{H}$. Thus, $\theta^(t)$ can be initialised for example so that $p(h|\theta)$ forms a uniform distribution and h is sampled as one of the members of $\mathcal{H}$. However, in our case, $\mathcal{H}$ is a family of neural networks (or perhaps linear classifiers). This means we cannot model $p(h^(i)|\theta^(t))$ easily.

Lanctot et al., however start with an initial $\mathcal{H}$ and increase it by adding iteratively a newly found h as a best response to the current strategy of other players (they assume Nash Eq.). This might be reformulated (hopefully) to the Stackelberg Game setting.