Near-optimal Evasion of Randomized Convex-inducing Classifier in Adversarial Environments

PRESENTER: POORIA MADANI (@POORIA_MDN)
FACILITATOR: TAHSEEN SHABAB

MAY 23RD , 2019

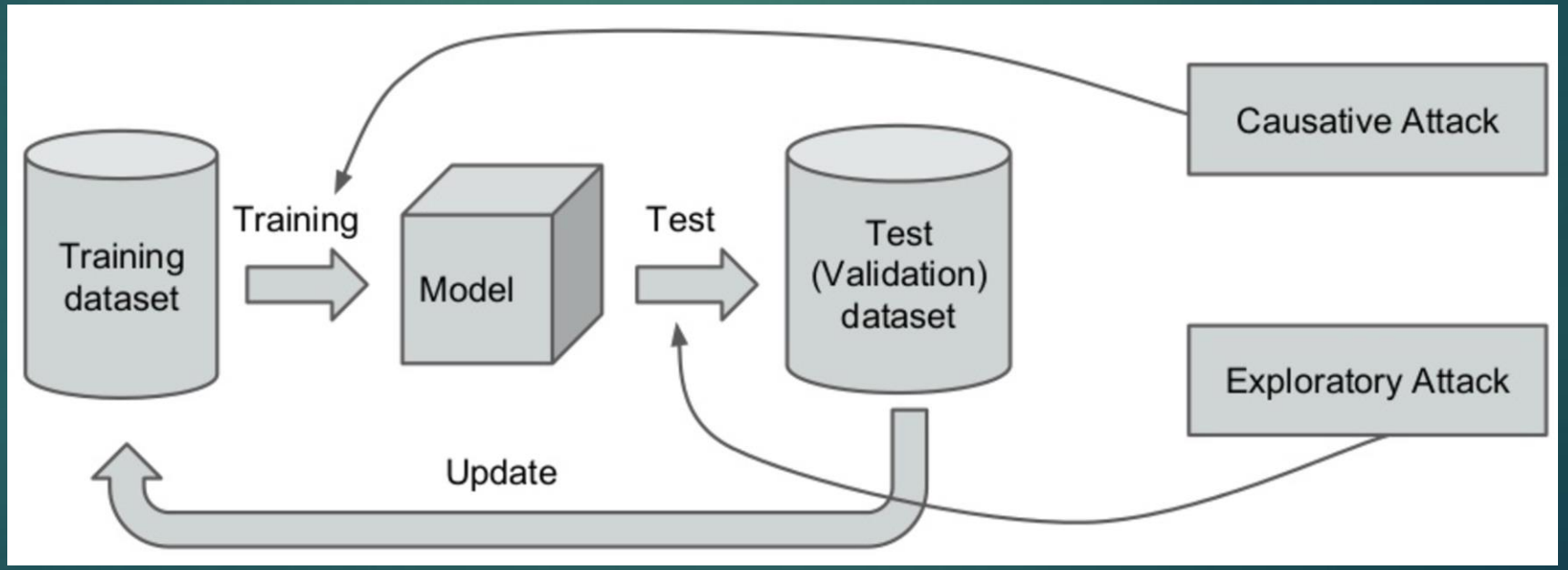
Cybersecurity is a game!

Outline

- Threat Models
- Define Adversarial Classifier Reverse Engineering (ACRE)
  - What is an optimal evasion?
- Optimal Evasion of Convex Inducing Classifiers
  - Complexity Analysis
  - Moving Target Defense (MTD)
- Active Line Search
- Simulations and Results
- Discussions

Motivation: Threat Models

- Adversaries are smart and motivated individuals – they break all the assumptions you have made about the data.



Source: [*Xiao Huang,* 2015]

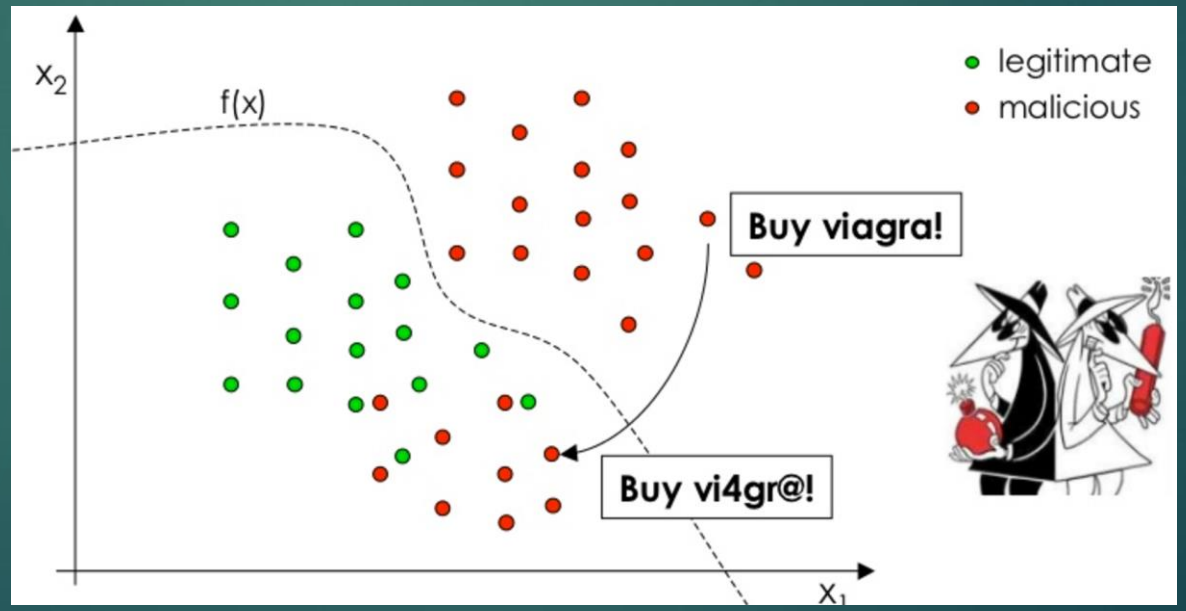
Threat Models – cont'd

	Integrity	**Availability**	**Confidentiality**
Run-time	Evasion	-	Model Extraction (e.g., hill climbing attack)
Training-time	Poisoning (i.e., implementing backdoors)	Poisoning (i.e., maximizing classification error)	-

Threat Models – cont'd

- Optimal Evasion

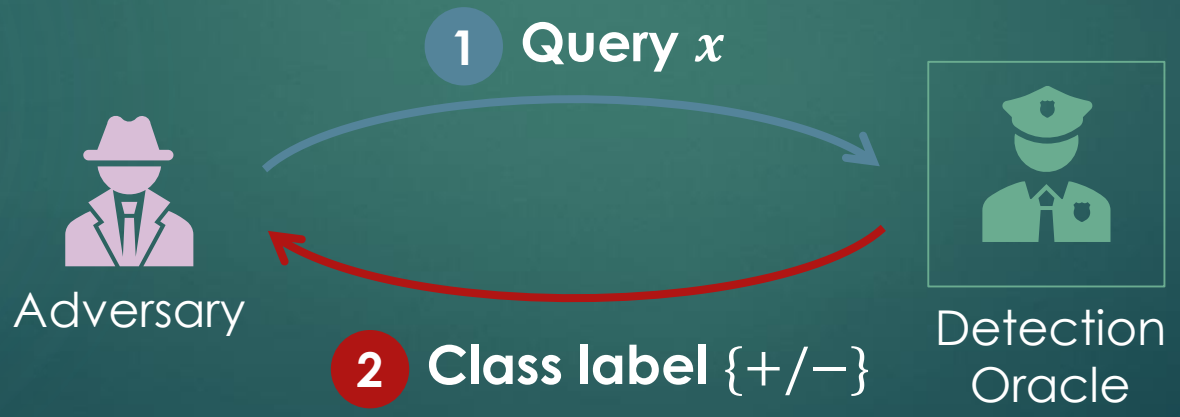
  evade detection with minimum number of change made to an attack instance/vector.



Source: [Biggio et. al. , ACML 2011]

Threat Models – cont'd

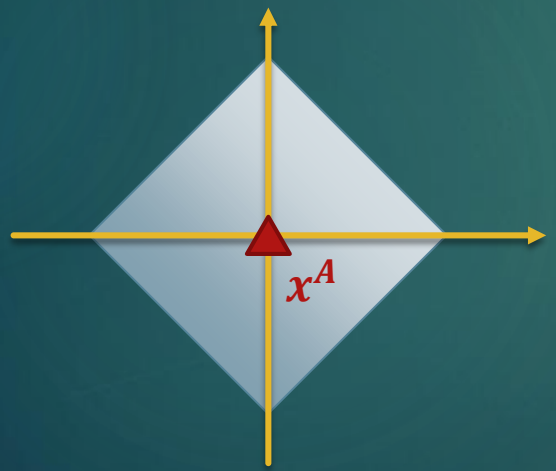
- Problem Setup
  - Adversary has a detectable malicious instance x^A
  - Adversary is aware of one instance x^- that belongs to benign class
  - Adversary can construct an arbitrary x and query the detection system to find out its label (e.g., +/-)



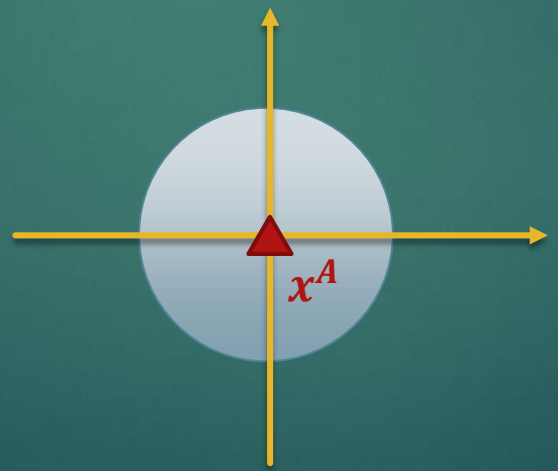
Minimal Adversarial Cost (MAC)

- Adversarial Cost Function

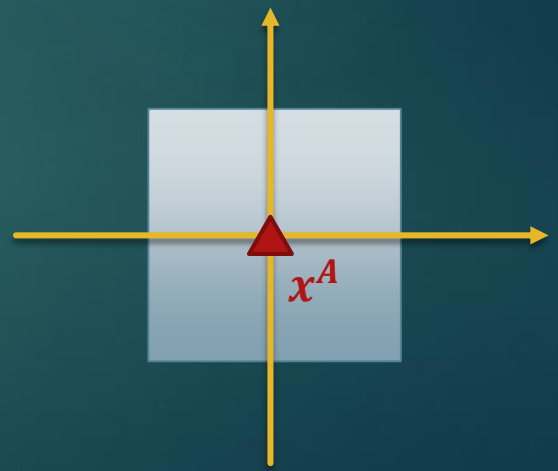
$$A_p^c\left(x - x^A\right) = \left(\sum_{d=1}^{D} c_d \left|x_d - x_d^A\right|^p\right)^{1/p}$$



$p = 1$



$p = 2$



$p = \infty$

Minimal Adversarial Cost (MAC) – cont'd

- Adversarial Cost Function

Initial Malicious Instance

$$A_p^c\left(x - x^A\right) = \left(\sum_{d=1}^{D} c_d \left|x_d - x_d^A\right|^p\right)^{1/p}$$

- Then, for a given classifier f

$$MAC(f, A) \triangleq \inf_{x \in X_f^-} [A(x - x^A)]$$

Benign Class

- Instance of minimal adversarial cost

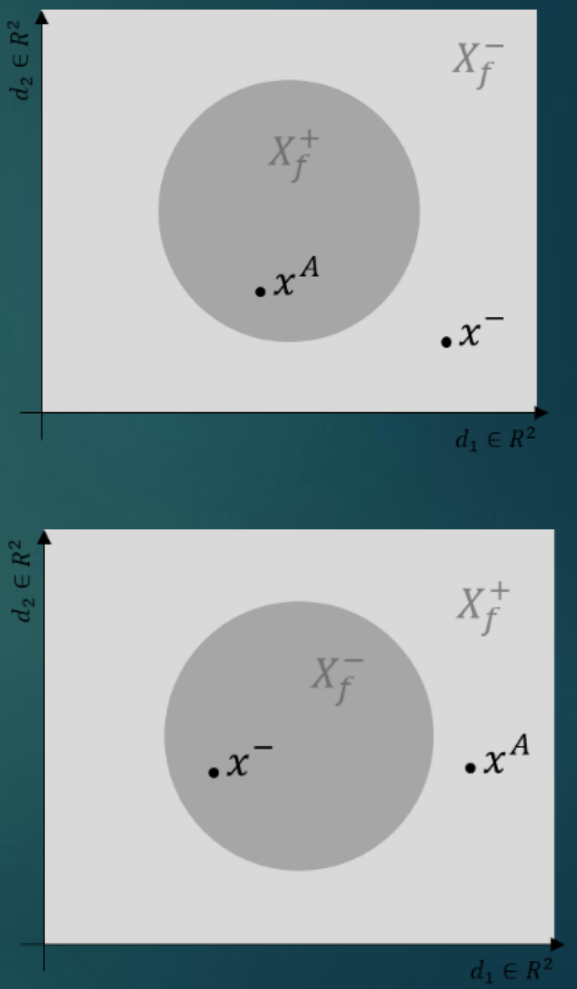
$$\epsilon - IMAC(f, A) \triangleq \left\{x \in X_f^- \mid A(x - x^A) \leq (1 + \epsilon) \cdot MAC(f, A)\right\}$$

- Near Optimal Evasion (ACRE searchable)

A family classifiers F is $\epsilon - IMAC$ searchable under a family cost function if for all $f \in F$ and some A , there is an algorithm that finds $x \in \epsilon - IMAC$ using polynomial-many membership queries. [Nelson *et al.* 2010]

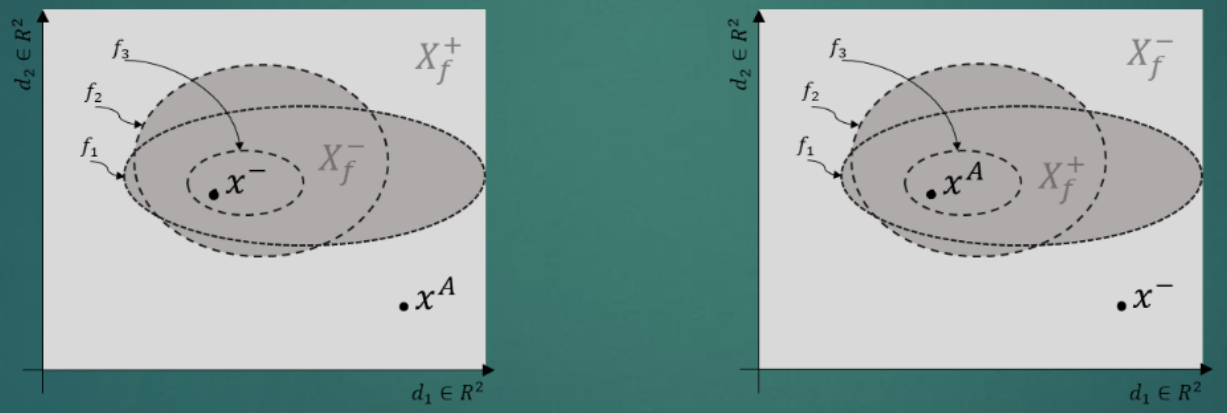
ACRE Searchability of Convex Bodies

- Convex-family classifiers forms an important collection of algorithms used in many adversarial environments.
  - *linear classifiers, hyper-sphere boundary anomaly detectors, one-class anomaly detectors*, and …
- For ℓ_1 (i.e., p=1) adversarial cost function, the family of convex inducing classifiers are ACRE searchable.
- For $\ell_{>1}$ the problem is NP-Complete.



Moving Target Defense

- It is hypothesized, that use of multiple classifiers at runtime and randomly selecting among them at runtime could make ACRE searchability harder.



- Construct multiple models, select one at random to handle a given query.

Reformulation of ACRE Searchability

- Given the randomized strategy of the defender:

  Find an instance of ϵ-IMAC that evade detection with probability better than P.

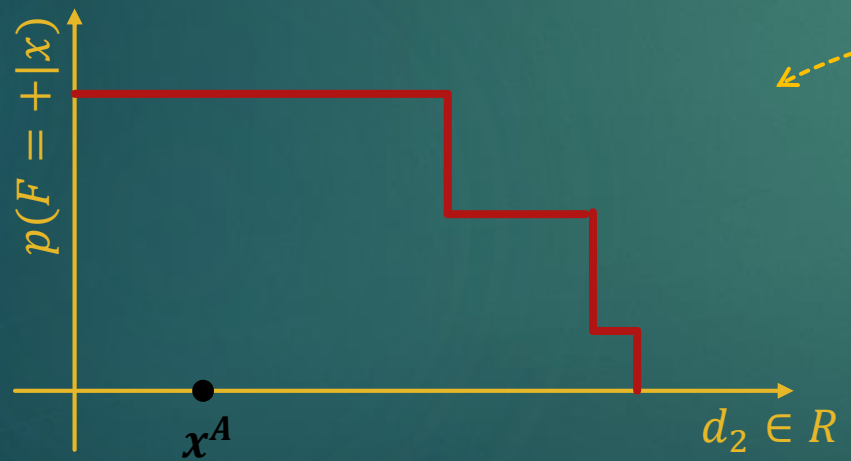
- The goal of adversary is construct

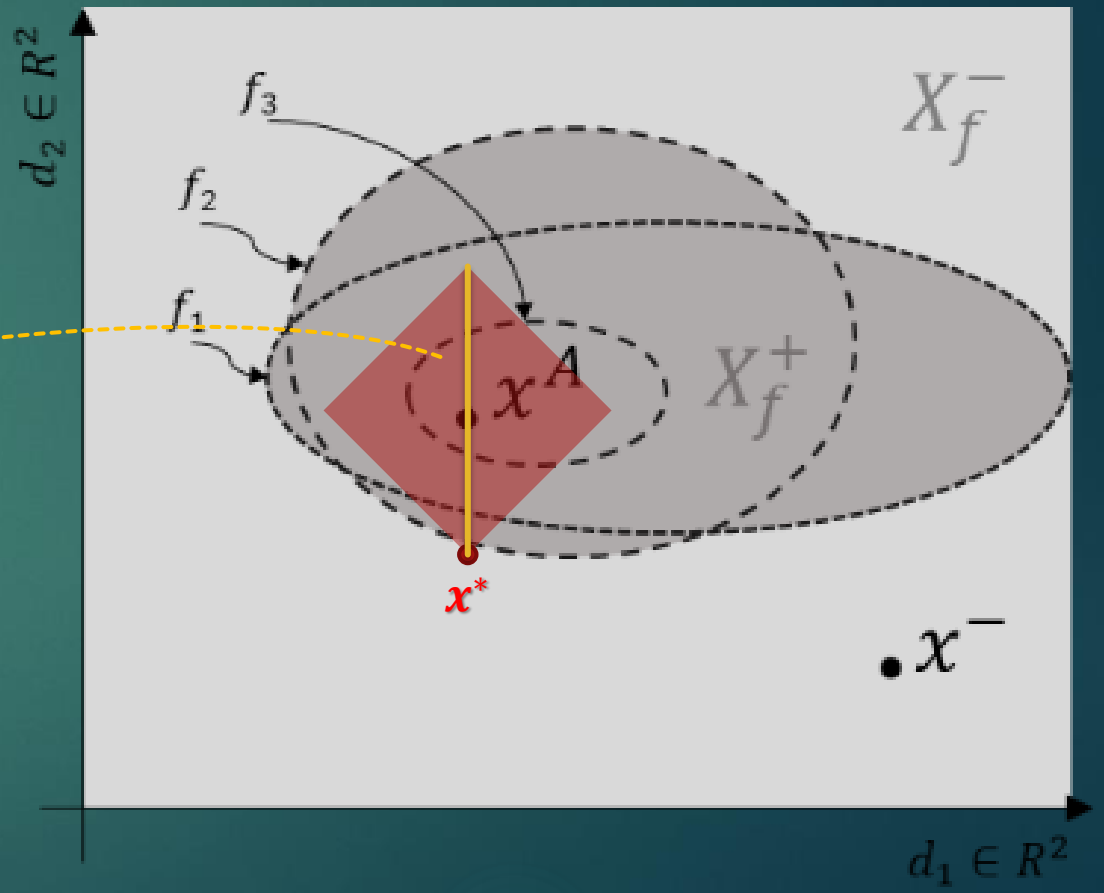
$$\inf_{\mathbf{x}^* \in \boldsymbol{X}_f^-} \boldsymbol{A}_1^{\boldsymbol{c}} \left(\boldsymbol{x}^* - \boldsymbol{x}^{\boldsymbol{A}} \right)$$

w.r.t. $p(f = +|x^*) \leq P$

Active Line Search

- For ℓ_1 cost function and a malicious (i.e., X_f^+) convex family





Active Line Search – cont'd

- $p(F = +|x, d)$ is representing the probability of success of a Bernoulli random variable at point x.

- Use Gaussian Process Regression (GPR) to learn $p(F = +|x, d)$ for different x along each feature dimension.
  - Learn the probability of success (θ) through sampling and submitting membership queries to the defender's black box.

$$\begin{bmatrix} \theta \\ \theta^* \end{bmatrix} \sim GPR(\mu, \Sigma) \qquad \Sigma = \begin{pmatrix} \kappa + \sigma(\theta)I & k_* \\ k_*^T & k_{**} \end{pmatrix} \qquad \kappa(q_i, q_j) = \exp\left(-\frac{1}{2\iota^2}(q_i - q_j)^2\right)$$

  - Interpolate θ^* for other x that have not been observed.

$$\theta^* = k_*^T(\kappa + \sigma I)^{-1}\theta \qquad Var[\theta^*] = k_* - k_*^T(\kappa + \sigma I)^{-1}k_*$$

Active Line Search – cont'd

Algorithm 1: Algorithm to find intializer.

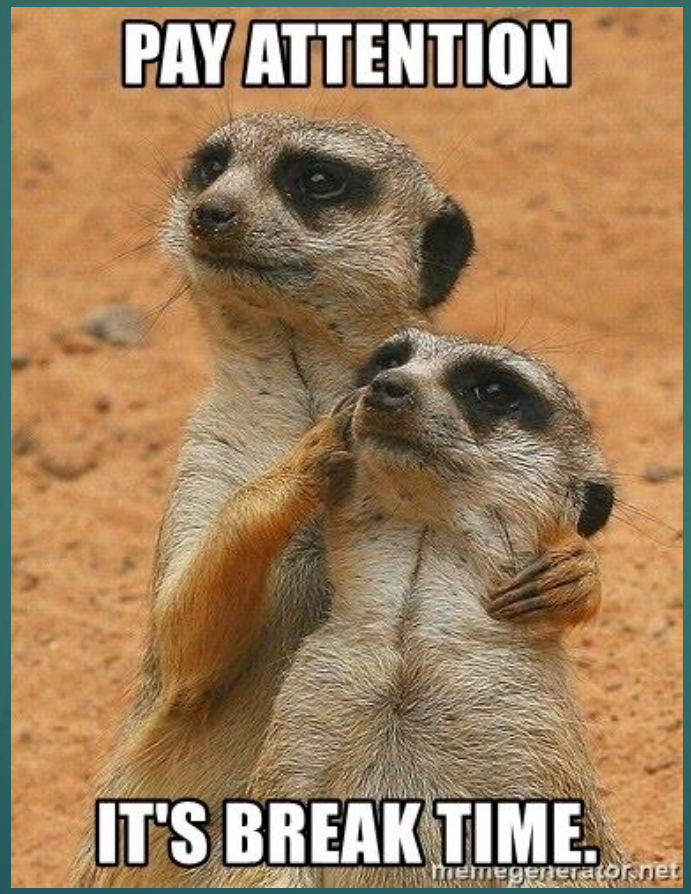
Data: x^A, x_-, C,ε,T_σ,P
Result: x^* $p\varepsilon$-IMAC instance

$\vec{\theta} \leftarrow [1, 0]$
$\vec{X} \leftarrow [x^A, x^-]$
$x^* \leftarrow x^-$
do
  construct GPR (MEAN($\vec{\theta}$), KERNEL($\vec{X}$))
  foreach $d_i \in D$ **do**
    foreach $\{x \mid \mathbf{x}^A \pm C_d \cdot \delta_d$ *that are* ε *spaced*$\}$ **do**
      $\theta_i, var_i \leftarrow \hat{f}(x^i)$; // GPR prediction at x^i
      if var_i *is maximum* **then**
        $max_i \leftarrow x^i$
      end
      if $\theta_i < P$ **and** $var_i \approx 0$ **and** $|x_d^A - x_d^i| \leq C_d$ **then**
        $C_d \leftarrow |x_d^A - x_d^i|$
        if $||x^A - x^i||_1 \leq ||x^A - x^*||_1$ **then**
          $x^* \leftarrow x^i$
        end
      end
    end
  end
  query the oracle and find θ_i and $\sigma(\theta_i)$ for max_i
  $\vec{X} \leftarrow \vec{X} \bigcup max_i$
  $\vec{\theta} \leftarrow \vec{\theta} \bigcup \theta_i$
  update KERNEL and noise $\sigma(\theta_i)$ based on Equation 4
while *maximum variance is above* T_σ;
return x^*

- Begin by two known points to the adversary: x^A and x^- and their corresponding $\theta = 1$ and $\theta = 0$ and train a GRP.
- Then, get the trained GRP to interpolate all (ϵ separated) data points along each feature vector, and select the one with highest variance (x_i).
- Find an estimate for θ_i of the selected point by multiple membership queries.
- Update the covariance matrix with newly acquired information and uncertainty.
- Retrain the GRP

$$Var(\theta_i) = \sqrt{\frac{\theta_i(1-\theta_i)}{N}}$$

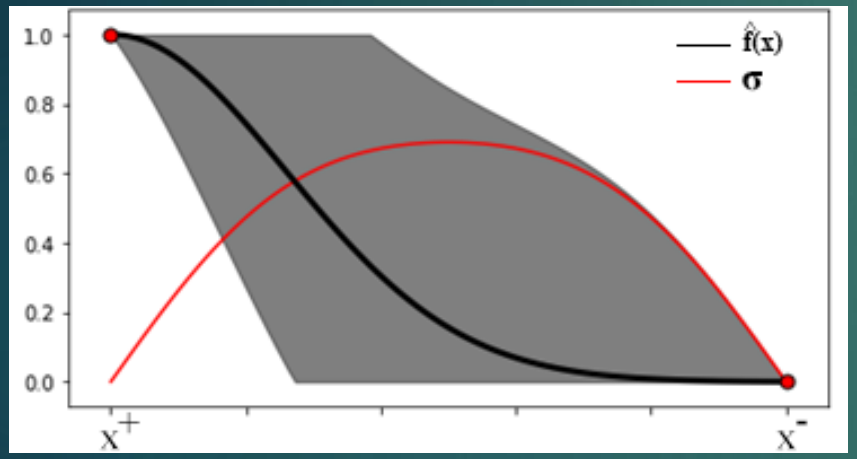
BREAK



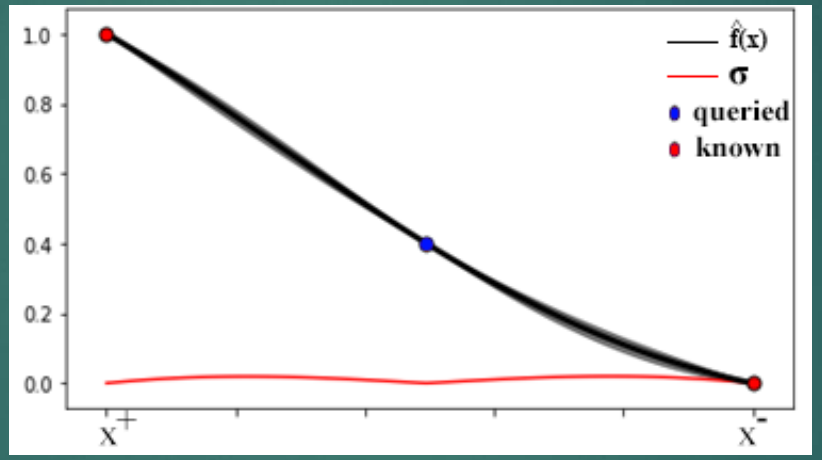
Toy Example (1)

- Example of 1-D randomized decision points

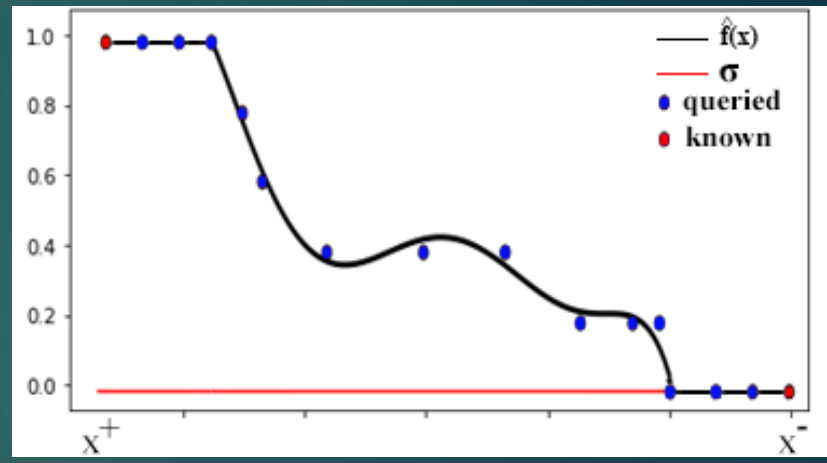
Initialization



Iteration 1

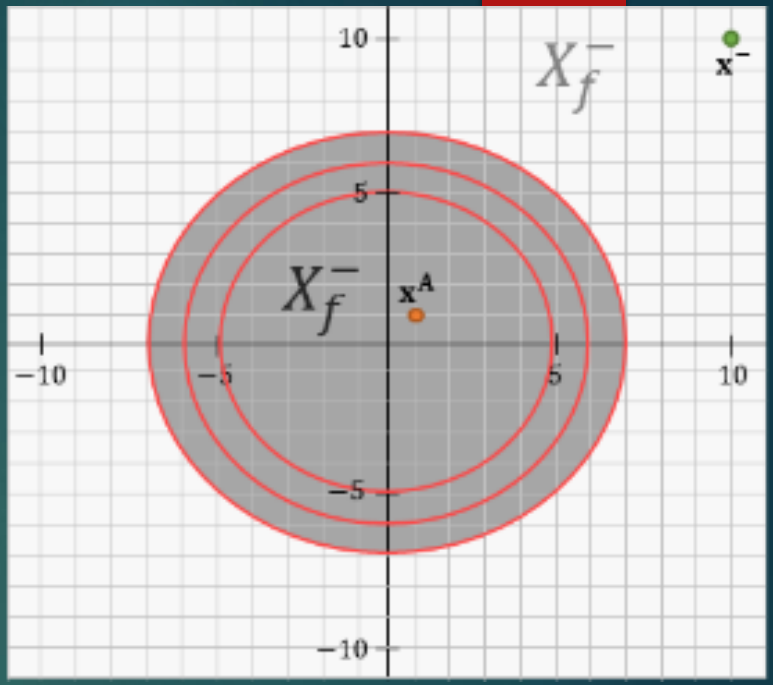


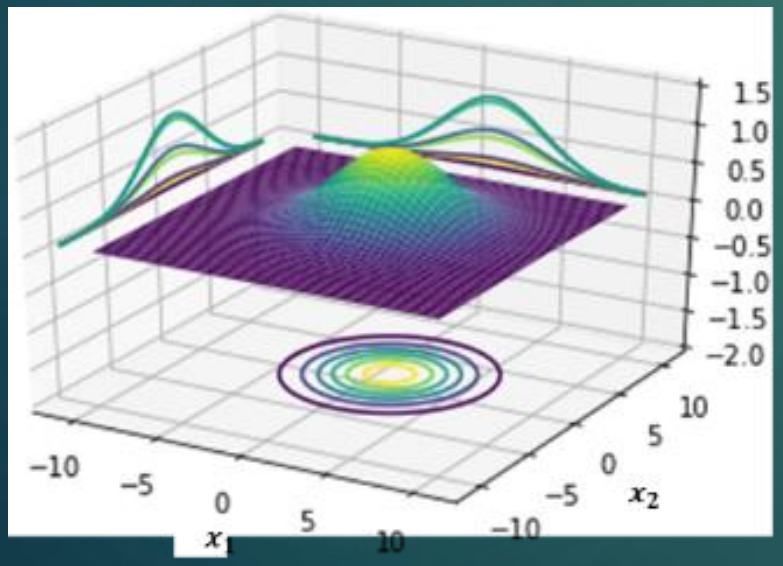
Last Iteration



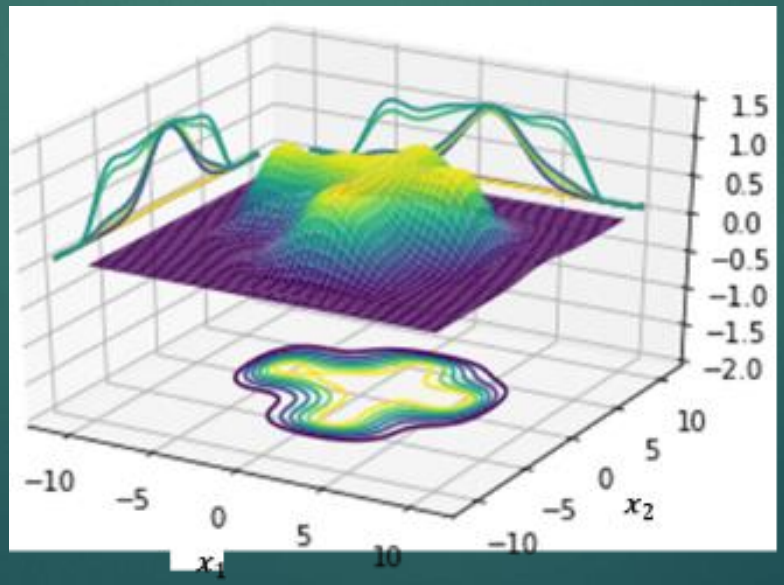
Toy Example (2)

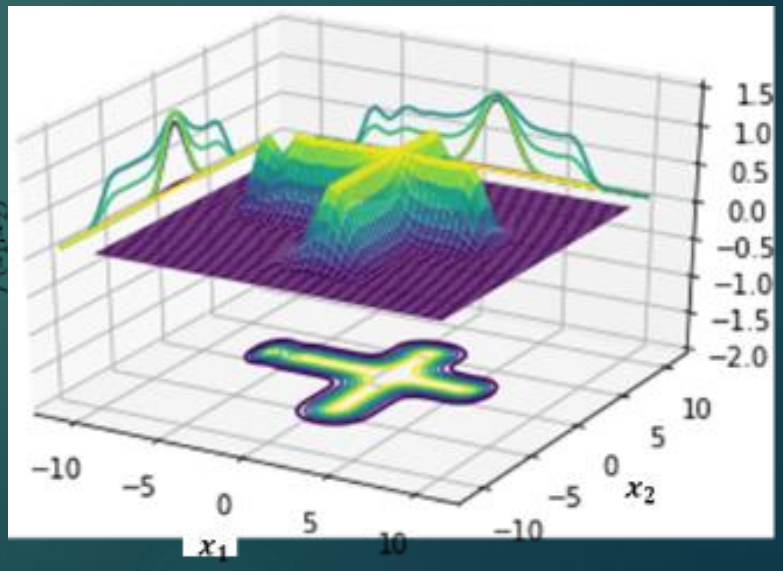
- Example of 2-D randomized decision hyper-spheres





Initialization





Final Iteration

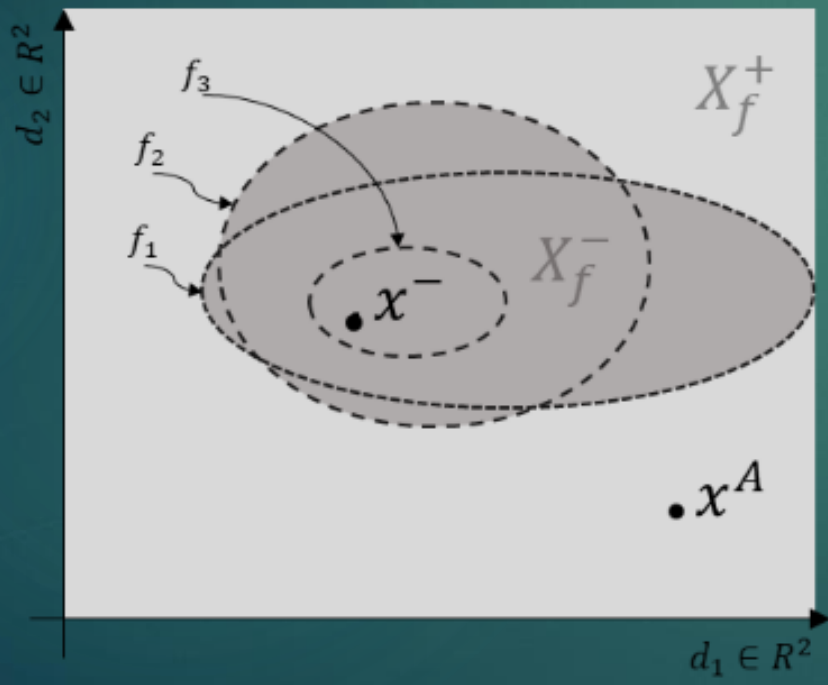
Toy Examples' Analysis

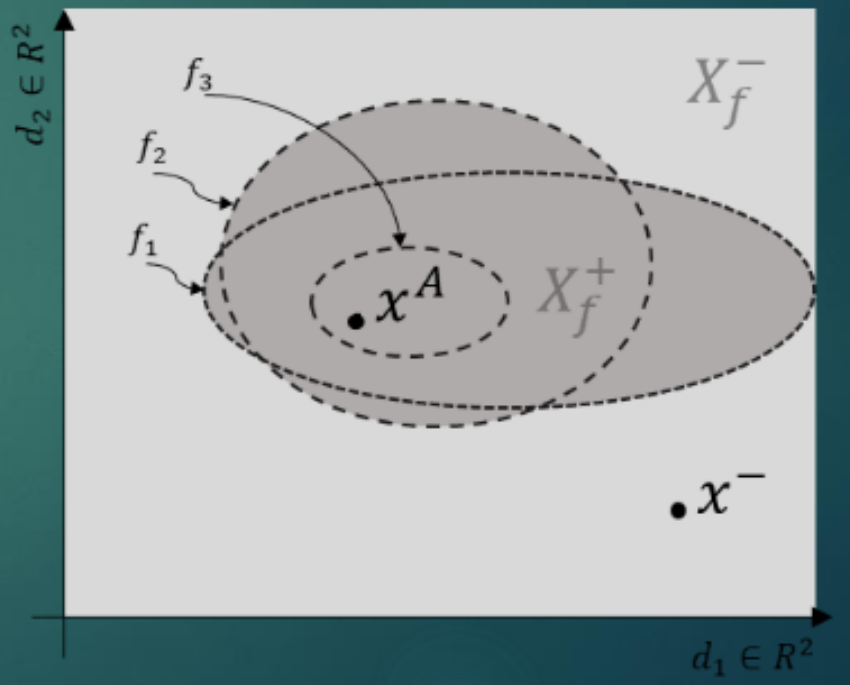
- Randomization certainly have increased the number of required query.
- Ignoring randomization can result in non-optimal result.
- Our proposed active line search algorithm converges with time complexity of $O(d \cdot \frac{|x^A - x^-|}{\epsilon})$.

Method	To Example	P	#Queries	Avg Error relative to MAC
Nelson et al.	1-D	1.0	**34**	−0.6183020
		0.5	N/A	N/A
	2-D	1.0	**57**	−1.5764189
		0.5	N/A	N/A
Active Multiline Search	1-D	1.0	122	**0.0022201**
		0.5	122	**0.0175023**
	2-D	1.0	254	**0.5510204**
		0.5	254	**0.5306122**

What about when X_f^- is convex?

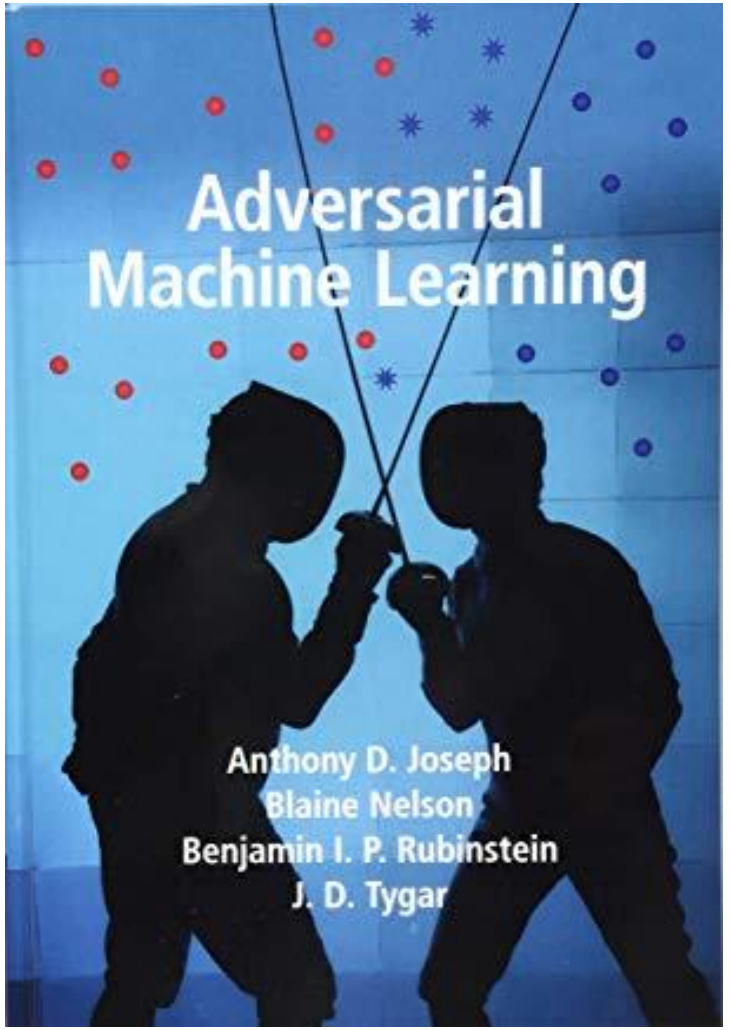
- We believe such case is not ACRE searchable!
  Can you help us prove it?





Closing Remarks

- Randomization, certainly, would create more trouble for an adversary to successfully carry-out an ACRE search.
- The randomization, when X_f^+ is convex, can still be evaded using polynomial-many queries.
- How would randomization affect ACRE search when X_f^-?
- How would feature-space transformation (with or without) randomization affect ACRE search?



Discussion Points

- Does randomization have an opportunity cost?
- I can't help but think about the early misconceptions of encryption where randomized hashing is/was often termed as more secure by novice practitioners which is not as per best practice. Is this true for randomization in this project?
- Would you happen to know if there are controls in place today to detect such abnormal sequence of inputs from potentials hackers?
- Would it be possible to elaborate on why one is a convex problem and not the other?

Discussion Points – cont'd

- Have you experimented with the constant which bounds the binomial sampling?
  - How do we pick the optimal constant?
- Do you think the sampling performed by Hackers can trigger existing anomaly detection tools used in production environments, thereby preventing the attack altogether.
  - Ex. If we were to use the above approach on GMAIL, do you think google could possibly model the sampling (ex. Sending the same email with small deviations in an iterative manner) as an anomaly? STOP THE PROBLEM AT THE SOURCE
- In our data-driven world, do you think it makes sense for SOC Analysts to have a visualiztion of decision boundaries handy for them to audit?
  - Rule-based detection is audited by experts for IPS/IDS.
  - What do you think is a best practice for us going forward?

Discussion Points – cont'd

- Would you have any intuition on explainability of models leveraging the MAB model you depicted?