Learning with strategic agents: from adversarial learning to game-theoretic statistics

Patrick Loiseau, EURECOM (Sophia-Antipolis)

Graduate Summer School: Games and Contracts for Cyber-Physical Security

IPAM, UCLA, July 2015

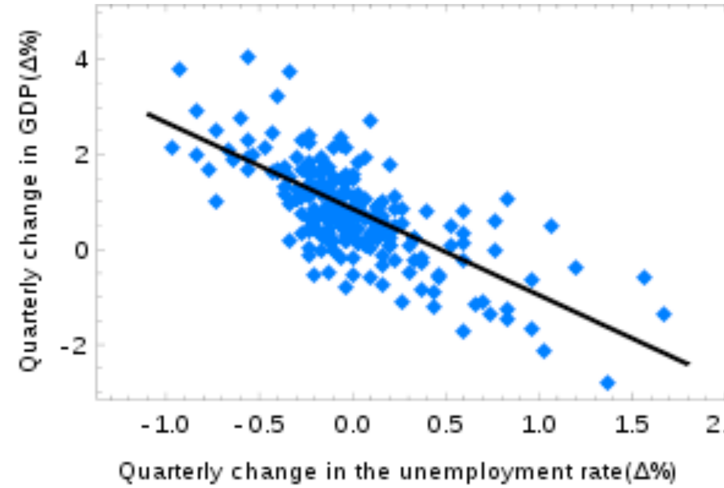
Supervised machine learning

Cats

vs

Dogs

Cat or dog?



- Supervised learning has many applications
  - Computer vision, medicine, economics
- Numerous successful algorithms
  - GLS, logistic regression, SVM, Naïve Bayes, etc.

Learning from data generated by strategic agents

- Standard machine learning algorithms are based on the "iid assumption"

- The iid assumption fails in some contexts
  - Security: data is generated by an adversary
    - Spam detection, detection of malicious behavior in online systems, malware detection, fraud detection
  - Privacy: data is strategically obfuscated by users
    - Learning from online users personal data, recommendation, reviews

→ where data is generated/provided by strategic agents in reaction to the learning algorithm

→ How to learn in these situations?

Content

Main objective: illustrate what game theory brings to the question "how to learn?" on the example of:

Classification from strategic data

1. Problem formulation
2. The adversarial learning approach
3. The game-theoretic approach
   a. Intrusion detection games
   b. Classification games

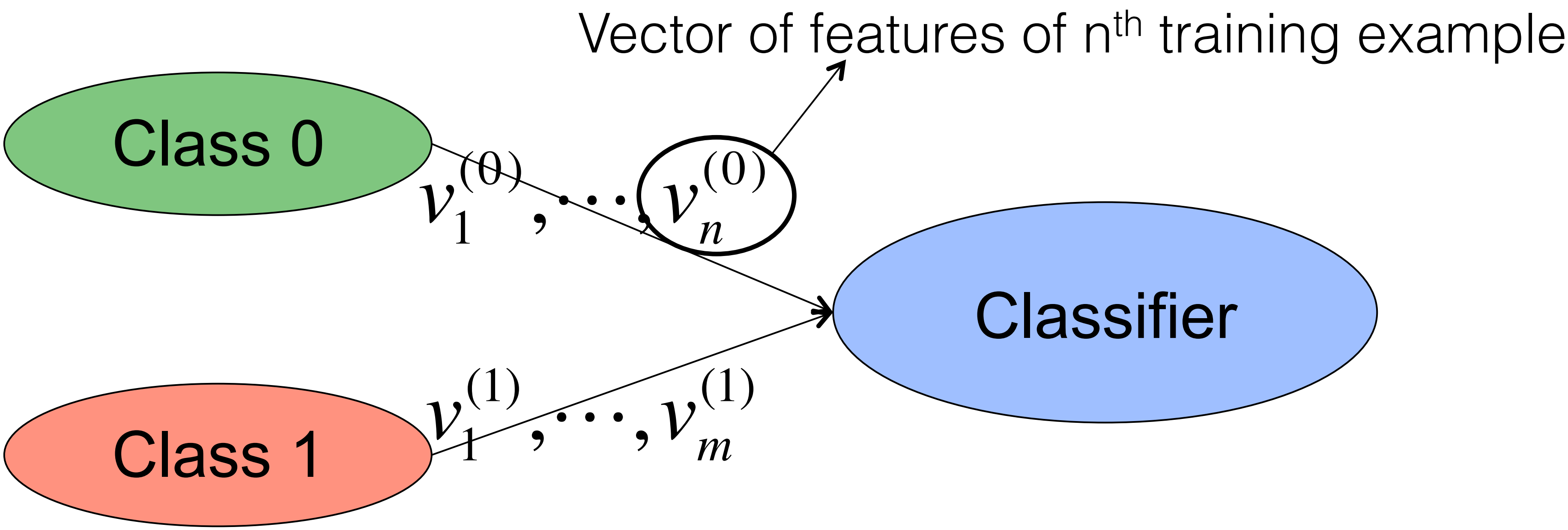
Content

Main objective: illustrate what game theory brings to the question "how to learn?" on the example of:

Classification from strategic data

1. Problem formulation
2. The adversarial learning approach
3. The game-theoretic approach
   a. Intrusion detection games
   b. Classification games

Binary classification

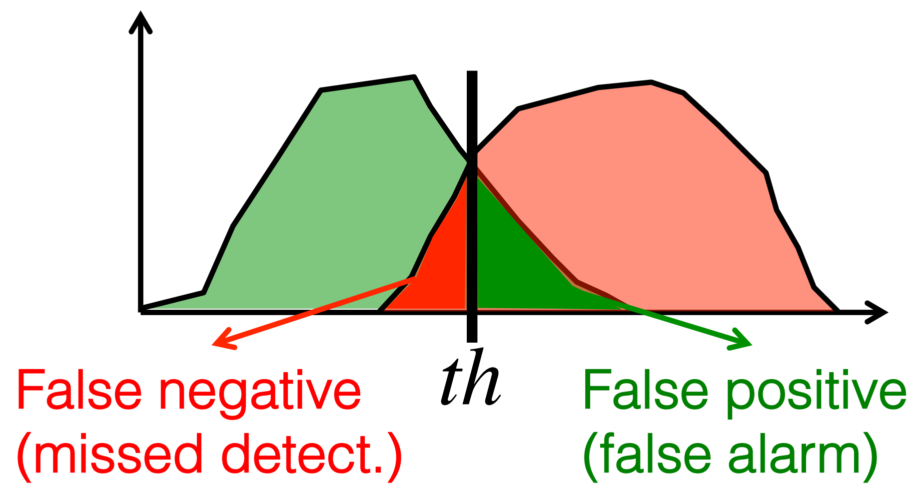


- Classifier's task
  - From $v_1^{(0)}, \cdots, v_n^{(0)}, v_1^{(1)}, \cdots, v_m^{(1)}$, make decision boundary
  - Classify new example v based on which side of the boundary

Binary classification

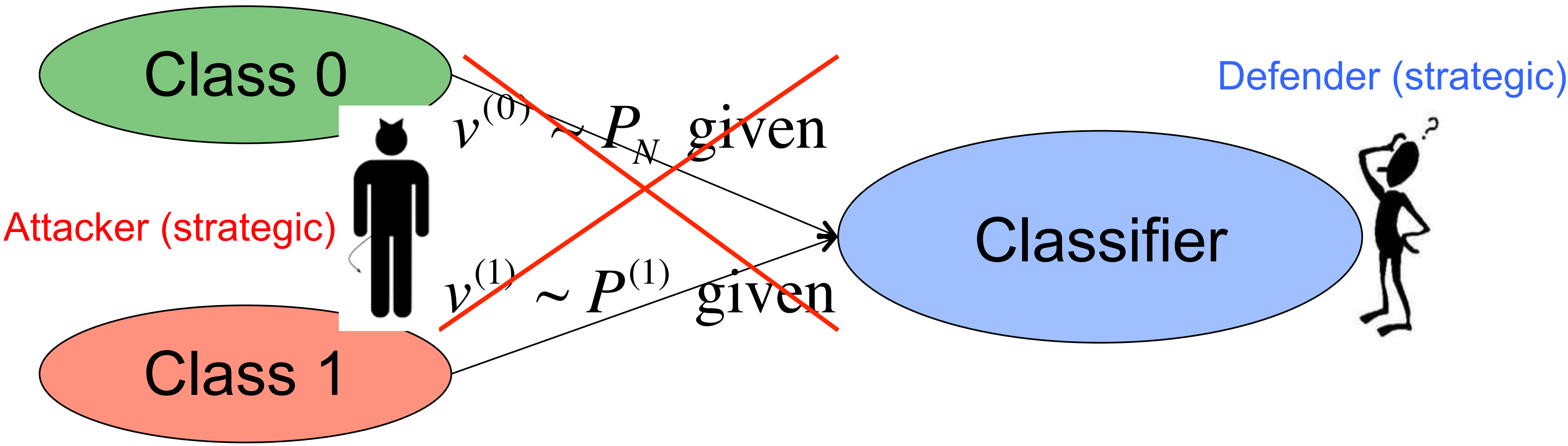
- Single feature ($v_1^{(0)}, \cdots$ scalar)

New example v: $\begin{array}{l} \text{class } 0 \text{ if } v < th \\ \text{class } 1 \text{ if } v > th \end{array}$



- Multiple features ($v_1^{(0)}, \cdots$ vector)
  - Combine features to create a decision boundary
  - Logistic regression, SVM, Naïve Bayes, etc.

Binary classification from strategic data



- Attacker modifies the data in some way in reaction to the classifier

Content

Main objective: illustrate what game theory brings to the question "how to learn?" on the example of:

Classification from strategic data

1. Problem formulation
2. The adversarial learning approach
3. The game-theoretic approach
   a. Intrusion detection games
   b. Classification games

Machine learning and security literature

- A large literature at the intersection of machine learning and security since mid-2000
  - [Huang et al., AISec ’11]
  - [Biggio et al., ECML PKDD ’13]
  - [Biggio, Nelson, Laskov, ICML ’12]
  - [Dalvi et al., KDD ’04]
  - [Lowd, Meek, KDD ’05]
  - [Nelson et al., AISTATS ’10, JMLR ’12]
  - [Miller et al. AISec ’04]
  - [Barreno, Nelson, Joseph, Tygar, Mach Learn ’10]
  - [Barreno et al., AISec ’08]
  - [Rubinstein et al., IMC ’09, RAID ’08]
  - [Zhou et al., KDD ’12]
  - [Wang et al., USENIX SECURITY ’14]
  - [Zhou, Kantarcioglu, SDM ’14]
  - [Vorobeychik, Li, AAMAS ’14, SMA ’14, AISTATS ’15]
  - …

Different ways of altering the data

- Two main types of attacks:
  - Causative: the attacker can alter the training set
    - Poisoning attack
  - Exploratory: the attacker cannot alter the training set
    - Evasion attack

- Many variations:
  - Targeted vs indiscriminate
  - Integrity vs availability
  - Attacker with various level of information and capabilities

- Full taxonomy in [Huang et al., AISec ’11]

Poisoning attacks

- General research questions
  - What attacks can be done?
    - Depending on the attacker capabilities
  - What defense against these attacks?

- 3 examples of poisoning attacks
  - SpamBayes
  - Anomaly detection with PCA
  - Adversarial SVM

Poisoning attack example (1): SpamBayes [Nelson et al., 2009]

- SpamBayes: simple content based spam filter
- 3 attacks with 3 objectives:
  - Dictionary attack: send spam with all token so user disables filter
    - Controlling 1% of the training set is enough
  - Focused attack: make a specific email appear spam
    - Works in 90% of the cases
  - Pseudospam attack: send spam that gets mislabeled so that user receives spam
    - User receives 90% of spam if controlling 10% of the training set
- Counter-measure: RONI (Reject on negative impact)
  - Remove from the training set examples that have a large negative impact

Poisoning attack example (2): Anomaly detection using PCA [Rubinstein et al. 09]

- Context: detection of DoS attacks through anomaly detection; using PCA to reduce dimensionality

- Attack: inject traffic during training to alter the principal components to evade detection of the DoS attack
  - With no poisoning attack: 3.67% evasion rate
  - 3 levels of information on traffic matrices, injecting 10% of the traffic
    - Uninformed → 10% evasion rate
    - Locally informed (on link to be attacked) → 28% evasion rate
    - Globally informed → 40% evasion rate

- Defense: “robust statistics”
  - Maximize maximum absolute deviation instead of variance

Poisoning attack example (3): adversarial SVM [Zhou et al., KDD '12]

- Learning algorithm: support vector machine
- Adversary's objective: alter the classification by modifying the features of class 1 training examples
  - Restriction on the range of modification (possibly dependent on the initial feature)
- Defense: minimize SVM cost with worse-case possible attack
  - Zero-sum game "in spirit"

Evasion attacks

- Fixed classifier, general objective of evasion attacks:
  - By querying the classifier, find a “good” negative example
- “Near optimal evasion”: find negative instance of minimal cost
  - [Lowd, Meek, KDD ’05]: Linear classifier (with continuous features and linear cost)
    - Adversarial Classifier Reverse Engineering (ACRE): polynomial queries
  - [Nelson et al., AISTATS ’10]: extension to convex-inducing classifiers
- “Real-world evasion”: find “acceptable” negative instance
- Defenses
  - Randomization: no formalization or proofs

Content

Main objective: illustrate what game theory brings to the question “how to learn?” on the example of:

Classification from strategic data

1. Problem formulation
2. The adversarial learning approach
3. The game-theoretic approach
    a. Intrusion detection games
    b. Classification games

Game theory and security literature

- A large literature on game theory for security since mid-2000
  - Surveys:
    - [Manshaei et al., ACM Computing Survey 2011]
    - [Alpcan Basar, CUP 2011]
  - Game-theoretic analysis of intrusion detection systems
    - [Alpcan, Basar, CDC ’04, Int Symp Dyn Games ’06]
    - [Zhu et al., ACC ’10]
    - [Liu et al, Valuetools ’06]
    - [Chen, Leneutre, IEEE TIFS ’09]
  - Many other security aspects approached by game theory
    - Control [Tambe et al.]
    - Incentives for investment in security with interdependence [Kunreuther and Heal 2003], [Grossklags et al. 2008], [Jiang, Anantharam, Walrand 2009], [Kantarcioglu et al, 2010]
    - Cyber insurance [Lelarge, Bolot 2008-2012], [Boehme, Schwartz 2010], [Shetty, Schwartz, Walrand 2008-2012], [Schwartz et al. 2014]
    - Economics of security [Anderson, Moore 2006]
    - Robust networks design: [Gueye, Anantharam, Walrand, Schwartz 2011-2013], [Laszka et al, 2013-2015]
    - …

Intrusion Detection System (IDS): simple model

- IDS: Detect unauthorized use of network
  - Monitor traffic and detect intrusion (signature or anomaly based)
  - Monitoring has a cost (CPU (e.g., for real time))

- Simple model:
  - Attacker: {attack, no attack} ({a, na})
  - Defender: {monitoring, no monitoring} ({m, nm})
  - Payoffs

$$P^A = \begin{array}{cc} m & nm \\ \end{array} \qquad P^D = \begin{array}{cc} m & nm \end{array}$$

$$P^A = \begin{bmatrix} -\beta_c & \beta_s \\ 0 & 0 \end{bmatrix}, \quad P^D = \begin{array}{c} \begin{bmatrix} \alpha_c & -\alpha_s \\ -\alpha_f & 0 \end{bmatrix} \end{array} \begin{array}{l} a \\ na \end{array}$$

  - "Safe strategy" (or min-max)
    - Attacker: na
    - Defender: m if $\alpha_s > \alpha_f$, nm if $\alpha_s < \alpha_f$

Nash equilibrium: mixed strategy (i.e., randomized)

- Payoffs:

$$P^A = \begin{bmatrix} -\beta_c & \beta_s \\ 0 & 0 \end{bmatrix}, \quad P^D = \begin{array}{cc} \begin{array}{cc} \text{m} & \text{nm} \end{array} & \\ \begin{bmatrix} \alpha_c & -\alpha_s \\ -\alpha_f & 0 \end{bmatrix} & \begin{array}{c} \text{a} \\ \text{na} \end{array} \end{array}$$

- Non-zero sum game
- There is no pure strategy NE
- Mixed strategy NE:

$$p_a = \frac{\alpha_f}{\alpha_f + \alpha_c + \alpha_s}, \quad p_m = \frac{\beta_s}{\beta_c + \beta_s}$$

  - Be unpredictable
  - Neutralize the opponent (make him indifferent)
  - Opposite of own optimization (indep. own payoff)

Game-theoretic analysis of intrusion detection

- In networks:
  - [Alpcan, Basar '04 '06 '11]
    - Initial papers
  - [Chen, Leneutre '09]
    - Nash equilibrium with heterogeneous values targets
  - [Liu et al. '06]
    - Bayesian games
  - [Zhu et al. '10]
    - Stochastic games
- In key physical locations (airports, ports, etc.)
  - [Tambe et al. ~'00—present]
    - Stackelberg equilibrium

Heterogeneous networks [Chen, Leneutre, IEEE TIFS 2009]

- N independent targets T={1, …, N}

- Target *i* has value W_i
- Payoff of attack for target *i*

	Monitor	Not monitor
Attack	$(1-2a)W_i - C_a W_i,$ $-(1-2a)W_i - C_m W_i$	$W_i - C_a W_i, -W_i$
Not attack	$0, -bC_f W_i - C_m W_i$	$0, 0$

- Total payoff: sum on all targets

- Strategies
  - Attacker chooses {p_i, i=1..N}, proba to attack i $\sum_i p_i \leq P$
  - Defender chooses {q_i, i=1..N}, proba to monitor i $\sum_i q_i \leq Q$

Sensible targets

- Sets T_S (sensible targets) T_Q (quasi-sensible targets) uniquely defined by

Definition 3: The sensible target set $\mathcal{T}_S$ and the quasi-sensible target set $\mathcal{T}_Q$ are defined such that:

$$\begin{cases} W_i > \dfrac{|\mathcal{T}_S| \cdot (1 - C_a) - 2aQ}{(1 - C_a)(\sum_{j \in \mathcal{T}_S} \frac{1}{W_j})} & \forall i \in \mathcal{T}_S \\ W_i = \dfrac{|\mathcal{T}_S| \cdot (1 - C_a) - 2aQ}{(1 - C_a)(\sum_{j \in \mathcal{T}_S} \frac{1}{W_j})} & \forall i \in \mathcal{T}_Q \\ W_i < \dfrac{|\mathcal{T}_S| \cdot (1 - C_a) - 2aQ}{(1 - C_a)(\sum_{j \in \mathcal{T}_S} \frac{1}{W_j})} & \forall i \in \mathcal{T} - \mathcal{T}_S - \mathcal{T}_Q \end{cases} \quad (1)$$

High value

Low value

where $|\mathcal{T}_S|$ is the cardinality of $\mathcal{T}_S$, $\mathcal{T} - \mathcal{T}_S - \mathcal{T}_Q$ denotes the set of targets in the target set $\mathcal{T}$ but neither in $\mathcal{T}_S$ nor in $\mathcal{T}_Q$.

- Theorem:
  - A rational attack does not attack in $T - T_S - T_Q$
  - A rational defender does defend in $T - T_S - T_Q$

Nash equilibrium – case 1

- Attacker and defender use up all their available resources: $\sum_i p_i = P$ and $\sum_i q_i = Q$

- Nash equilibrium given by

$$p_i^* = \begin{cases} \frac{P_A}{W_i \sum_{j=1}^{N_A} \frac{1}{W_j}} - \left(\frac{N_A}{W_i \sum_{j=1}^{N_A} \frac{1}{W_j}} - 1 \right) \cdot \frac{bC_f + C_m}{2a + bC_f}, & i \in \mathcal{T}_S \\ \in \left[0, \frac{P_A}{W_i \sum_{j=1}^{N_A} \frac{1}{W_j}} - \left(\frac{N_A}{W_i \sum_{j=1}^{N_A} \frac{1}{W_j}} - 1 \right) \cdot \frac{bC_f + C_m}{2a + bC_f} \right], & i \in \mathcal{T}_Q \\ 0, & i \in \mathcal{T} - \mathcal{T}_S - \mathcal{T}_Q \end{cases}$$

$$q_i^* = \begin{cases} \frac{1}{2a} \left(1 - C_a - \frac{N_A(1 - C_a) - 2aQ}{W_i \sum_{j=1}^{N_A} \frac{1}{W_j}} \right), & i \in \mathcal{T}_S \\ 0, & i \in \mathcal{T} - \mathcal{T}_S \end{cases}$$

Sensible (and quasi-sensible) nodes attacked and defended

Non-sensible nodes not attacked and not defended

Nash equilibrium – case 2

- If the attack power P is low relative to the cost of monitoring, the defender does not use all his available resources: $\sum_i p_i = P$ and $\sum_i q_i < Q$

- Nash equilibrium given by

$$p_i^* \begin{cases} = \frac{bC_f + C_m}{2a + bC_f}, & W_i > W_{N_D+1} \\ \in \left[0, \frac{bC_f + C_m}{2a + bC_f}\right], & W_i = W_{N_D+1} \\ = 0, & W_i < W_{N_D+1} \end{cases}$$

$$q_i^* = \begin{cases} \frac{1 - C_a}{2a}\left(1 - \frac{W_{N_D+1}}{W_i}\right), & W_i > W_{N_D+1} \\ 0, & W_i \leq W_{N_D+1} \end{cases}$$

where $N_D = \lfloor (2a + bC_f)P/(bC_f + C_m) \rfloor$

Sensible (and quasi-sensible) nodes attacked and defended

Non-sensible nodes not attacked and not defended

Monitor more the targets with higher values

Nash equilibrium – case 3

- If P and Q are large, or cost of monitoring/attack is too large, neither attacker nor defender uses all available resources: $\sum_i p_i < P$ and $\sum_i q_i < Q$
- Nash equilibrium given by

$$\begin{cases} p_i^* = \dfrac{bC_f + C_m}{2a + bC_f} \\ q_i^* = \dfrac{1 - C_a}{2a} \end{cases} \qquad i \in \mathcal{T}$$

  - All targets are sensible
  - Equivalent to N independent IDS
  - Monitoring/attack independent of W_i
    - Due to payoff form (cost of attack proportional to value)

➤ All IDS work: assumption that payoff is sum on all targets

EURECOM Sophia Antipolis

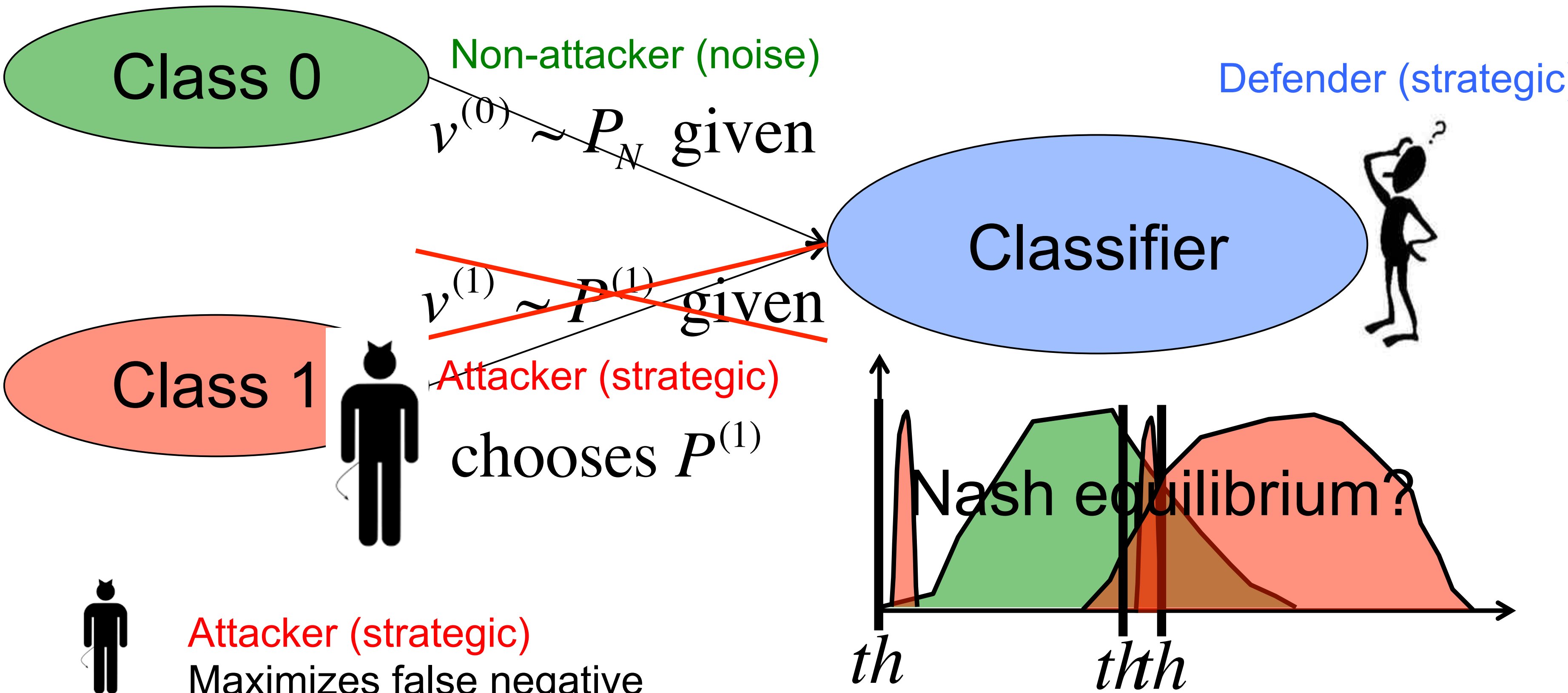
Content

Main objective: illustrate what game theory brings to the question "how to learn?" on the example of:

Classification from strategic data

1. Problem formulation
2. The adversarial learning approach
3. The game-theoretic approach
   a. Intrusion detection games
   b. Classification games

Classification games



Attacker (strategic)
Maximizes false negative

Defender (strategic)
~~Minimizes false negative (zero-sum)~~

A first approach

- [Brückner, Scheffer, KDD ’12, Brückner, Kanzow, Scheffer, JMLR ’12]

- Model:
  - Defender selects the parameters of a pre-specified generalized linear model
  - Adversary selects a modification of the features
  - Continuous cost in the probability of class 1 classification

- Result:
  - Pure strategy Nash equilibrium

A more flexible model [Dritsoula, L., Musacchio, 2012, 2015]

- Model specification

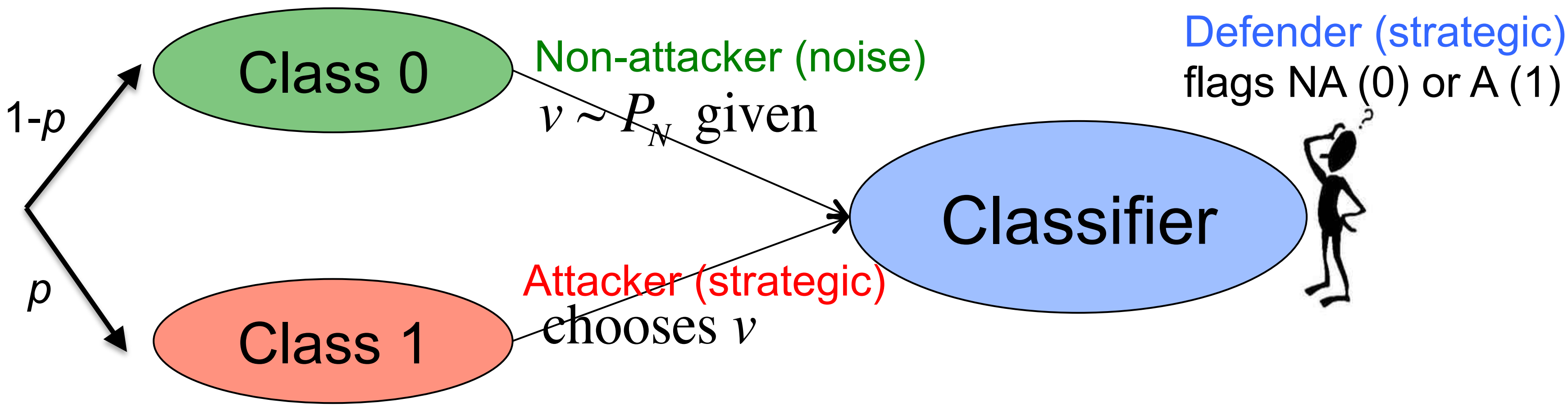
- Game-theoretic analysis to answer the questions:

  - How should the defender perform classification?
    - How to combine the features?
    - How to select the threshold?

  - How will the attacker attack?
    - How does the attacker select the attacks features?

  - How does the performance change with the system's parameters?

Model: players and actions



- Attacker chooses $v \in V$ → Set of feature vectors
- Defender chooses $c \in C$ → Set of classifiers $\{0,1\}^{|V|}$
  - Classifier $c : V \to \{0,1\}$
- Two-players game $G = \langle V, C, \underbrace{P_N, p, c_d, c_{fa}}_{\text{Payoff-relevant Parameters}} \rangle$

Model: payoffs

- Attacker's payoff:

Cost if detected

$$U^A(v,c) = R(v) - c_d 1_{c(v)=1}$$

Reward from attack

- Defender's payoff:

Cost of false alarm

$$U^D(v,c) = p\left(-R(v) + c_d 1_{c(v)=1}\right) + (1-p)c_{fa}\left(\sum_{v' \in V} P_N(v') 1_{c(v')=1}\right)$$

Rescaling

$$U^D(v,c) = -U^A(c,v) + \frac{(1-p)}{p} c_{fa}\left(\sum_{v' \in V} P_N(v') 1_{c(v')=1}\right)$$

EURECOM Sophia Antipolis

Nash equilibrium

- Mixed strategies:
  - Attacker: probability distribution α on V
  - Defender: probability distribution β on C

- Utilities extended: $U^A(\alpha,\beta) = \sum_{v\in V}\sum_{c\in C}\alpha_v U^A(v,c)\beta_c$

- Nash equilibrium: (α,β) s.t. each player is at best-response:

$$\alpha^* \in \operatorname*{argmax}_{\alpha} U^A(\alpha,\beta^*)$$

$$\beta^* \in \operatorname*{argmax}_{\beta} U^D(\alpha^*,\beta)$$

"Easy solution": linear programming (almost zero-sum game)

$$U^A(v,c) = R(v) - c_d 1_{c(v)=1} - \frac{(1-p)}{p} c_{fa} \left(\sum_{v' \in V} P_N(v') 1_{c(v')=1} \right)$$

$$U^D(v,c) = -U^A(c,v) + \frac{(1-p)}{p} c_{fa} \left(\sum_{v' \in V} P_N(v') 1_{c(v')=1} \right)$$

- The non-zero-sum part depends only on $c \in C$
- Best-response equivalent to zero-sum game
- Solution can be computed by LP, BUT
  - The size of the defender's action set is large
  - Gives no information on the game structure

Main result 1: defender combines features based on attacker's reward

- Define C^T: set of threshold classifiers on $R(v)$

$$C^T = \left\{ c \in C : c(v) = 1_{R(v) \geq t} \ \forall v, \text{ for some } t \in \Re \right\}$$

Theorem:

For every NE of $G = \langle V, C, P_N, p, c_d, c_{fa} \rangle$, there exists a NE of $G^T = \langle V, C^T, P_N, p, c_d, c_{fa} \rangle$ with the same attacker's strategy and the same equilibrium payoffs

- Classifiers that compare $R(v)$ to a threshold are optimal for the defender
  - Different from know classifiers (logistic regression, etc.)
  - Reduces a lot the size of the defender's strategy set

Main result 1: proof's key steps

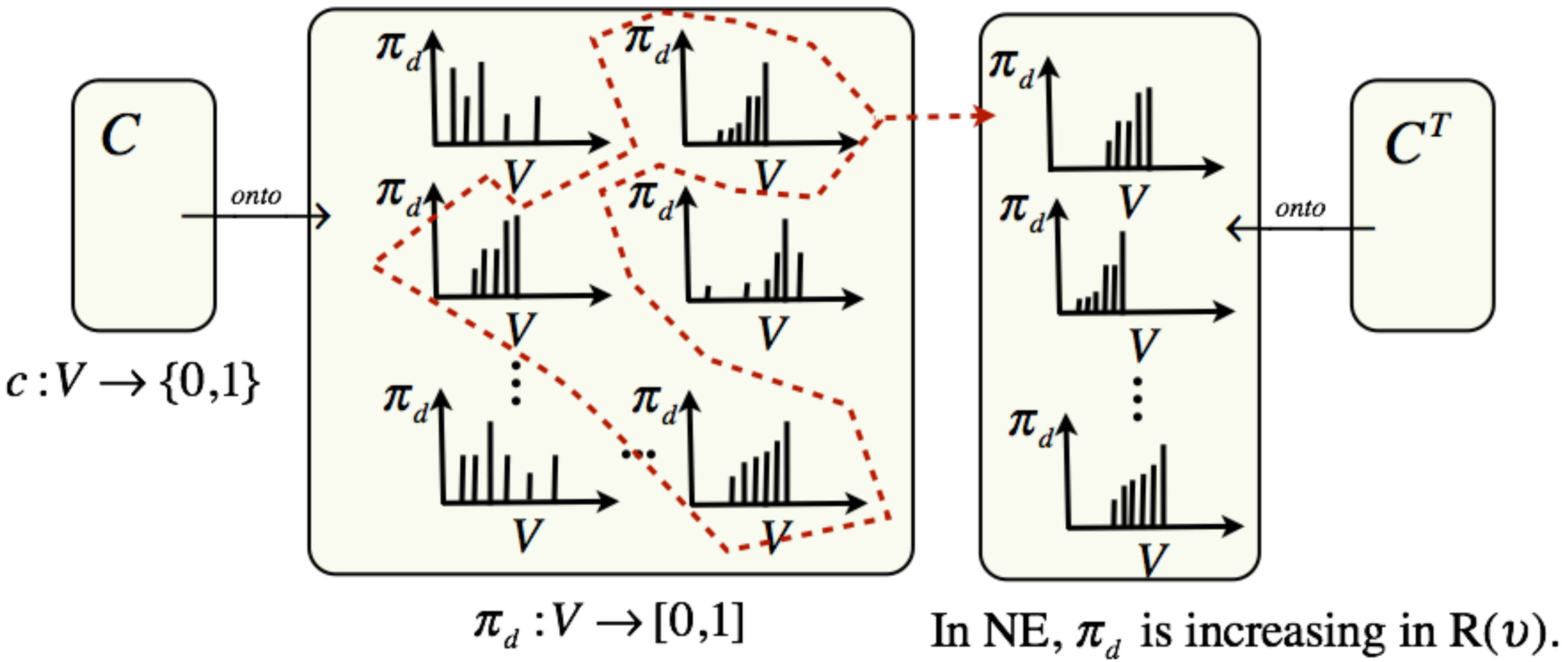
1. The utilities depend on β only through the probability of class 1 classification:

$$\pi_d(v) = \sum_{c \in C} \beta_c 1_{c(v)=1}$$

2. At NE, if $P_N(v) > 0$ for all v, then $\pi_d(v)$ increases with $R(v)$

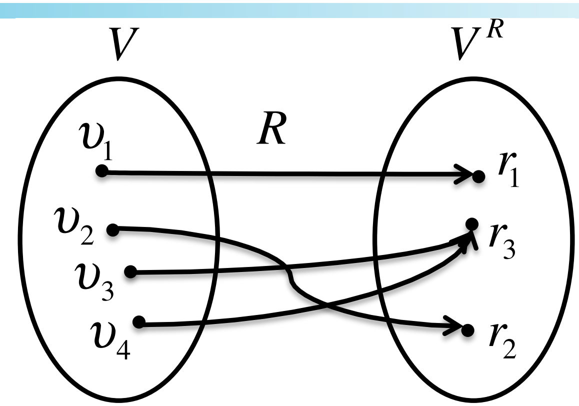
3. Any $\pi_d(v)$ that increases with $R(v)$ can be achieved by a mix of threshold strategies in C^T

Main result 1: illustration



Reduction of the attacker's strategy space

- V^R: set of rewards



Proposition:

$G^T = \langle V, C^T, P_N, p, c_d, c_{fa} \rangle$ and $G^{R,T} = \langle V^R, C^T, P_N^R, p, c_d, c_{fa} \rangle$
have the same equilibrium payoffs

- $P_N^R(r) = \sum_{v:R(v)=r} P_N(v)$: non-attacker's probability on V^R

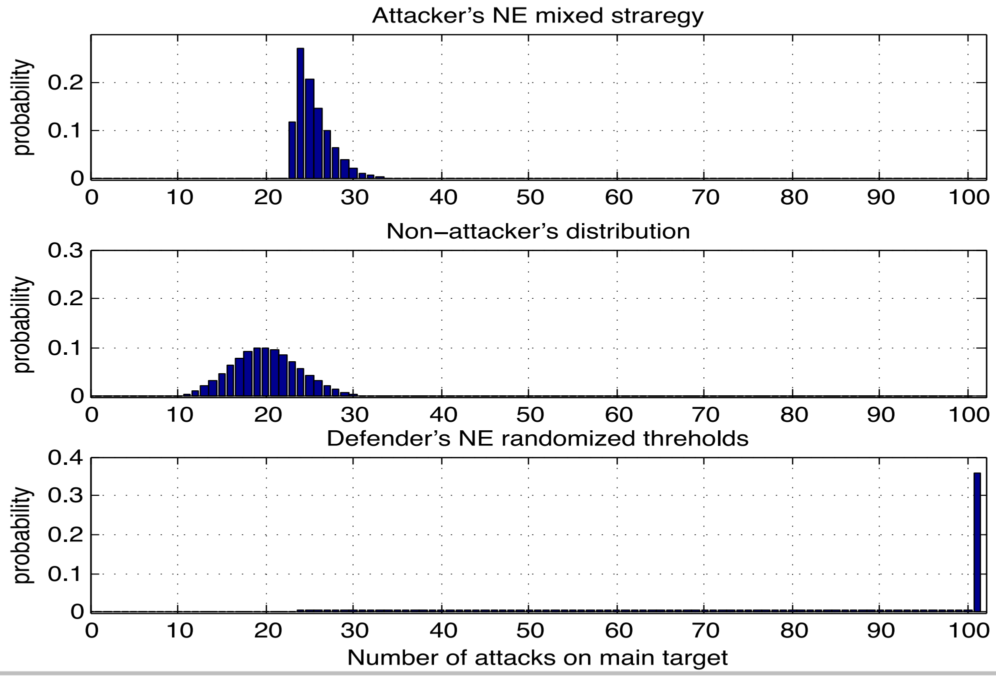
➢ It is enough to study $G^{R,T} = \langle V^R, C^T, P_N^R, p, c_d, c_{fa} \rangle$

Main result 2: attacker's equilibrium strategy mimics the non-attacker

Lemma:

If (α, β) is a NE of $G = \langle V, C, P_N, p, c_d, c_{fa} \rangle$, then

$$\alpha_v = \frac{1-p}{p} \frac{c_{fa}}{c_d} P_N(v), \text{ for all v s.t. } \pi_d(v) \in (0,1)$$



- Attacker's strategy: scaled version of the non-attacker distribution on a subset

EURECOM Sophia Antipolis

Game rewriting in matrix form

$$\left|C^T\right| = \left|V^R\right| + 1$$

- Game $G^{R,T} = \left\langle V^R, C^T, P_N^R, p, c_d, c_{fa} \right\rangle$
  - Attacker chooses attack reward in $V^R = \{r_1 < r_2 < \cdots\}$
  - Defender chooses threshold strategy in C^T

$$U^A(\alpha,\beta) = -\alpha'\Lambda\beta \quad \text{and} \quad U^D = \alpha'\Lambda\beta - \mu'\beta$$

$$\Lambda = c_d \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 & 0 \\ \vdots & 1 & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & \ddots & \vdots & \vdots \\ \vdots & & & \ddots & 0 & \vdots \\ 1 & \cdots & \cdots & \cdots & 1 & 0 \end{pmatrix} - \begin{pmatrix} r_1 \\ \vdots \\ \vdots \\ \vdots \\ r_{\left|V^R\right|} \end{pmatrix} \cdot 1'_{\left|V^R\right|+1}$$

$$\mu_i = \frac{1-p}{p} c_{fa} \sum_{r \geq r_i} P_N^R(r)$$

EURECOM Sophia Antipolis

Main result 3: Nash equilibrium structure (i.e., how to choose the threshold)

Theorem:

At a NE of $G^{R,T} = \left\langle V^R, C^T, P_N^R, p, c_d, c_{fa} \right\rangle$, for some k:

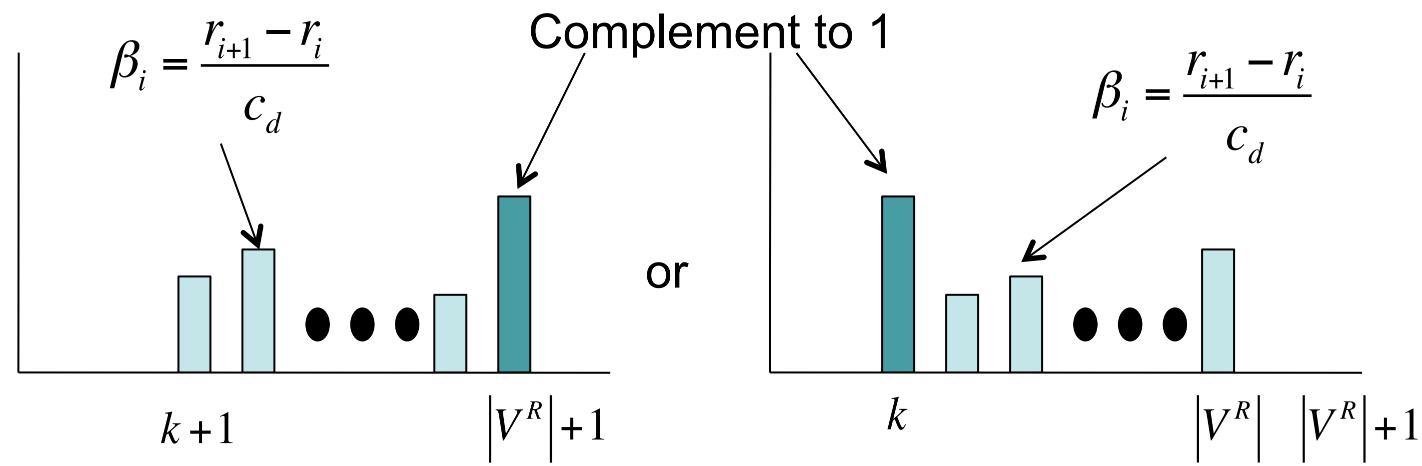
- The attacker's strategy is $\left(0,\cdots,0,\alpha_k,\cdots,\alpha_{|V^R|}\right)$
- The defender's strategy is $\left(0,\cdots,0,\beta_k,\cdots,\beta_{|V^R|},\beta_{|V^R|+1}\right)$

where
$$\beta_i = \frac{r_{i+1} - r_i}{c_d}, \text{ for } i \in \left\{k+1,\cdots,\left|V^R\right|\right\}$$
$$\alpha_i = \frac{1-p}{p}\frac{c_{fa}}{c_d}P_N^R(r_i), \text{ for } i \in \left\{k+1,\cdots,\left|V^R\right|-1\right\}$$

NE computation

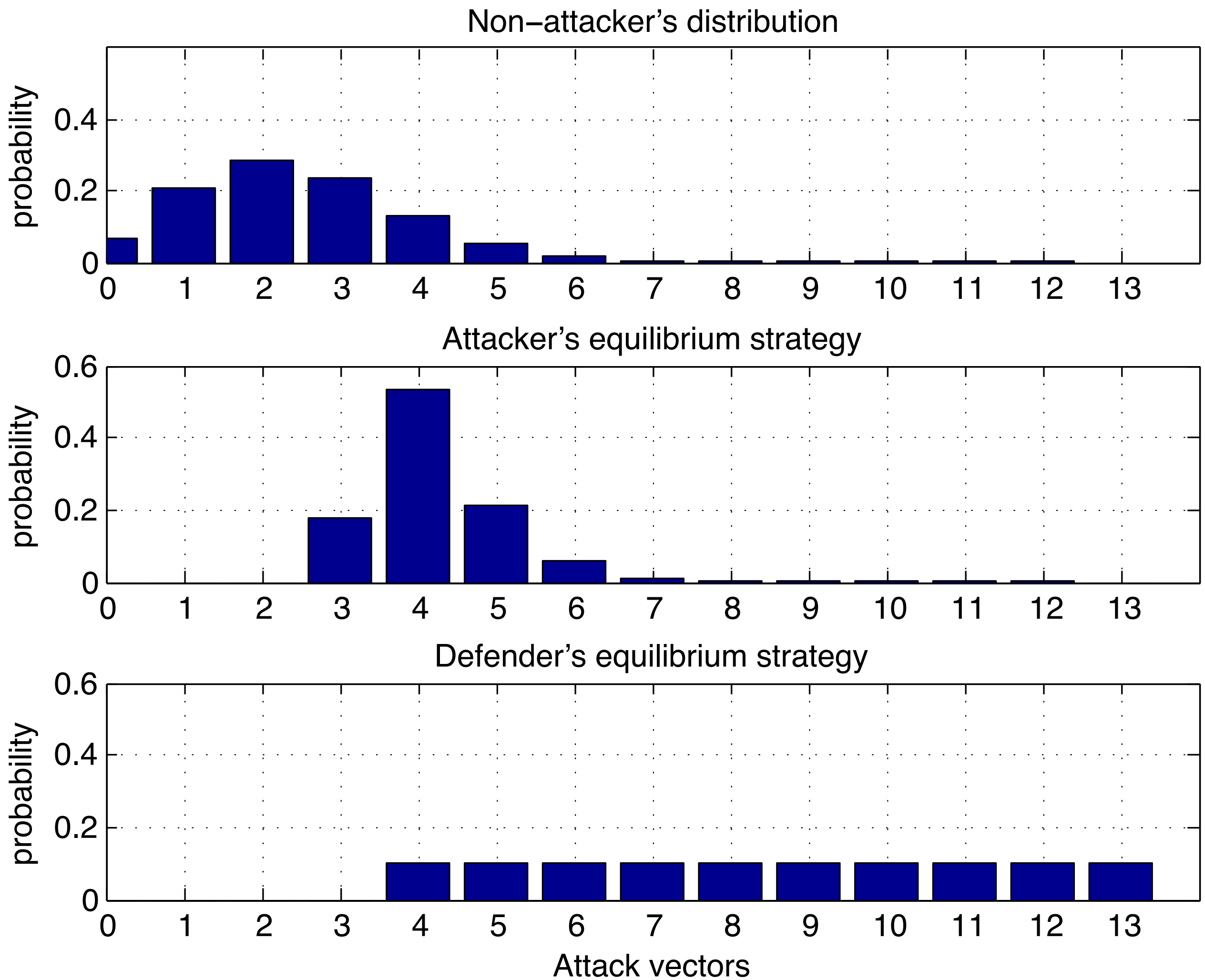
- Defender: try all vectors β of the form (for all *k*)

β:
Mix of defender threshold strategies



- Take the one maximizing payoff
  - Unique maximizing β → unique NE.
  - Multiple maximizing β → any convex combination is a NE
- Attacker: Use the formula
  - Complete first and last depending on β

Nash equilibrium illustration



- Case

$$r_i = i \cdot c_a$$

Main result 3: proof's key steps

1. At NE, β maximizes $\min \Lambda\beta - \mu'\beta$

- Solve LP: $$\begin{aligned} &\text{maximize} && z - \mu'\beta \\ &\text{s.t.} && \Lambda\beta \geq z \cdot 1_{|V^R|},\ \beta \geq 0,\ 1_{|V^R|+1} \cdot \beta = 1 \end{aligned}$$

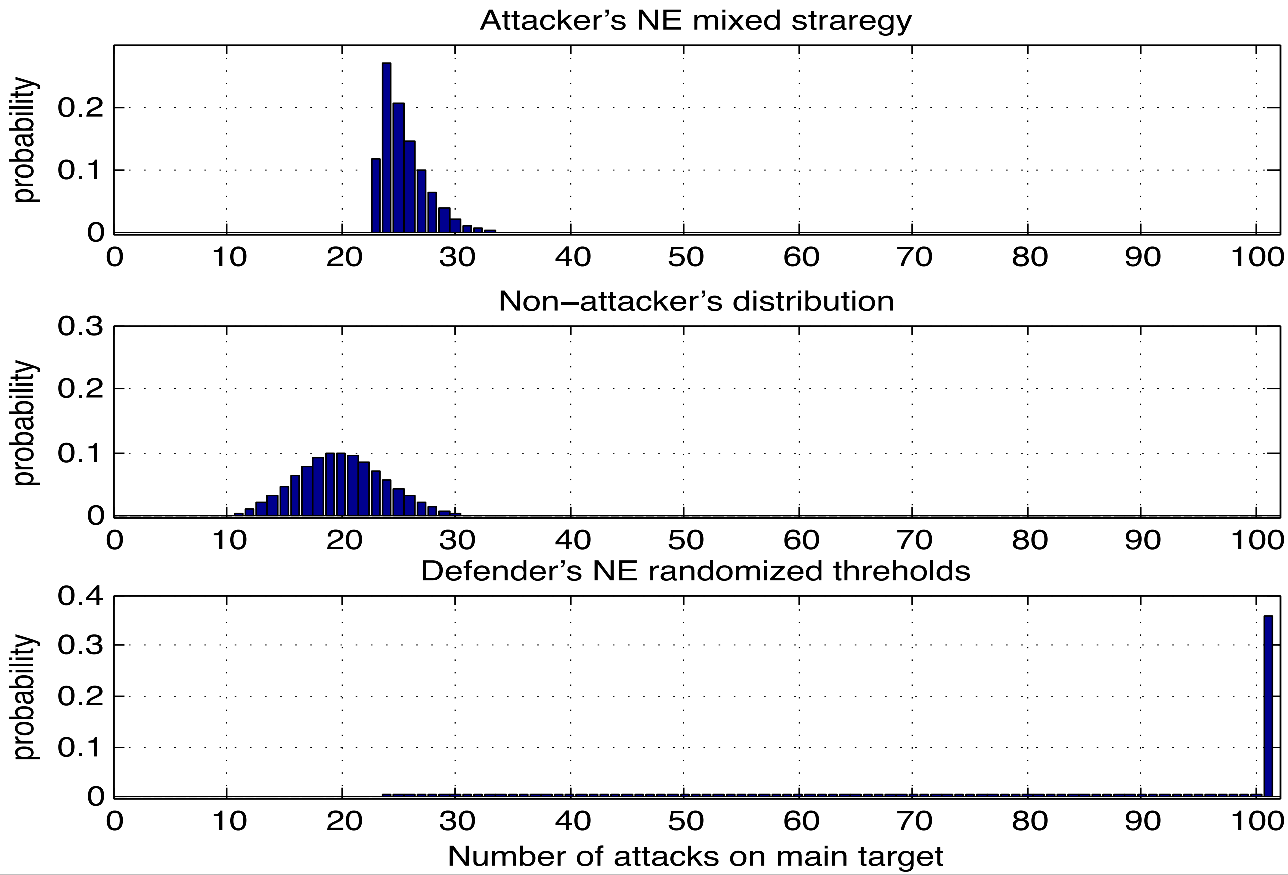
- extreme points of $\Lambda x \geq 1_{|V^R|},\ x \geq 0 \quad \left(\beta = x / \|x\|\right)$

2. Look at polyhedron and eliminate points that are not extreme

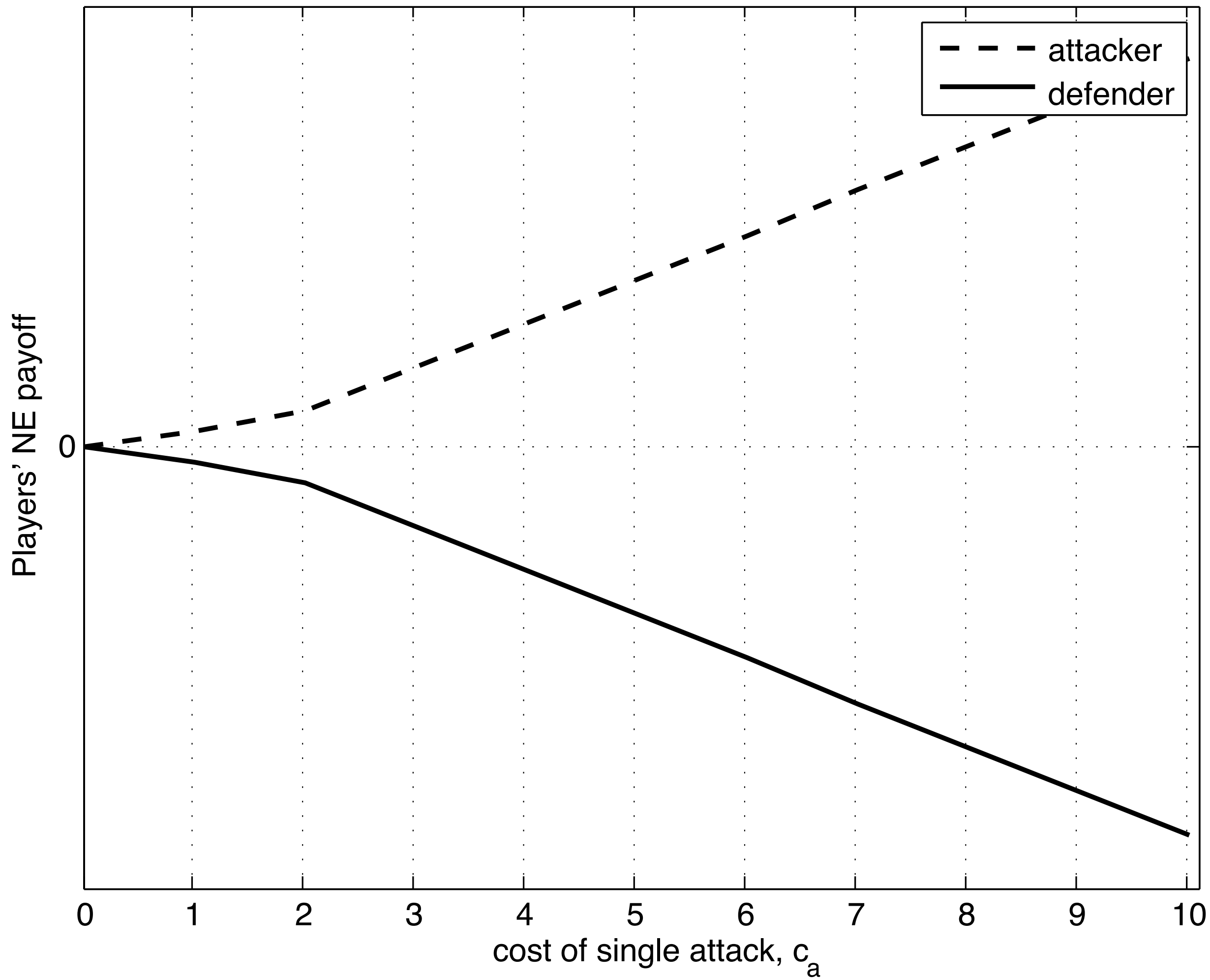
$$\begin{aligned} c_d x_1 + (r_{|V^R|} - r_1 + \varepsilon)\|x\| &\geq 1 \\ c_d (x_1 + x_2) + (r_{|V^R|} - r_2 + \varepsilon)\|x\| &\geq 1 \\ &\vdots \\ c_d (x_1 + x_2 + \cdots + x_{|V^R|}) + \varepsilon\|x\| &\geq 1 \end{aligned}$$

Example

- Case $r_i = i \cdot c_a, N = 100, P_N \sim Bino(\theta), p = 0.2$

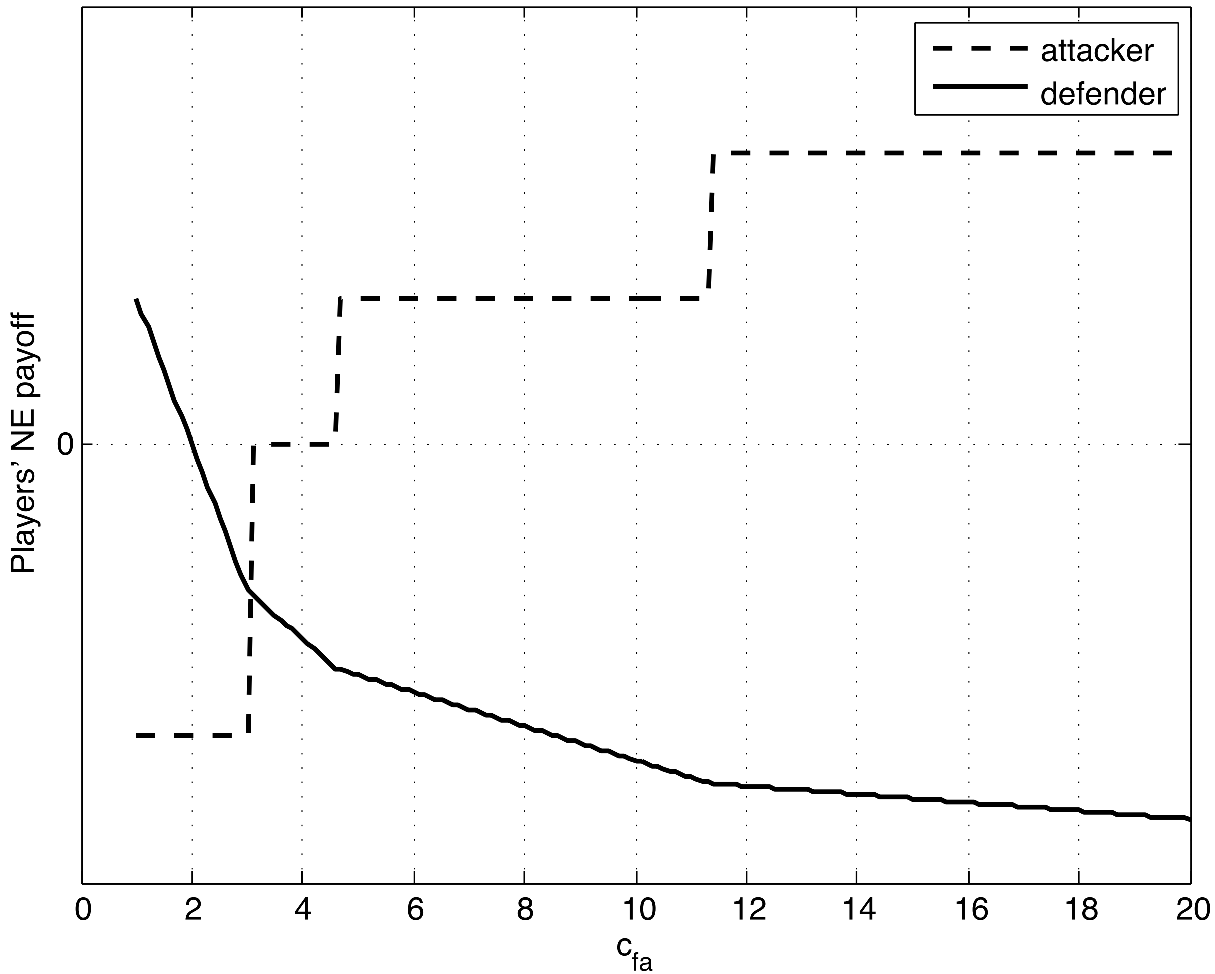


Example (2): variation with cost of attack



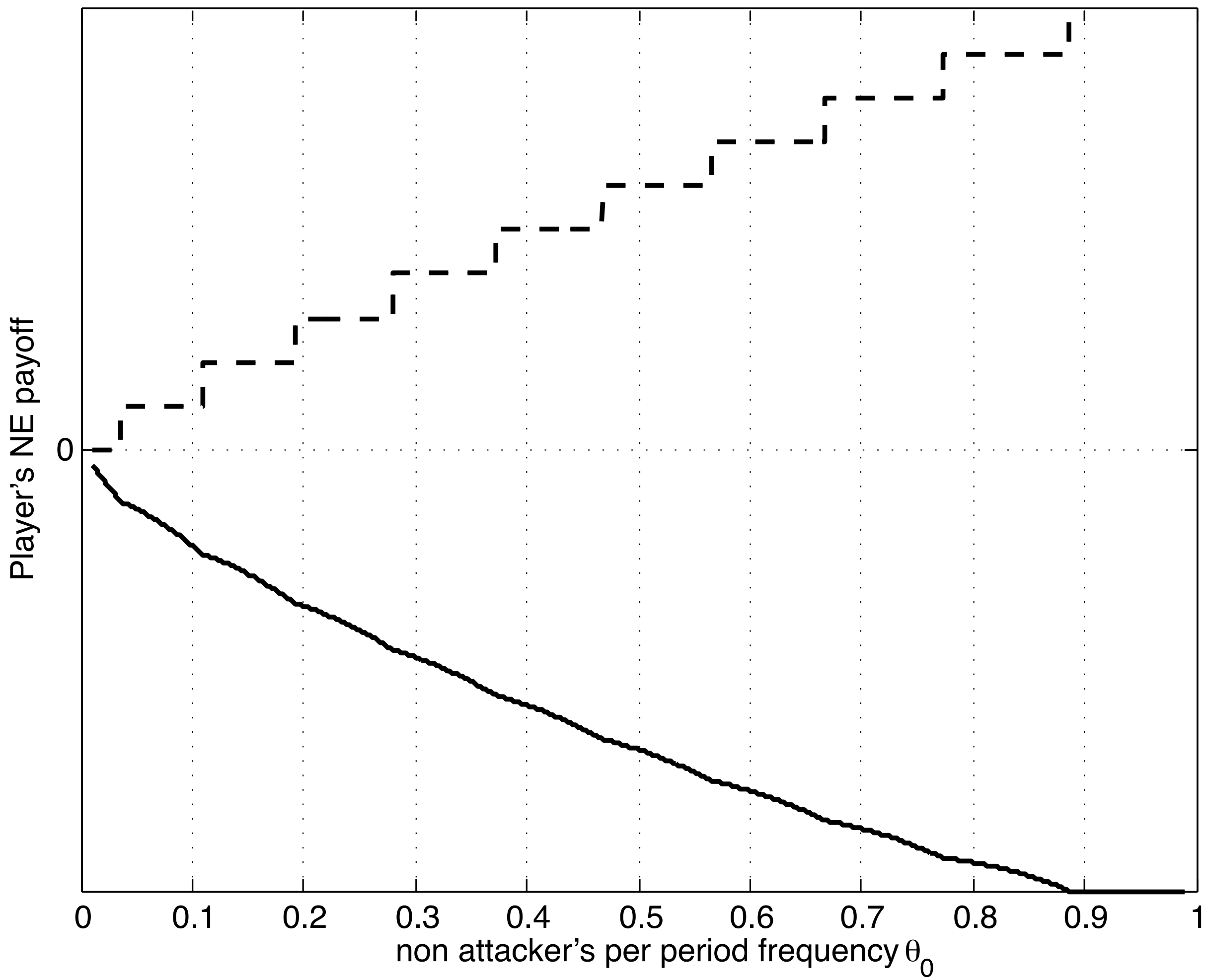
EURECOM
Sophia Antipolis

Example (3): variation with false alarm cost



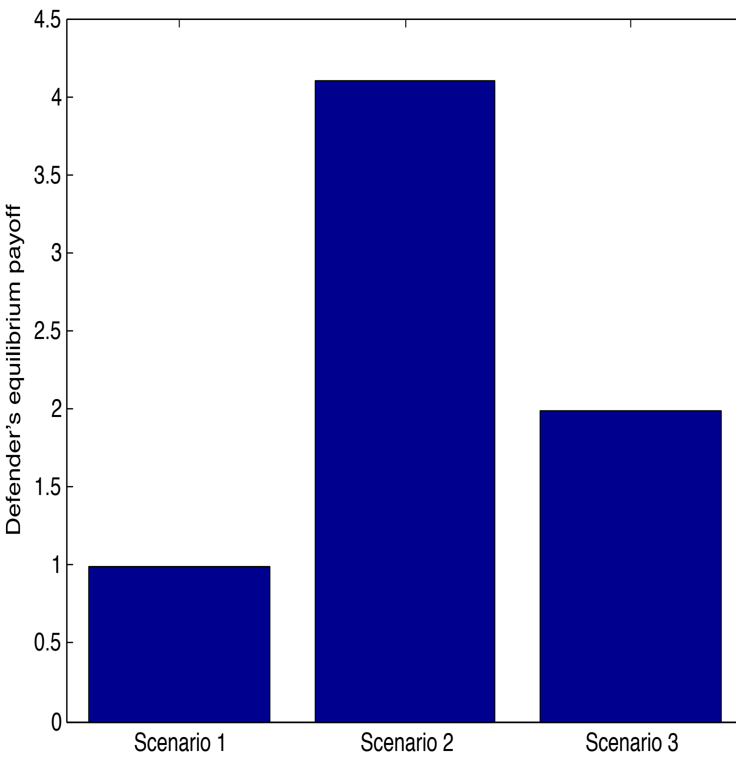
EURECOM
Sophia Antipolis

Example (4): Variation with noise strength



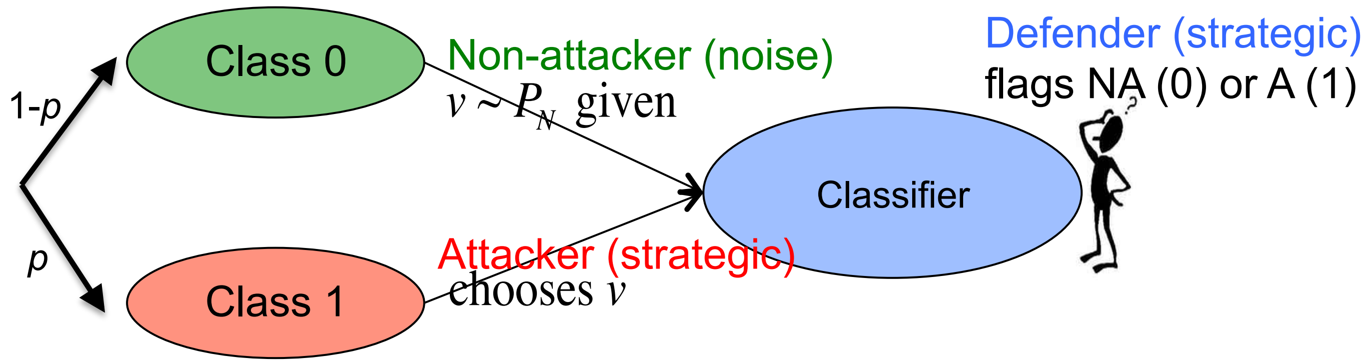
Example (5): is it worth investing in a second sensor?

- There are two features
- 3 scenarios:
  - 1: defender classifies on feature 1 only
    - Attacker uses maximal strength on feature 2
  - 2: defender classifies on features 1 and 2 but attacker doesn't know
    - Attacker uses maximal strength on feature 2
  - 3: defender classifies on features 1 and 2 and attacker knows
    - Attacker adapts strength on feature 2
- Is it worth investing?
  - Compare the investment cost to the payoff difference!



Conclusion: binary classification from strategic data

- Game theory provides new insights into learning from data generated by a strategic attacker



- Analysis of a simple model (Nash equilibrium):
  - Defender should combine features according to attacker's reward → not use a known algorithm
    - Mix on threshold strategies proportionally to marginal reward increase, up to highest threshold
  - Attacker mimics non-attacker on defender's support

Extensions and open problems

- Game theory can bring to other learning problems with strategic agents!

- Models with one strategic attacker [security]
  - Extensions of the classification problem
    - Model generalization, multiclass, regularization, etc.
  - Unsupervised learning
    - Clustering
  - Sequential learning
    - Dynamic classification

- Models with many strategic agents [privacy]
  - Linear regression, recommendation

Patrick.Loiseau@eurecom.fr

THANK YOU

Outline

1. Classification from strategic data
    a. The adversarial learning approach
    b. The game-theoretic approach
2. Linear regression from strategic data
    a. The game-theoretic approach

General motivation and questions

- An analyst wants to learn from data using linear regression
  - Medicine, economics, etc.

- Data provided by humans are revealed strategically
  - Privacy concerns: users add noise
  - Effort put by users to provide good data
  - Data manipulation

- ➢ Incentives are an integral part of the learning problem

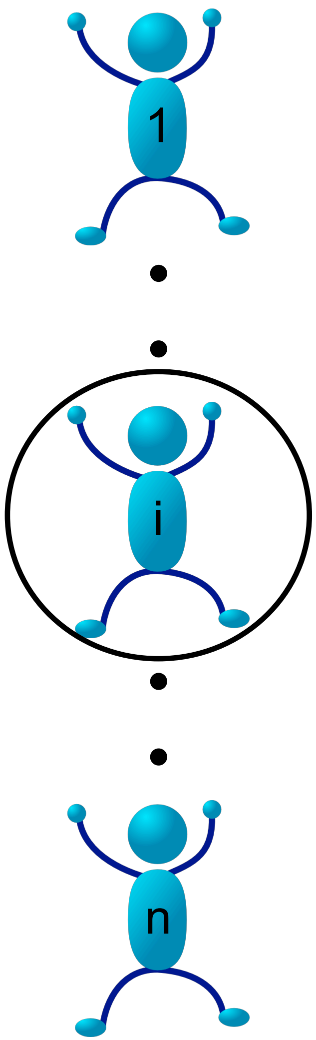
- Research questions
  - How to model users objectives? What will be the outcome?
  - What is the loss of efficiency due to strategic aspects?
  - How to design a learning algorithm that gives good incentives to users?

Why do users reveal data?

- Because they are paid for it
  - Mechanism design problem: the learning algorithm is fixed and you ask “how to pay users to obtain optimal accuracy with minimal cost”
  - [Ghosh, Roth, 2011], [Dandekar et al., 2012], [Roth, Schoenebeck, 2012], [Ligett, Roth, 2012], [Cai et al., 2015], etc.

- Because they have an interest in the result from the learning algorithm
  - Interest in the result in a user’s direction
    - What algorithm can guarantee that users don’t lie?
    - [Dekel, Fischer, Procaccia, SODA ’08]
  - Interest in the global result: information as a public good
    - Without payment, which algorithm is optimal?
    - [Ioannidis, L., WINE ’13], [Chessa, Grossklags, L., FC ’15, CSF ’15]

Model (1): linear model of user data



$$y_i = \beta^T x_i + \varepsilon_i$$

Private data of *i*
$\in R$

Model parameter
$\in R^d$
(unknown)

Public features of *i*
$\in R^d$

Inherent noise of *i*
mean 0, variance σ^2

User *i* adds noise

Added noise of *i*
mean 0, variance σ_i^2

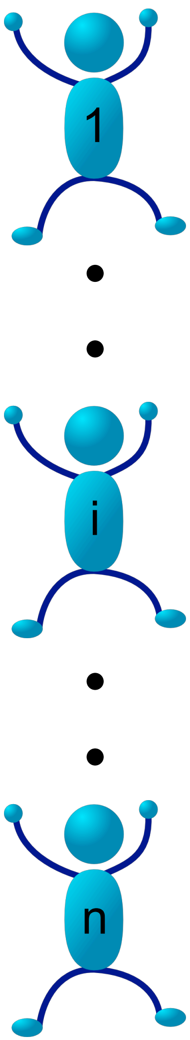
$$\tilde{y}_i = \beta^T x_i + \underbrace{\varepsilon_i + z_i}$$

Data reported by *i*
$\in R$

Total noise of *i*
mean 0, variance $\sigma^2 + \sigma_i^2$

EURECOM
Sophia Antipolis

Model (2): analyst's parameter estimation

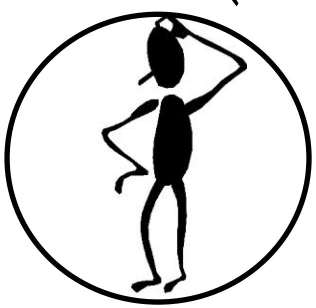


$(n \times 1)$ vector of reported data

$(n \times d)$ matrix of public features

$$\hat{\beta} = \left(X^T \Lambda X\right)^{-1} X^T \Lambda \tilde{y}$$

inverse variance of $\tilde{y}_1$



weights $\Lambda = \begin{pmatrix} \frac{1}{\sigma^2 + \sigma_1^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sigma^2 + \sigma_n^2} \end{pmatrix}$

- Generalized least-square estimator
  - Unbiased, covariance $V = \left(X^T \Lambda X\right)^{-1}$
  - Gauss-Markov/Aitken thm: smallest covariance amongst all linear unbiased estimators

Model (3): utilities/cost functions

- User *i* chooses inverse variance

$$\lambda_i = \frac{1}{\sigma^2 + \sigma_i^2} \in \left[0, 1/\sigma^2\right]$$

  – "contribution to result accuracy (public good)"

- Minimize cost

$$J_i(\lambda_i, \lambda_{-i}) = c_i(\lambda_i) + f(\lambda_i, \lambda_{-i})$$

Privacy cost
Increasing convex

Estimation cost

$$f(\lambda_i, \lambda_{-i}) = F(V(\lambda_i, \lambda_{-i}))$$

F, hence *f*, increasing convex

$$\text{Examples:} \quad F_1(\cdot) = trace(\cdot), \quad F_2(\cdot) = \|\cdot\|_F^2 = trace(\cdot^2)$$

EUREKOM
Sophia Antipolis

Nash equilibrium [Ioannidis, L., 2013]

- If $<d$ users contribute, infinite estimation cost
  ➔ trivial equilibria
- Main equilibrium result

Theorem:

There exists a unique non-trivial equilibrium

- Proof:
  - Potential game $$\Phi(\lambda_i, \lambda_{-i}) = \sum_i c_i(\lambda_i) + f(\lambda_i, \lambda_{-i})$$
  - Potential is convex

Equilibrium efficiency

- Social cost: sum of cost of all users

$$C(\vec{\lambda}) = \sum_i c_i(\lambda_i) + nf(\vec{\lambda})$$

- Inefficiency of eq. measure by price of stability:

$$PoS = \frac{C(\vec{\lambda}^{NE})}{C(\vec{\lambda}^{SO})}$$

Social cost at the non-trivial Nash equilibrium

Minimal social cost

- Remarks:
  - Same as *PoA* if we remove the trivial equilibria
  - $PoS \geq 1$, "large *PoS*: inefficient", "small *PoS*: efficient"

Equilibrium efficiency (2)

- A first result:

Theorem:

The *PoS* increases at most linearly: $PoS \leq n.$

  – Obtained only from potential structure: by positivity of the estimation and privacy costs:

$$\frac{1}{n}C(\vec{\lambda}^{NE}) \leq \Phi(\vec{\lambda}^{NE}) \leq \Phi(\vec{\lambda}^{SO}) \leq C(\vec{\lambda}^{SO})$$

  – Works for any estimation cost, i.e., any scalarization F

  – But quite rough!

Equilibrium efficiency (3) [Ioannidis, L., 2013]

- Monomial privacy costs: $c_i(\lambda_i) = c_i \cdot \lambda_i^k,\ \ c_i > 0, k \geq 1$

Theorem:

If the estimation cost is $F_1(\cdot) = trace(\cdot)$, then $PoS \leq n^{1/(k+1)}$

If the estimation cost is $F_2(\cdot) = \|\cdot\|_F^2$, then $PoS \leq n^{2/(k+2)}$

- Sharper bounds: $n^{1/2}$ for trace, $n^{2/3}$ for Frobenius
- "More convex" privacy cost → slower *PoS* increase
  - Worst case: linear privacy cost (k=1)
- Proof: KKT and $\frac{\partial tr(V(\vec{\lambda}))}{\partial \lambda_i} = -x_i^T V^2 x_i,\quad \frac{\partial \|V(\vec{\lambda})\|_F^2}{\partial \lambda_i} = -x_i^T V^3 x_i \quad \left(V = \left(X^T \Lambda X\right)^{-1}\right)$

EURECOM Sophia Antipolis

Equilibrium efficiency (4) [Ioannidis, L., 2013]

- Worst-case extends beyond monomials

Theorem:

With the estimation cost is $F_1(\cdot) = trace(\cdot)$:

if $nc_i'(\lambda) \leq c_i'(n^{1/2}\lambda)$, then $PoS \leq n^{1/2}$

With the estimation cost is $F_2(\cdot) = \|\cdot\|_F^2$:

if $nc_i'(\lambda) \leq c_i'(n^{1/3}\lambda)$, then $PoS \leq n^{2/3}$

- More general than monomials, but
  - c_i grows ~larger than λ^3 for F_1 and λ^4 for F_2
- Proof based on Brouwer's fixed-point thm

What is the best estimator? [IL '13] Aitken-like theorem

- Why generalized least-square?

Theorem (Aitken, 1935):

GLS yields smallest covariance amongst linear unbiased estimators. (Λ fixed!)

- Linear estimator: $$\hat{\beta} = L\tilde{y}, \quad L = \overbrace{\left(X^T \Lambda X\right)^{-1} X^T \Lambda}^{\text{GLS}} + D^T$$

- What about the strategic setting?

Theorem:

In the strategic setting, GLS gives optimal covariance amongst linear unbiased estimators. (Λ depends on the estimator!)

EURECOM Sophia Antipolis

Can we improve the estimation? [Chessa, Grossklags, L. FC '15, CSF '15]

- Case where the analyst only estimates the mean (d=1 and all x_i's are the same)

- Theorem: for a well chosen η, the analyst can strictly improve the estimator's variance by restricting the inverse variance chosen by the user to $\{0\} \cup [\eta, 1/\sigma^2]$

- Improves by a constant factor (PoS still increases the same with n)

Open questions

- General model
  - Linear regression with regularization
  - Recommendation
- Selection of agent to ask data from
- Combine monetary incentives with the users interest in the result

Is the iid assumption always valid?

- Security
  - Spam detection, detection of malicious behavior in online systems, malware detection, fraud detection
- Personal data
  - Privacy research: users obfuscating data before revealing it to an analyst, incentivizing high quality data, recommendations, reviews
- Data to learn from is generated or provided by humans
  - Strategic agents reacting to the learning algorithm
- How to learn in this situation?

Outline

1. Classification from strategic data
    a. The adversarial learning approach
    b. The game-theoretic approach
2. Linear regression from strategic data
    a. The game-theoretic approach

What's not covered here…

- Main focus of the tutorial: illustrate what game theory can bring on simple examples

- Non-covered topics:
  - Unsupervised learning
  - Sequential learning
    - Multi-armed bandits, prediction with expert advice

Outline

1. Classification from strategic data

   a. The adversarial learning approach

   b. The game-theoretic approach

2. Linear regression from strategic data

   a. The game-theoretic approach

Outline

1. Classification from strategic data
   a. The adversarial learning approach
   b. The game-theoretic approach

2. Linear regression from strategic data
   a. The game-theoretic approach

Outline

1. Classification from strategic data

    a. The adversarial learning approach

    b. The game-theoretic approach

2. Linear regression from strategic data

    a. The game-theoretic approach

Open problems

- Generalized model: how is the NE classifier affected
  - Generalized payoffs
  - Generalized action sets
  - Kernel based features
  - Regularization
  - Multi-class classification

- Dynamic classification
  - Learning the attacker's utility
  - Optimizing trade-off between acquiring vs using reputation

- Unsupervised learning
  - Clustering