Network Economics
--
Lecture 4: Incentives and games in security

Patrick Loiseau
EURECOM
Fall 2016

References

- J. Walrand. “Economics Models of Communication Networks”, in Performance Modeling and Engineering, Zhen Liu, Cathy H. Xia (Eds), Springer 2008. (Tutorial given at SIGMETRICS 2008).
  - Available online: http://robotics.eecs.berkeley.edu/~wlr/Papers/EconomicModels_Sigmetrics.pdf
- N. Nisam, T. Roughgarden, E. Tardos and V. Vazirani (Eds). “Algorithmic Game Theory”, CUP 2007. Chapter 17, 18, 19, etc.
  - Available online: http://www.cambridge.org/journals/nisan/downloads/Nisan_Non-printable.pdf

Outline

1. Interdependence: investment and free riding
2. Information asymmetry
3. Attacker *versus* defender games
   – Classification games

Outline

1. Interdependence: investment and free riding
2. Information asymmetry
3. Attacker *versus* defender games
   – Classification games

Incentive issues in security

- Plenty of security solutions…
  - Cryptographic tools
  - Key distribution mechanisms
  - etc.
- …useless if users do not install them
- Examples:
  - Software not patched
  - Private data not encrypted
- Actions of a user affects others! → game

A model of investment

- Jiang, Anantharam and Walrand, “How bad are selfish investments in network security”, IEEE/ACM ToN 2011
- Set of users N = {1, …, n}
- User i invests $x_i \geq 0$ in security
- Utility:

$$u_i(x) = u_0 - d_i(x) \quad \text{where} \quad d_i(x) = g_i\left(\sum_j \alpha_{ji} x_j\right) + x_i$$

- Assumptions:

Free-riding

- Positive externality → we expect free-riding
- Nash equilibrium x^{NE}
- Social optimum x^{SO}
- We look at the ratio: $$\rho = \frac{\sum_i d_i(x^{NE})}{\sum_i d_i(x^{SO})}$$
- Characterizes the ‘price of anarchy’

Remarks

- Interdependence of security investments
- Examples:
  - DoS attacks
  - Virus infection
- Asymmetry of investment importance
  - Simpler model in Varian, “System reliability and free riding”, in Economics of Information Security, 2004

Price of anarchy

- Theorem:

$$\rho \leq \max_j \left\{ 1 + \sum_{i \neq j} \beta_{ji} \right\} \quad \text{where} \quad \beta_{ji} = \frac{\alpha_{ji}}{\alpha_{ii}}$$

and the bound is tight

Comments

- There exist pure strategy NE
- $1+\sum_{i\neq j}\beta_{ji}=\sum_i \beta_{ji}$ is player j's importance to the society
- PoA bounded by the player having the most importance on society, regardless of $g_i(.)$

Examples

Bound tightness

Investment costs

- Modify the utility to

$$u_i(x) = u_0 - d_i(x) \quad \text{where} \quad d_i(x) = g_i\left(\sum_j \alpha_{ji} x_j\right) + c_i x_i$$

- The result becomes

$$\rho \le \max_j \left\{1 + \sum_{i \ne j} \beta_{ji}\right\} \quad \text{where} \quad \beta_{ji} = \frac{\alpha_{ji}}{\alpha_{ii}} \frac{c_i}{c_j}$$

Outline

1. Interdependence: investment and free riding
2. Information asymmetry
3. Attacker *versus* defender games
   – Classification games

Information asymmetry

- Hidden actions
  - See previous lecture
- Hidden information
  - Market for lemons
  - Example: software security

Market for lemons

- Akerlof, 1970
  - Nobel prize in 2001
- 100 car sellers
  - 50 have bad cars (lemons), willing to sell at $1k
  - 50 have good cars, willing to sell at $2k
  - Each knows its car quality
- 100 car buyers
  - Willing to buy bad cars for $1.2k
  - Willing to buy good cars for $2.4k
  - Cannot observe the car quality

Market for lemons (2)

- What happens? What is the clearing price?
- Buyer only knows average quality
  - Willing to pay \$1.8k
- But at that price, no good car seller sells
- Therefore, buyer knows he will buy a lemon
  - Pay max \$1.2k
- No good car is sold

Market for lemon (3)

- This is a market failure
  - Created by externalities: bad car sellers imposes an externality on good car sellers buy decreasing the average quality of cars on the market
- Software security:
  - Vendor can know the security
  - Buyers have no reason to trust them
    - So they won’t pay a premium
- Insurance for older people

Outline

1. Interdependence: investment and free riding
2. Information asymmetry
3. Attacker *versus* defender games
   – Classification games

Network security *[Symantec 2011]*

- Security threats increase due to technology evolution
  - Mobile devices, social networks, virtualization
- Cyberattacks is the first risk of businesses
  - 71% had at least one in the last year
- Top 3 losses due to cyberattacks
  - Downtime, employee identity theft, theft of intellectual property
- Losses are substantial
  - 20% of businesses lost > $195k

→Tendency to start using analytical models to optimize response to security threats

→Use of machine learning (classification)

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

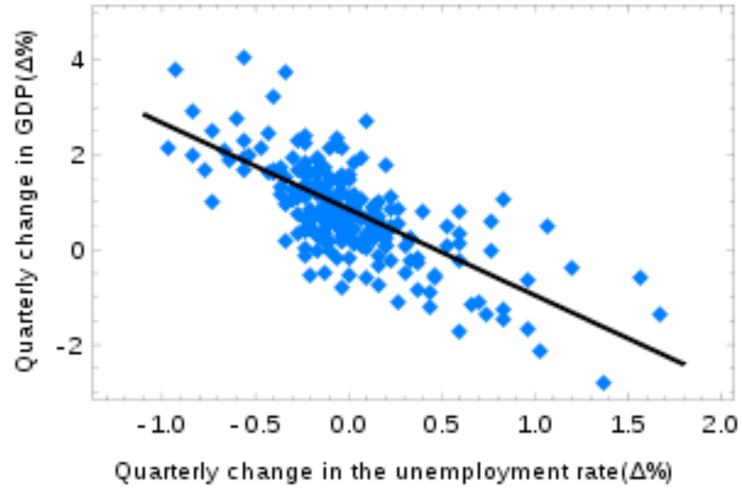
Supervised machine learning

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

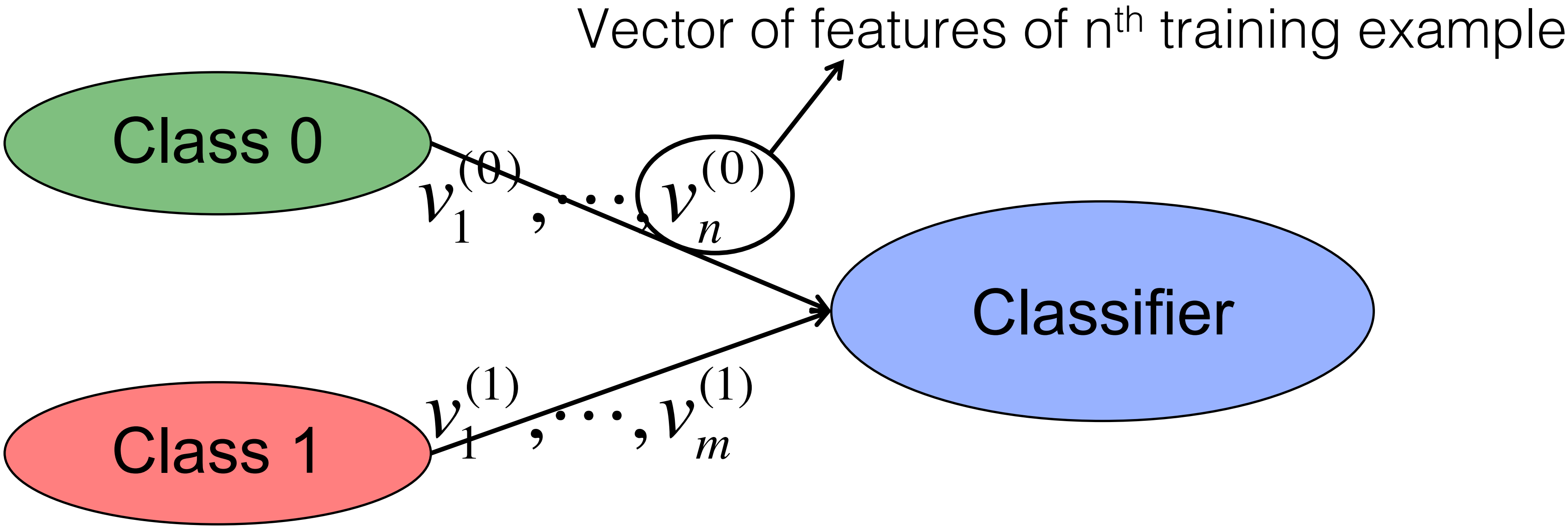
Content

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



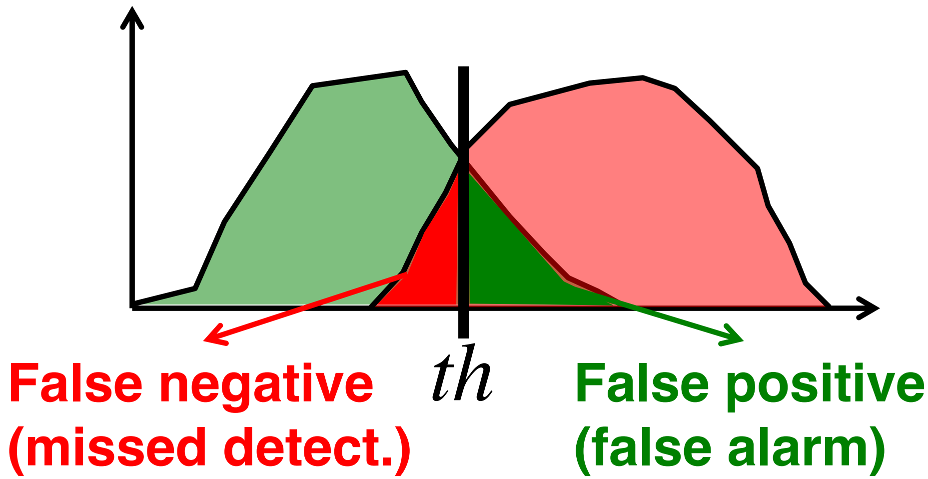
- Classifier's task
  - From $v_1^{(0)}, \cdots, v_n^{(0)}, v_1^{(1)}, \cdots, v_m^{(1)}$, make decision boundary
  - Classify new example v based on which side of the boundary

EURECOM Sophia Antipolis

Binary classification

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

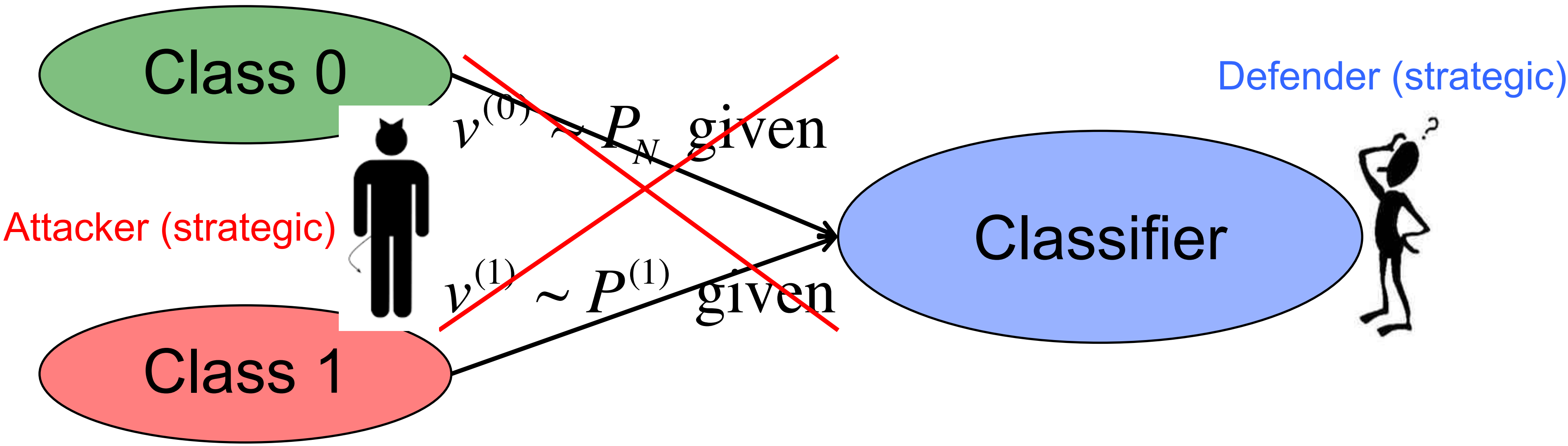
New example v: $\begin{matrix} \text{class } 0 \text{ if } v < th \\ \text{class } 1 \text{ if } v > th \end{matrix}$



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

EURECOM
Sophia Antipolis

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]

EURECOM
Sophia Antipolis

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

EURECOM
Sophia Antipolis

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"

EURECOM Sophia Antipolis

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

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

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

EURECOM Sophia Antipolis

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

- Payoffs:

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

- 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)

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

EURECOM Sophia Antipolis

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:

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

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 $T - T_S - T_Q$

EURECOM Sophia Antipolis

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

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

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

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

EURECOM Sophia Antipolis

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

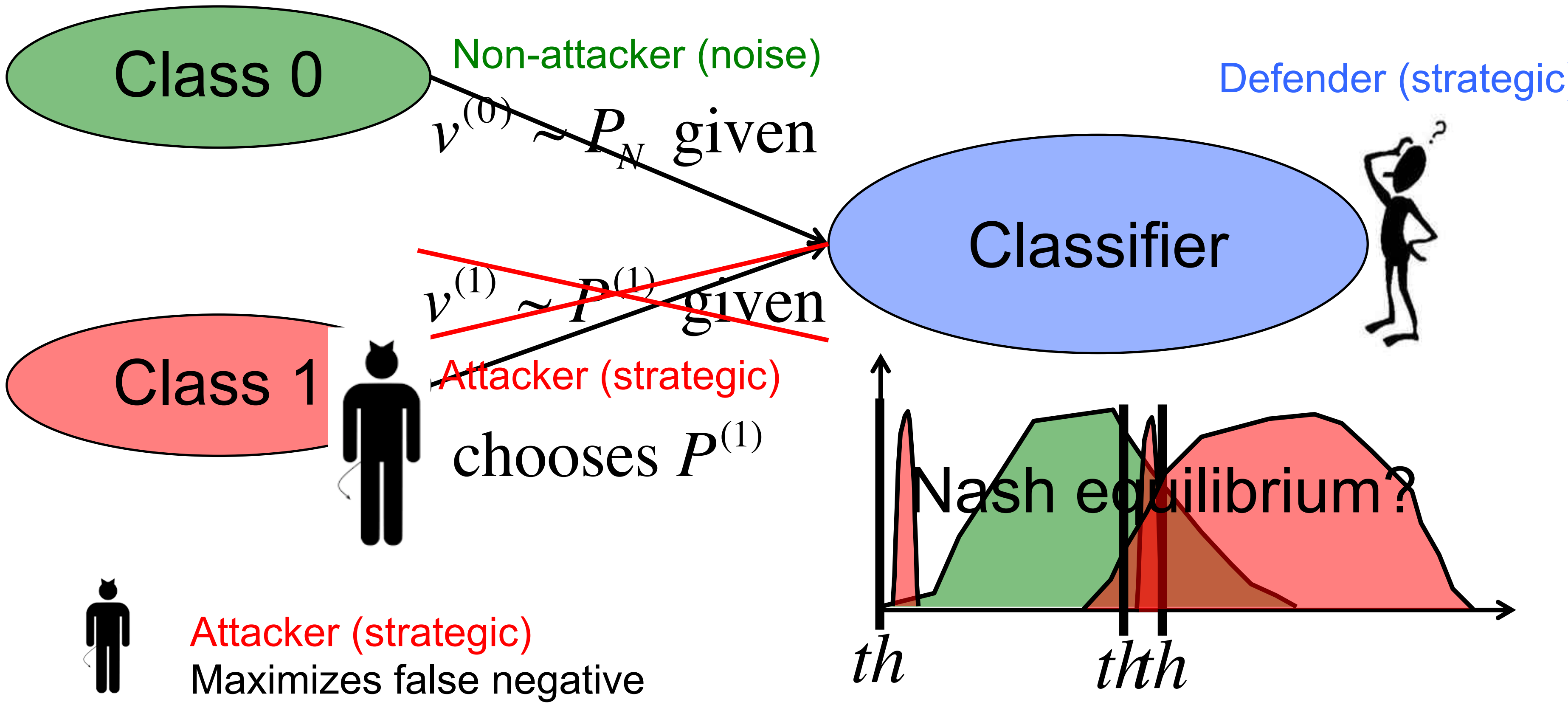
Content

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

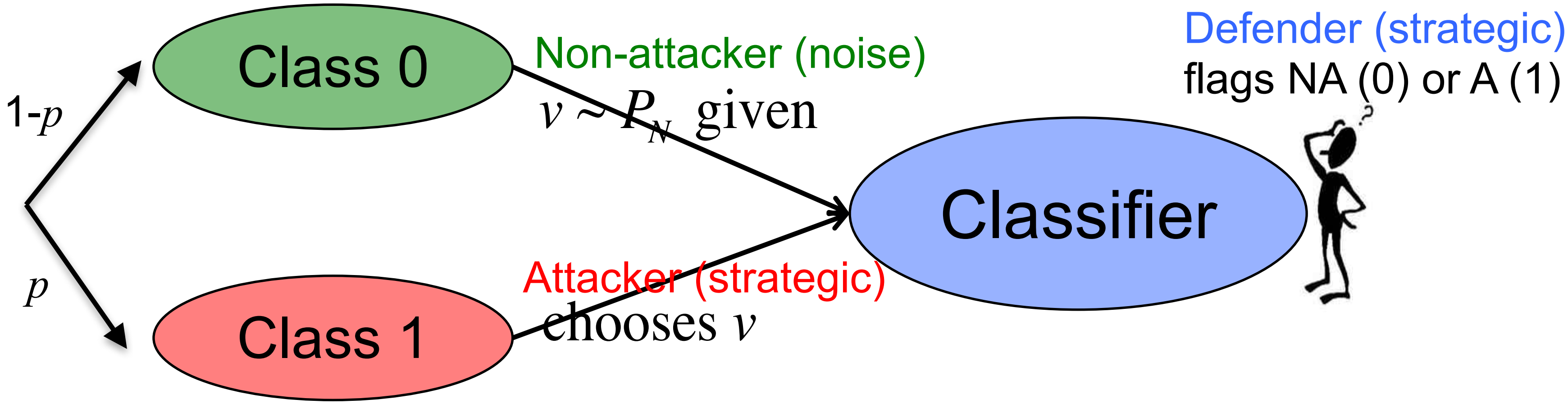
- Result:
  - Pure strategy Nash equilibrium

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 $G = \langle V, C, \overbrace{P_N, p, c_d, c_{fa}}^{\text{Payoff-relevant Parameters}} \rangle$

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:

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

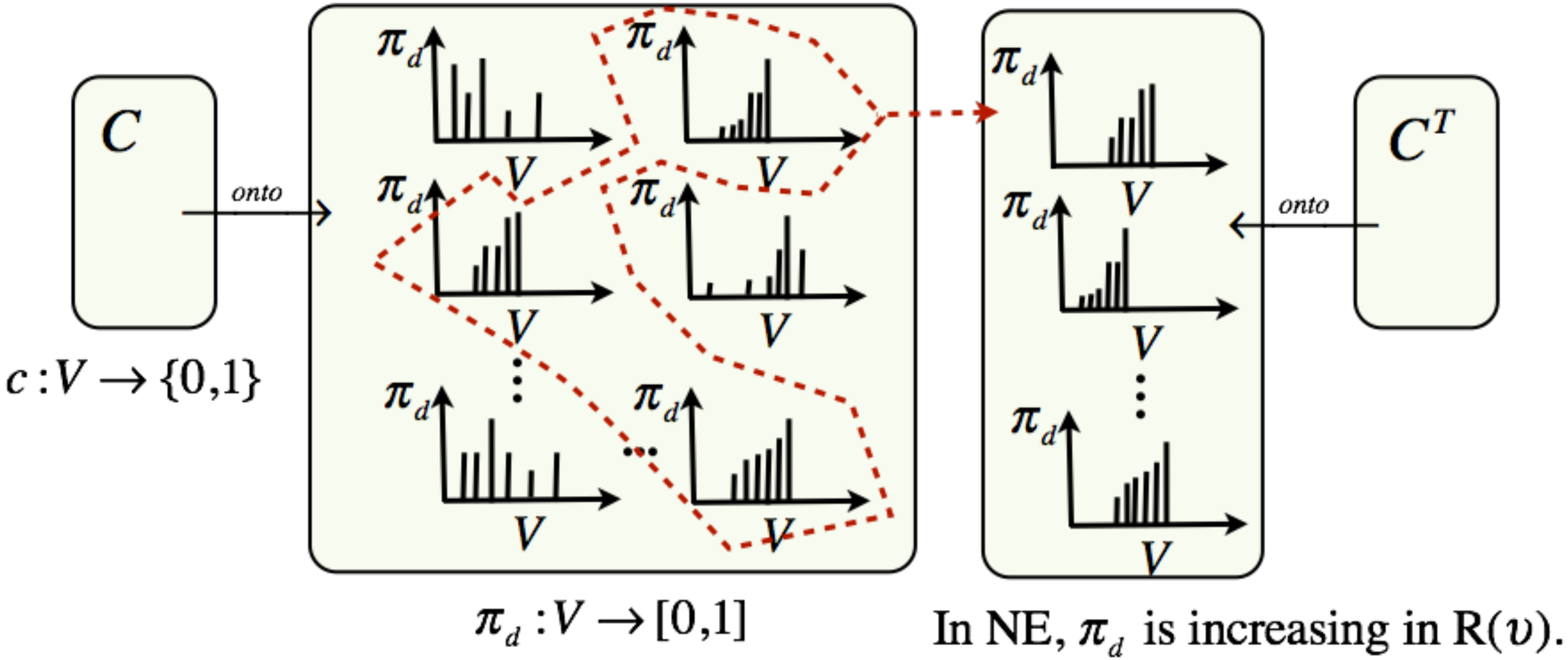
1. At NE, if $P_N(v) > 0$ for all v, then

   $\pi_d(v)$ increases with $R(v)$

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

EURECOM Sophia Antipolis

Main result 1: illustration



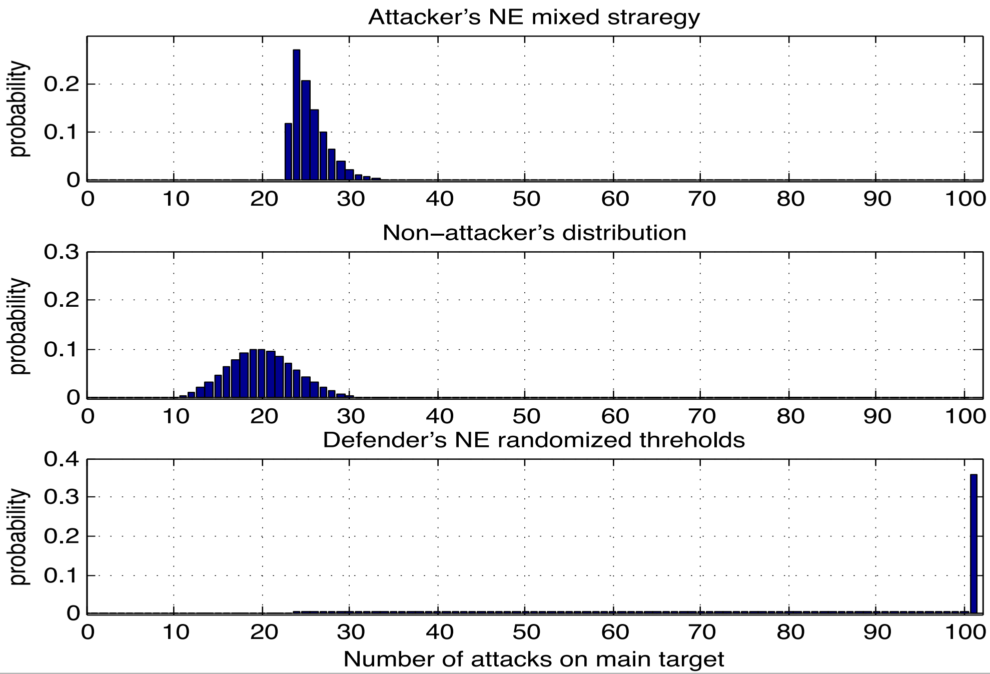
EURECOM Sophia Antipolis

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

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

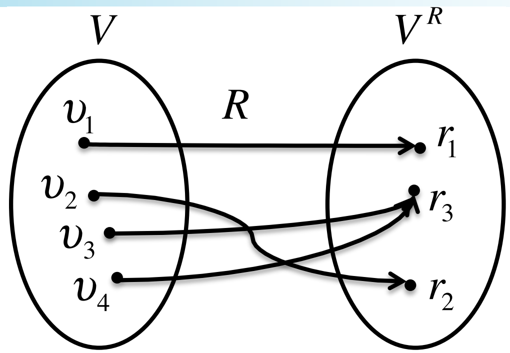


- Attacker's strategy: scaled version of the non-attacker distribution on a subset

EURECOM Sophia Antipolis

Reduction of attacker's strategy space

- V^R: set of rewards



Proposition:

If (α,β) is a NE of $G^T = \langle V, C^T, P_N, p, c_d, c_{fa} \rangle$ then (α',β) is a NE of $G^{R,T} = \langle V^R, C^T, P_N^R, p, c_d, c_{fa} \rangle$ with the same equilibrium payoffs, where $\alpha'_r = \sum_{v:R(v)=r} \alpha_v$.

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

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

Game rewriting in matrix form

$$|C^T| = |V^R| + 1$$

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

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

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

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

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

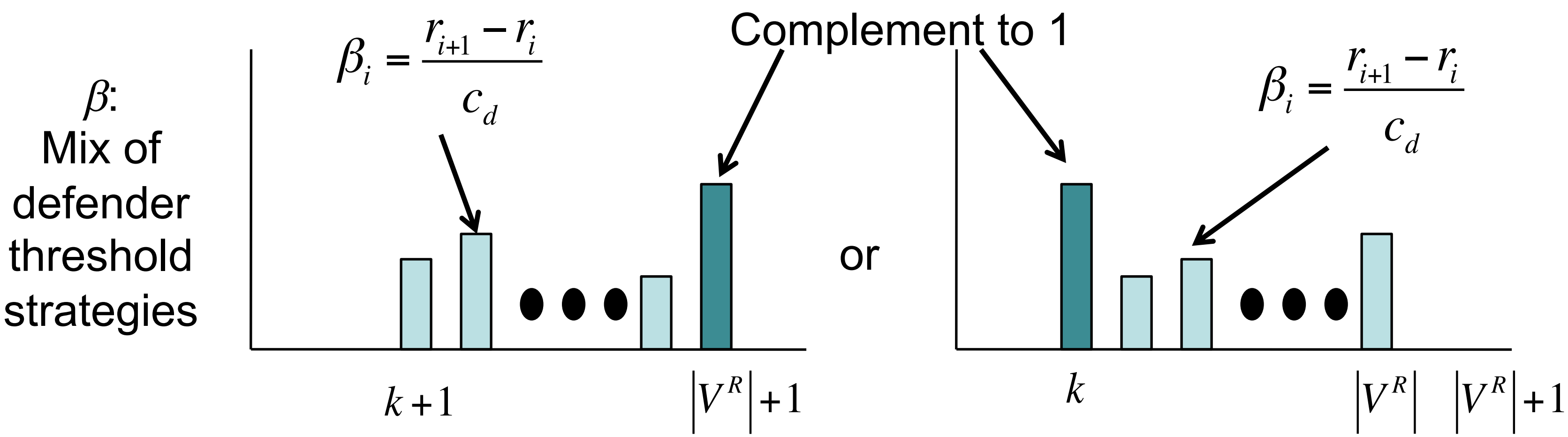
$$\beta_i = \frac{r_{i+1} - r_i}{c_d}, \text{ for } i \in \left\{k+1, \cdots, |V^R|\right\}$$

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

EURECOM
Sophia Antipolis

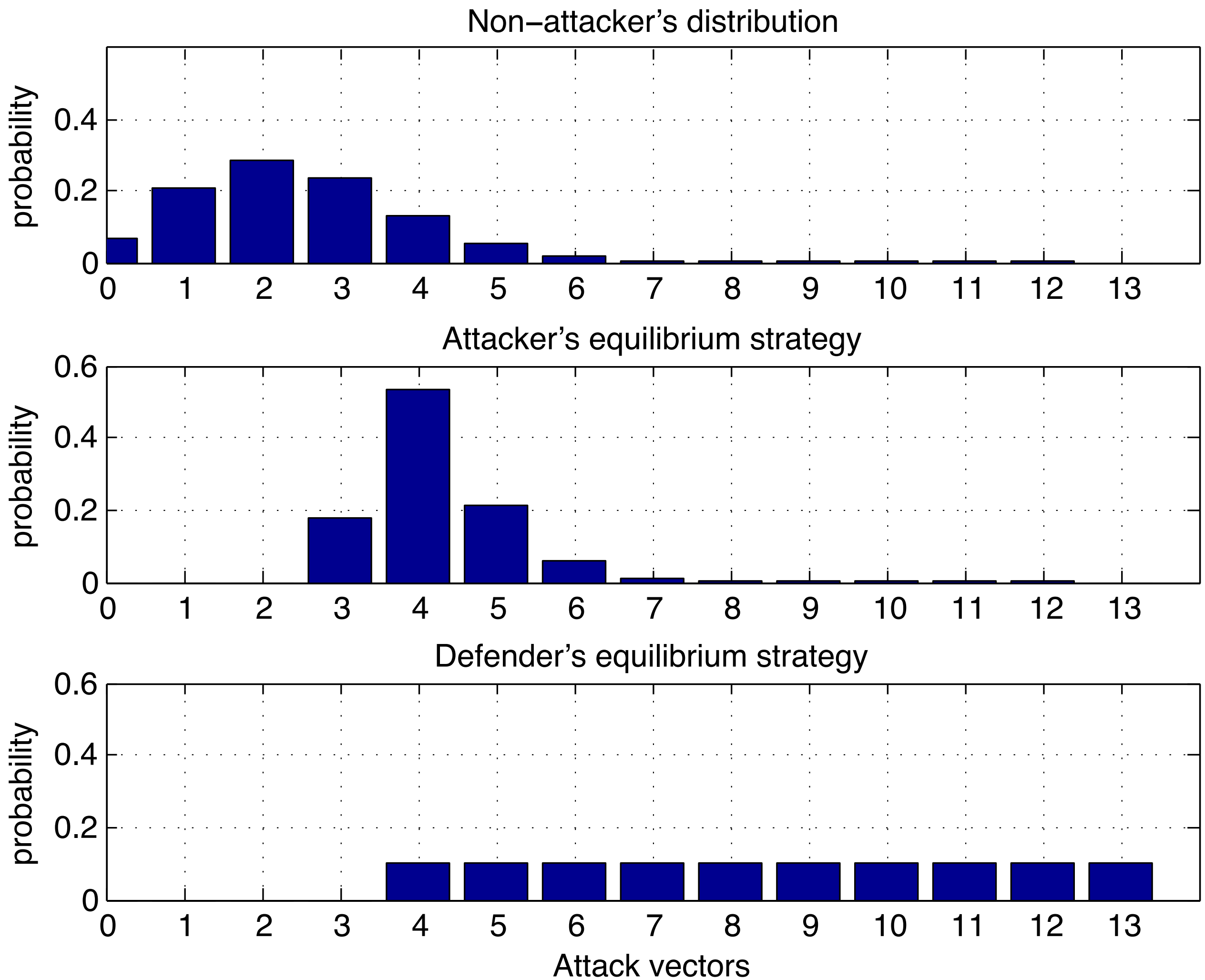
NE computation

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



- 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

$$r_i = i \cdot c_a$$

EURECOM Sophia Antipolis

Main result 3: proof's key steps

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

- Solve LP: maximize $z - \mu'\beta$

  s.t. $\Lambda\beta \geq z \cdot 1_{|V^R|}, \beta \geq 0, 1_{|V^R|+1} \cdot \beta = 1$

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

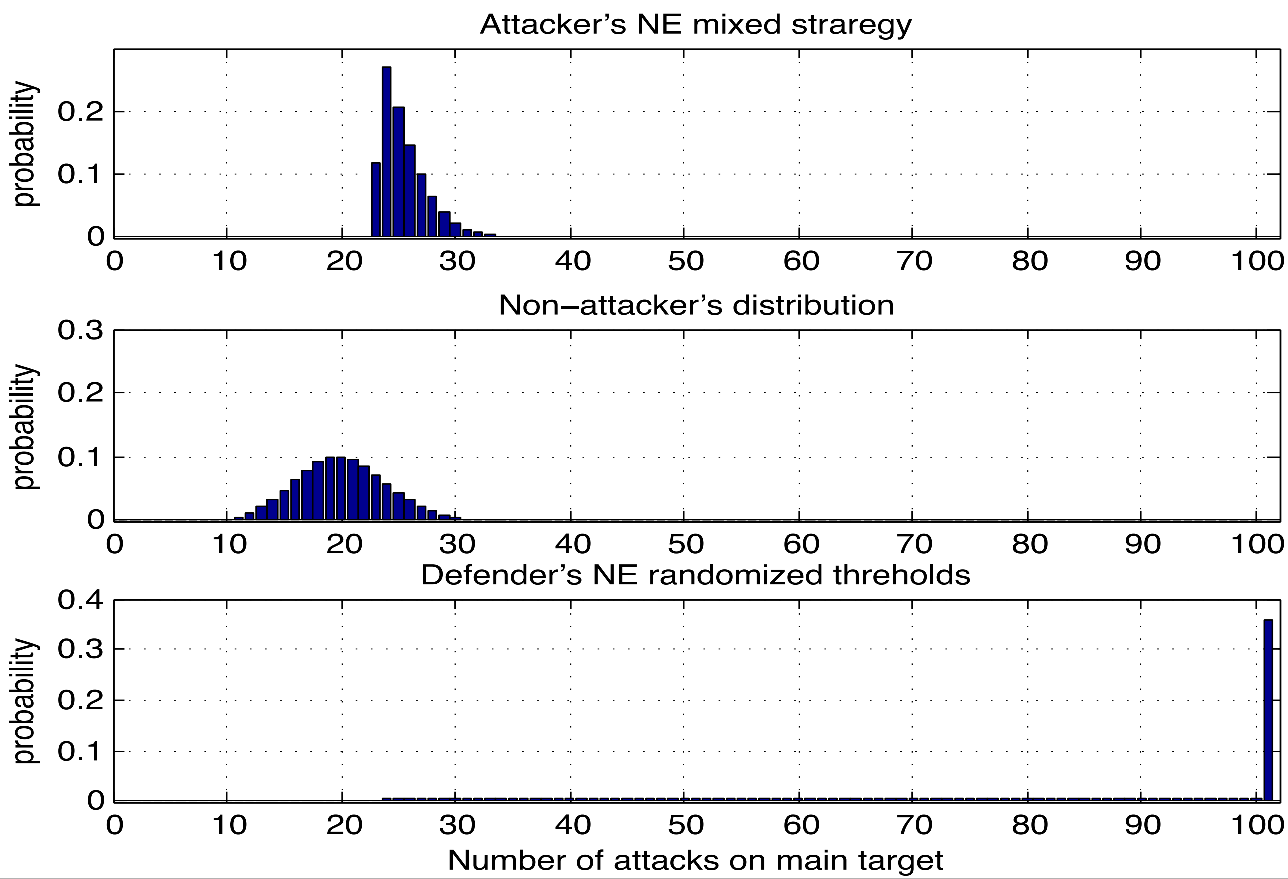
2. Look at polyhedron and eliminate points that are not extreme

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

EURECOM Sophia Antipolis

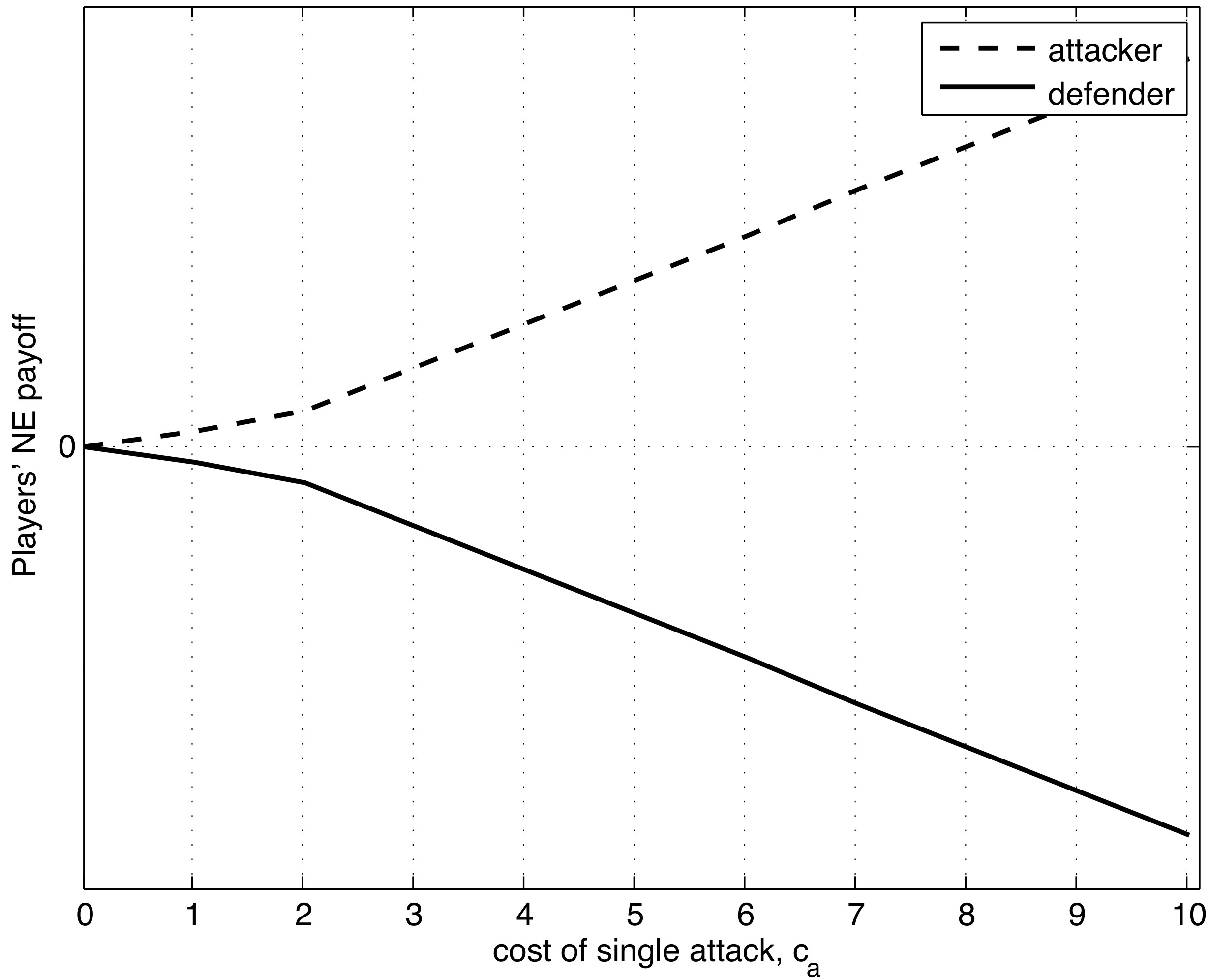
Example

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

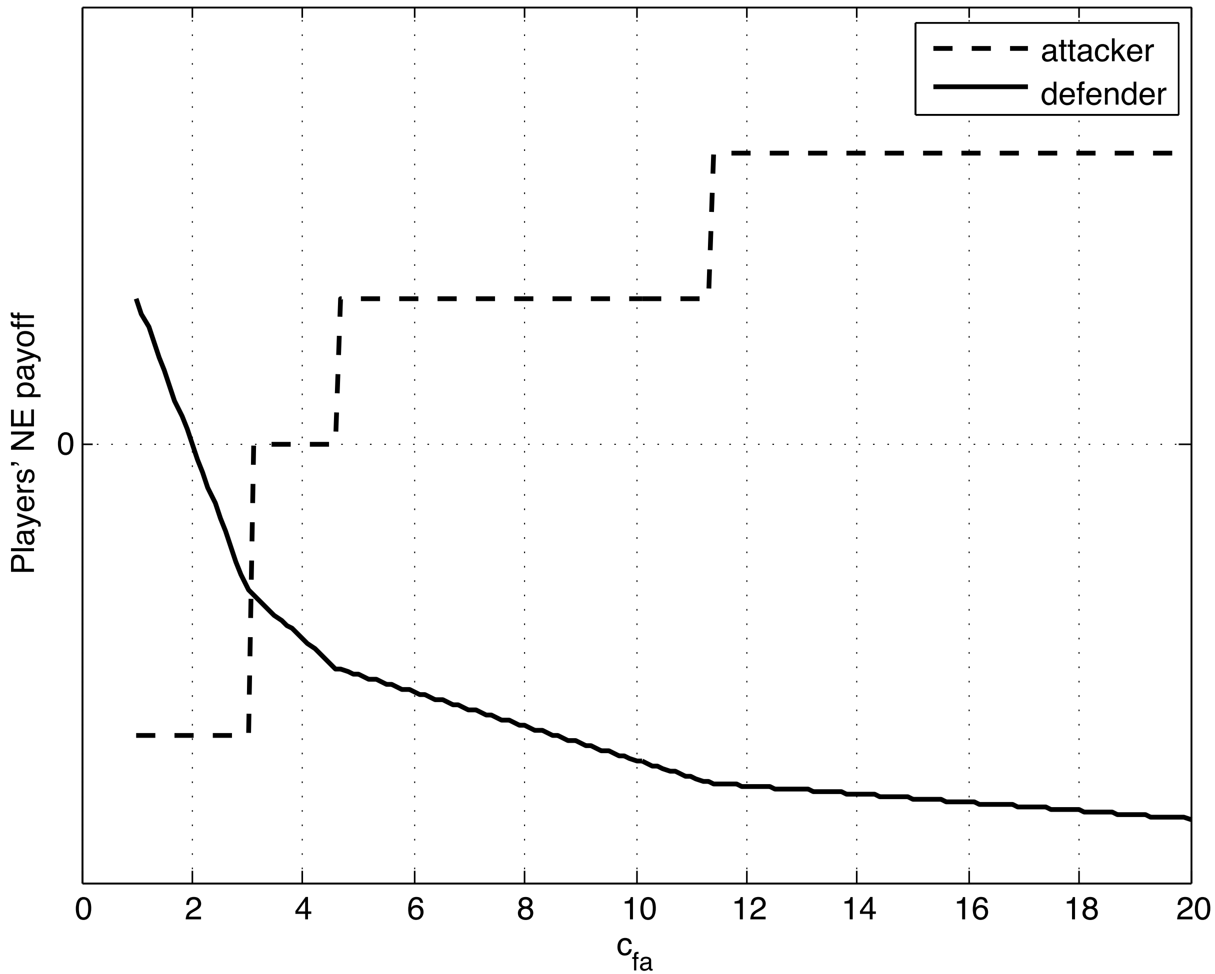


EURECOM
Sophia Antipolis

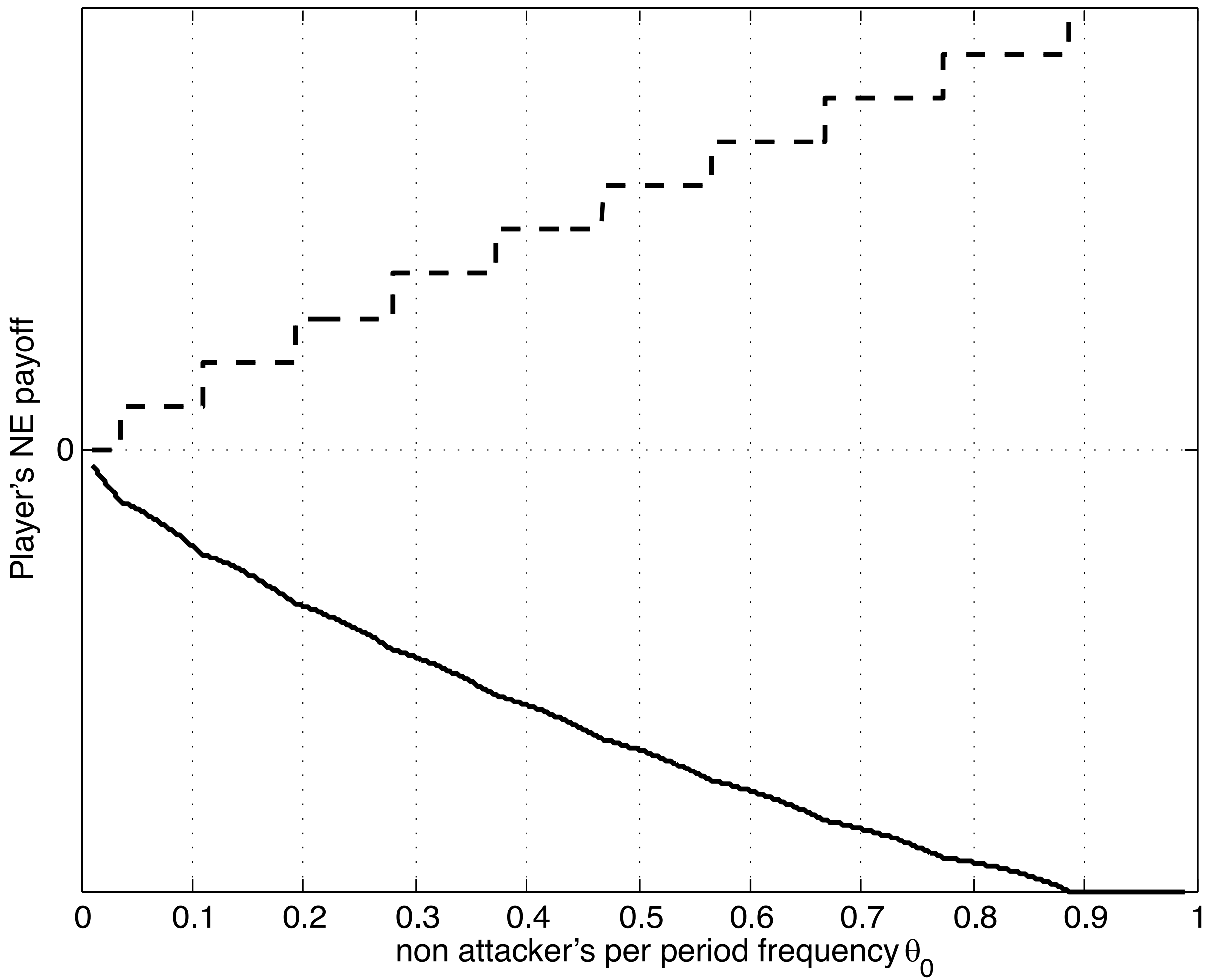
Example (2): variation with cost of attack



Example (3): variation with false alarm cost

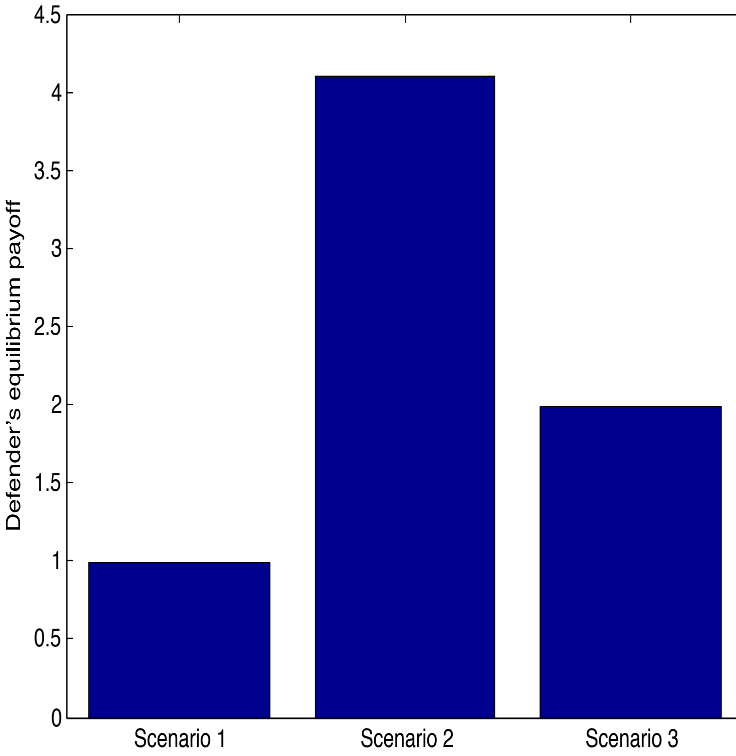


Example (4): Variation with noise strength



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

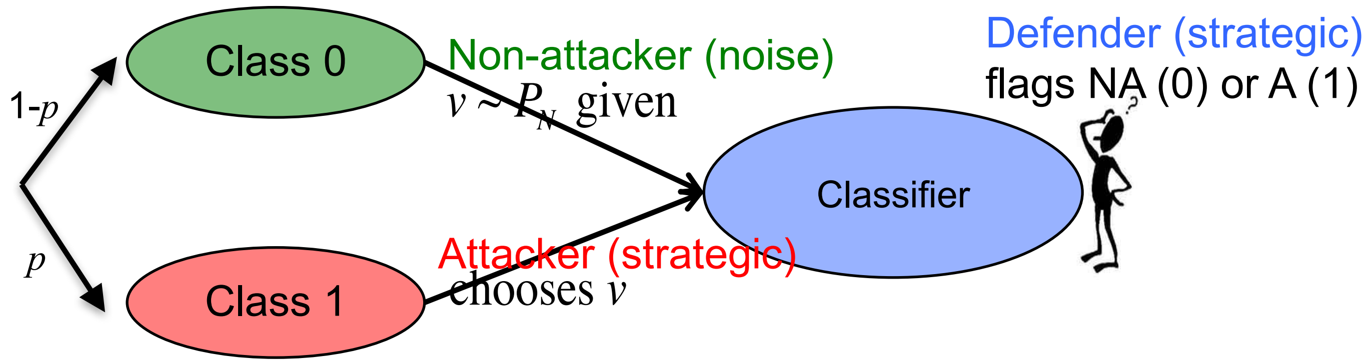
- There are two features
- 3 scenarios:
  - 1: defender classifies on feature 1 only
    - Attacker uses maximal strength on feature 2
  - 2: defender classifies on features 1 and 2 but attacker doesn't know
    - Attacker uses maximal strength on feature 2
  - 3: defender classifies on features 1 and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



EURECOM
Sophia Antipolis

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

EURECOM Sophia Antipolis

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