SAND2015-2534C

A Minimax Approach to Sensor Fusion for Intrusion Detection

Matthew Pugh

Sandia National Laboratories

SAS 2015

Table of Contents

Introduction

Goal:

- Try to analyze and mitigate the worst case performance of the intrusion detection system

Introduction

Goal:

- Try to analyze and mitigate the worst case performance of the intrusion detection system

Framework:

- Assume we know the the statistical distribution of the background signal
  - Using results derived from other paper
  - **Sensor Fusion**: Combine all sensors into a single metric - Mahalanobis distance
  - Background signal is chi-squared distributed

Introduction

Goal:

- Try to analyze and mitigate the worst case performance of the intrusion detection system

Framework:

- Assume we know the the statistical distribution of the background signal
  - Using results derived from other paper
  - **Sensor Fusion**: Combine all sensors into a single metric - Mahalanobis distance
  - Background signal is chi-squared distributed
- Compute the worst-case event distribution
  - Assumes a cost associated with making a decision

Introduction

Goal:

- Try to analyze and mitigate the worst case performance of the intrusion detection system

Framework:

- Assume we know the the statistical distribution of the background signal
  - Using results derived from other paper
  - **Sensor Fusion**: Combine all sensors into a single metric - Mahalanobis distance
  - Background signal is chi-squared distributed
- Compute the worst-case event distribution
  - Assumes a cost associated with making a decision

A Different Perspective:

- False alarm constraints versus worst-case performance

Classic Example: Rock, Paper, Scissors

Alice and Bob play rock, paper, scissors

Payoff Matrix

Alice \Bob	Rock	Paper	Scissors
Rock	0	-1	1
Paper	1	0	-1
Scissors	-1	1	0

Classic Example: Rock, Paper, Scissors

Alice and Bob play rock, paper, scissors

Payoff Matrix

Alice \Bob	Rock	Paper	Scissors
Rock	0	-1	1
Paper	1	0	-1
Scissors	-1	1	0

Question: How should Alice and Bob play?

Classic Example: Rock, Paper, Scissors

Alice and Bob play rock, paper, scissors

Payoff Matrix

Alice \Bob	Rock	Paper	Scissors
Rock	0	-1	1
Paper	1	0	-1
Scissors	-1	1	0

Question: How should Alice and Bob play?

- Mixed strategies!
- Choose randomly according to some distribution
- Alice chooses according to $\mathbf{x}$ and Bob chooses according to $\mathbf{y}$

Classic Example: Rock, Paper, Scissors

Alice and Bob play rock, paper, scissors

Payoff Matrix

Alice \Bob	Rock	Paper	Scissors
Rock	0	-1	1
Paper	1	0	-1
Scissors	-1	1	0

Notation:
$\mathbf{x} = [\Pr[\text{Alice} = \text{Rock}], \Pr[\text{Alice} = \text{Paper}], \Pr[\text{Alice} = \text{Scissors}]]^T \in \mathbb{R}^3$
$\mathbf{y} = [\Pr[\text{Bob} = \text{Rock}], \Pr[\text{Bob} = \text{Paper}], \Pr[\text{Bob} = \text{Scissors}]]^T \in \mathbb{R}^3$
Payoff matrix: $\mathbf{M} \in \mathbb{R}^{3\times 3}$

$$\boxed{\textbf{Expected Payoff} = \mathbf{x}^T \mathbf{M} \mathbf{y}}$$

Classic Example: Rock, Paper, Scissors

Alice and Bob play rock, paper, scissors

Payoff Matrix

Alice \Bob	Rock	Paper	Scissors
Rock	0	-1	1
Paper	1	0	-1
Scissors	-1	1	0

Define $\beta(\mathbf{x}) = \min_{\mathbf{y}} \mathbf{x}^T \mathbf{M} \mathbf{y}$ and $\alpha(\mathbf{y}) = \max_{\mathbf{x}} \mathbf{x}^T \mathbf{M} \mathbf{y}$

Mixed Nash Equilibrium: A pair $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ such that

$$\beta\left(\tilde{\mathbf{x}}\right) = \tilde{\mathbf{x}}^T \mathbf{M} \tilde{\mathbf{y}} = \alpha\left(\tilde{\mathbf{y}}\right)$$

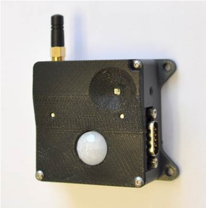
Test Bed

Sensor Module

- Tri-axis accelerometer
- Photo-detector
- Passive infrared sensor

Instrumented Room

- Placed 8 sensor modules along walls
- Modules connected via CAN bus



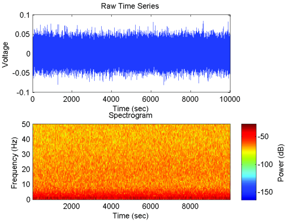
Objective

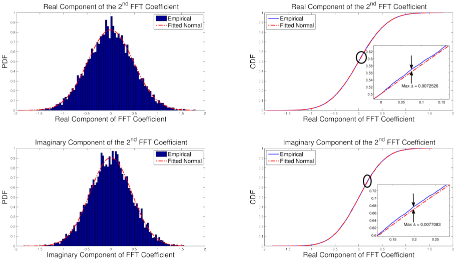
- Collect background data
- Collected data during entry
- Develop decision algorithm to minimize worst-case cost
    - Can handle arbitrary number of possible decisions

Previous Results

Goal: Find distribution on background data

- Analyze distribution of frequency components

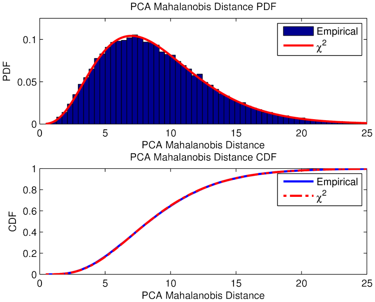


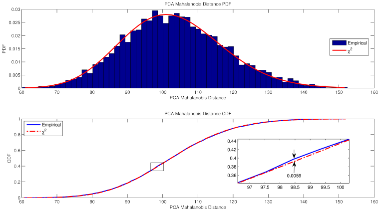


- Marginal Distributions: real and imaginary frequency components look Gaussian

Previous Results

PCA and Mahalanobis Distance



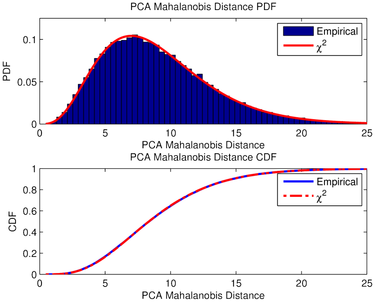


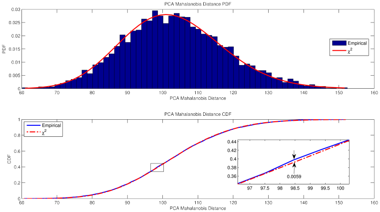
Metric with a known distribution

- Chi-squared distribution for Mahalanobis distance

Previous Results

PCA and Mahalanobis Distance





Metric with a known distribution

- Chi-squared distribution for Mahalanobis distance

Questions:

- If an adversary chose the event distribution, what would it look like?
- How could we design our algorithm to minimize the adverse effects?

The Problem

- How does the Mahalanobis distance distribution connect with *rock, paper, scissors*?

The Problem

- How does the Mahalanobis distance distribution connect with *rock, paper, scissors*?
  - In *rock,paper, scissors*, Bob tries to minimize payoff given a fixed distribution for Alice: $\beta(\mathbf{x}) = \min_{\mathbf{y}} \mathbf{x}^T \mathbf{M} \mathbf{y}$

The Problem

- How does the Mahalanobis distance distribution connect with *rock, paper, scissors*?
  - In *rock,paper, scissors*, Bob tries to minimize payoff given a fixed distribution for Alice: $\beta(\mathbf{x}) = \min_{\mathbf{y}} \mathbf{x}^T \mathbf{M} \mathbf{y}$
  - In our problem, we assume that the Mahalanobis distance distribution is fixed
  - Bob can choose a distribution $\mathbf{y}$ to minimize our payoff
    - We must define our payoff

The Problem

- How does the Mahalanobis distance distribution connect with *rock, paper, scissors*?
  - In *rock,paper, scissors*, Bob tries to minimize payoff given a fixed distribution for Alice: $\beta(\mathbf{x}) = \min_{\mathbf{y}} \mathbf{x}^T \mathbf{M} \mathbf{y}$
  - In our problem, we assume that the Mahalanobis distance distribution is fixed
  - Bob can choose a distribution $\mathbf{y}$ to minimize our payoff
    - We must define our payoff
  - Our recourse: Alice can modify the decision algorithm
    - For a given observed Mahalanobis distance value, Alice can optimize what decision is made to maximize payoff

The Problem

Problem:

- Every T seconds, we observe the Mahalanobis distance X computed from all of the sensors
  - Sensor fusion is in the metric

The Problem

Problem:

- Every T seconds, we observe the Mahalanobis distance X computed from all of the sensors
  - Sensor fusion is in the metric
- X is either generated from background noise or an event

The Problem

Problem:

- Every T seconds, we observe the Mahalanobis distance X computed from all of the sensors
  - Sensor fusion is in the metric
- X is either generated from background noise or an event
- **Task**: Determine what generated X
- **Goal**: Bound worst-case performance

The Problem

Problem:

- Every T seconds, we observe the Mahalanobis distance X computed from all of the sensors
  - Sensor fusion is in the metric
- X is either generated from background noise or an event
- **Task**: Determine what generated X
- **Goal**: Bound worst-case performance
- **Minimax approach**:
  - Find worst-case event distribution
  - Determine best decision to minimize cost
    - Cost needs to be defined
    - Cost can be subjective

Toy Example #1: Picking a Distribution

Binary Decision Problem: Samples are drawn from one of two possible distributions - decide from which one

- Background data $\sim U[0,1] = \mathbf{p}_{bg}$
- Event data $\sim$ Bob's choice $= \mathbf{p}_{event}$

Toy Example #1: Picking a Distribution

Binary Decision Problem: Samples are drawn from one of two possible distributions - decide from which one

- Background data $\sim U[0,1] = \mathbf{p}_{bg}$
- Event data $\sim$ Bob's choice $= \mathbf{p}_{event}$

Notation:

Decision Matrix: $T \in \mathbb{R}^{2\times N}$ where $T_{i,j} = \Pr[\alpha_i | X = x_k]$

Toy Example #1: Picking a Distribution

Binary Decision Problem: Samples are drawn from one of two possible distributions - decide from which one

- Background data $\sim U[0,1] = \mathbf{p}_{bg}$
- Event data $\sim$ Bob's choice $= \mathbf{p}_{event}$

Notation:

Decision Matrix: $T \in \mathbb{R}^{2\times N}$ where $T_{i,j} = \Pr[\alpha_i | X = x_k]$

- Note: 2 is the number of actions, N is the number of possible observations, α_i is the i^{th} decision, x_k is the k^{th} possible observed value
- Implication: For continuous distributions, **discretization is required**

Toy Example #1: Picking a Distribution

Binary Decision Problem: Samples are drawn from one of two possible distributions - decide from which one

- Background data $\sim U[0,1] = \mathbf{p}_{bg}$
- Event data $\sim$ Bob's choice $= \mathbf{p}_{event}$

Notation:
Decision Matrix: $T \in \mathbb{R}^{2\times N}$ where $T_{i,j} = \Pr[\alpha_i | X = x_k]$
Probability Matrix: $P \in \mathbb{R}^{N\times 2}$ where $P_{k,j} = \Pr[X = x_k | \omega_j]$

Toy Example #1: Picking a Distribution

Binary Decision Problem: Samples are drawn from one of two possible distributions - decide from which one

- Background data $\sim U[0,1] = \mathbf{p}_{bg}$
- Event data $\sim$ Bob's choice $= \mathbf{p}_{event}$

Notation:

Decision Matrix: $T \in \mathbb{R}^{2\times N}$ where $T_{i,j} = \Pr[\alpha_i | X = x_k]$
Probability Matrix: $P \in \mathbb{R}^{N\times 2}$ where $P_{k,j} = \Pr[X = x_k | \omega_j]$

- Note: 2 is the number of states of nature: **background** or **event**, ω_j is the j^{th} state of nature
- First column: $\mathbf{p}_{bg}$, second column: $\mathbf{p}_{event}$

Toy Example #1: Picking a Distribution

Binary Decision Problem: Samples are drawn from one of two possible distributions - decide from which one

- Background data $\sim U[0,1] = \mathbf{p}_{bg}$
- Event data $\sim$ Bob's choice $= \mathbf{p}_{event}$

Notation:
Decision Matrix: $T \in \mathbb{R}^{2\times N}$ where $T_{i,j} = \Pr[\alpha_i | X = x_k]$
Probability Matrix: $P \in \mathbb{R}^{N\times 2}$ where $P_{k,j} = \Pr[X = x_k | \omega_j]$
Loss Matrix: $\Lambda \in \mathbb{R}^{2\times 2}$ where $\Lambda_{i,j} = \lambda\left(\alpha_i | \omega_j\right)$

Toy Example #1: Picking a Distribution

Binary Decision Problem: Samples are drawn from one of two possible distributions - decide from which one

- Background data $\sim U[0,1] = \mathbf{p}_{bg}$
- Event data $\sim$ Bob's choice $= \mathbf{p}_{event}$

Notation:

Decision Matrix: $T \in \mathbb{R}^{2\times N}$ where $T_{i,j} = \Pr[\alpha_i | X = x_k]$
Probability Matrix: $P \in \mathbb{R}^{N\times 2}$ where $P_{k,j} = \Pr[X = x_k | \omega_j]$
Loss Matrix: $\Lambda \in \mathbb{R}^{2\times 2}$ where $\Lambda_{i,j} = \lambda(\alpha_i | \omega_j)$

- Λ has dimensions # of actions by # of states of nature
- The loss values can be **subjective**!

Toy Example #1: Picking a Distribution

Binary Decision Problem: Samples are drawn from one of two possible distributions - decide from which one

- Background data $\sim U[0,1] = \mathbf{p}_{bg}$
- Event data $\sim$ Bob's choice $= \mathbf{p}_{event}$

Notation:

Decision Matrix: $T \in \mathbb{R}^{2\times N}$ where $T_{i,j} = \Pr[\alpha_i | X = x_k]$
Probability Matrix: $P \in \mathbb{R}^{N\times 2}$ where $P_{k,j} = \Pr[X = x_k | \omega_j]$
Loss Matrix: $\Lambda \in \mathbb{R}^{2\times 2}$ where $\Lambda_{i,j} = \lambda\left(\alpha_i | \omega_j\right)$
Prior probabilities on state of nature: $p(\omega)$

Toy Example #1: Picking a Distribution

Binary Decision Problem: Samples are drawn from one of two possible distributions - decide from which one

- Background data $\sim U[0,1] = \mathbf{p}_{bg}$
- Event data $\sim$ Bob's choice $= \mathbf{p}_{event}$

Notation:

Decision Matrix: $T \in \mathbb{R}^{2\times N}$ where $T_{i,j} = \Pr[\alpha_i | X = x_k]$
Probability Matrix: $P \in \mathbb{R}^{N\times 2}$ where $P_{k,j} = \Pr[X = x_k | \omega_j]$
Loss Matrix: $\Lambda \in \mathbb{R}^{2\times 2}$ where $\Lambda_{i,j} = \lambda\left(\alpha_i | \omega_j\right)$
Prior probabilities on state of nature: $p(\omega)$

Question: Given the loss matrix Λ, background distribution $\mathbf{p}_{bg}$ and the prior probabilities $p(\omega)$:

- How would Bob select $\mathbf{p}_{event}$ to *maximize* loss?
- How would Alice design T to *minimize* loss?

Toy Example #1: Optimization Problem

Define the conditional risk as:

$$R\left(\alpha_i|x\right) = \sum_j \lambda\left(\alpha_i|\omega_j\right) p\left(\omega_j|x\right) = \sum_j \lambda\left(\alpha_i|\omega_j\right) \frac{p(x|\omega_j)p(\omega_j)}{p(x)}$$

Want to minimize risk: $\alpha(x) = \underset{\alpha_i}{\operatorname{argmin}} R\left(\alpha_i|x\right)$

Define the *risk* as:

$$R = \sum_i^N R\left(\alpha(x_i)|x_i\right) p(x_i) = \mathbf{1}^T \left(\left(\Lambda \cdot \mathsf{diag}(p)\right) \circ (TP)\right) \mathbf{1}$$

Toy Example #1: Optimization Problem

The minimax problem is

$$\begin{aligned} \min_{T \in \mathbb{R}^{p \times N}} \max_{\mathbf{p} \in \mathbb{R}^N} \quad & \mathbf{1}^T \left((\Lambda \cdot \mathsf{diag}(p)) \circ (TP) \right) \mathbf{1} \\ \text{subject to} \quad & \mathbf{p}^T \mathbf{1} = 1 \\ & \mathbf{p} \geq 0 \\ & T \geq 0 \\ & \mathbf{1}^T T = \mathbf{1}^T \\ & \mathbf{p}^T \mathbf{x} = \mu_{\text{event}} \end{aligned}$$

Constraints:

- Mean constraint
- Probability constraints
- Can add linear constraints e.g. moments

Toy Example #1: Optimization Problem

The minimax problem is

$$\begin{aligned} \min_{T \in \mathbb{R}^{p \times N}} \max_{\mathbf{p} \in \mathbb{R}^N} \quad & \mathbf{1}^T \left((\Lambda \cdot \text{diag}(p)) \circ (TP) \right) \mathbf{1} \\ \text{subject to} \quad & \mathbf{p}^T \mathbf{1} = 1 \\ & \mathbf{p} \geq 0 \\ & T \geq 0 \\ & \mathbf{1}^T T = \mathbf{1}^T \\ & \mathbf{p}^T \mathbf{x} = \mu_{\text{event}} \end{aligned}$$

Minimax Solution: There exists a unique answer to the problem!

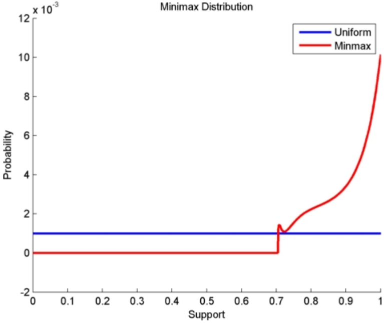
- Problem must be recast using linear programming duality to be put into convex optimization packages
- Solution seems to be sensitive to discretization and solver

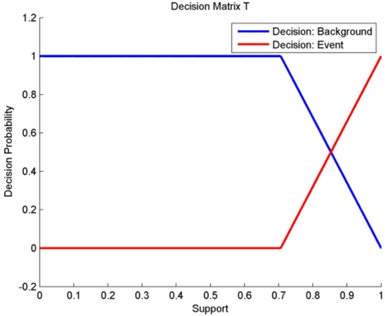
Toy Example #1: Results

Parameters:

- $\mathbf{p}_{bg} \sim U[0,1])$
- $[0,1]$ uniformly discretized into 1000 bins
- $\mu_{event} = 0.9$
- $p(\text{event}) = 0.1 = 1 - p(\text{background})$
- $\Lambda = \begin{bmatrix} -500 & 1000 \\ 15 & -1000 \end{bmatrix}$

Toy Example #1: Results





- Small probabilities due to discretization
- Randomized Decisions

Toy Example #2: Ternary Decision Problem

Problem:

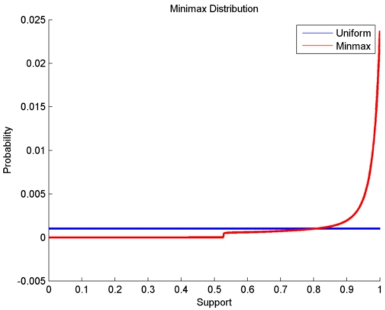
- Samples are drawn from two possible distributions
  - Background data $\sim U[0,1] = \mathbf{p}_{bg}$
  - Event data $\sim$ Bob's choice $= \mathbf{p}_{event}$
- Allow a third decision option: uncertain
- Task: Decide which distribution sample is drawn from or declare uncertainty
  - Can be extended to arbitrary number of decisions

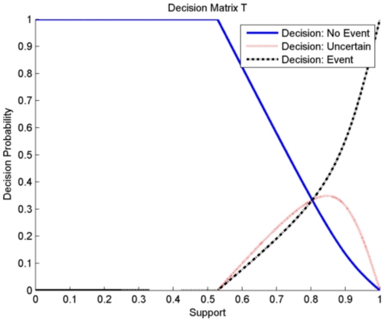
Toy Example #2: Ternary Decision Problem

Parameters:

- $\mathbf{p}_{bg} \sim U[0,1])$
- $[0,1]$ uniformly discretized into 1000 bins
- $\mu_{event} = 0.9$
- $p(\text{event}) = 0.1 = 1 - p(\text{background})$
- $\Lambda = \begin{bmatrix} -100 & 1000 \\ 50 & -500 \\ 100 & -1000 \end{bmatrix}$
  - Columns: $\{\text{background}, \text{event}\}$
  - Rows: $\{\text{background}, \text{uncertain}, \text{event}\}$

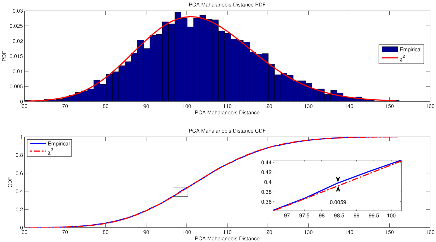
Toy Example #2: Ternary Decision Problem





Minimax Sensor Fusion: Analogy

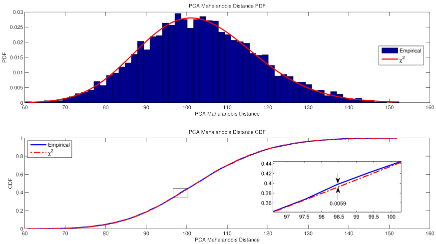
Background Distribution



- Chi-squared distribution for Mahalanobis distance
- Mahalanobis distance incorporates data from all PIR sensors

Minimax Sensor Fusion: Analogy

Background Distribution



The same problem as the toy examples:

- Observable (Mahalanobis distance) drawn from two possible distributions
  - Background Distribution $\sim \chi^2$
  - Event Distribution
- How to choose which distribution the observed Mahalanobis distance came from?

Minimax Sensor Fusion: Parameters

Discretization:

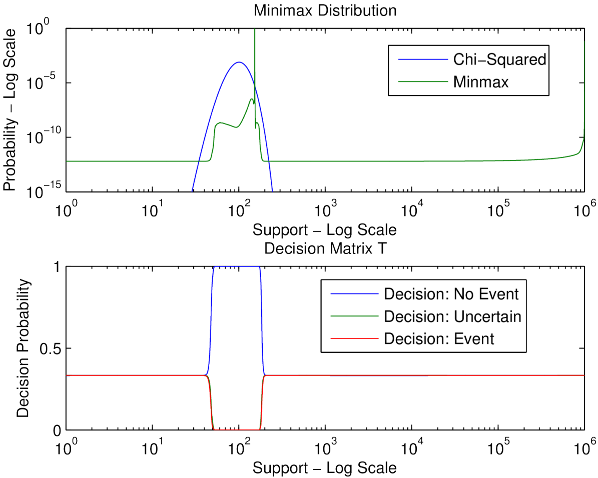
- Observables occur over massive scales
  - Average background: 101
  - Maximum event: 4.2×10^5
- How to discretization support?
  - Optimization sensitive to support
  - Feasibility - cannot have *too* many points
- Our approach:
  - Uniformly logarithmically spaced between 0 and $\left\lceil \log_{10} 4.2 \times 10^5 \right\rceil$ with 50000 points
  - $\Pr[x_i] = F_{\chi^2}(x_i) - F_{\chi^2}(x_{i-1})$

Minimax Sensor Fusion: Parameters

Parameters:

- $\mu_{event} = 6.674 \times 10^4 =$ Empirical mean on test data
- p(event) $= 1 \times 10^{-7}$
- Hypotheses: { No Event, Event }
- Actions: { No Event, Uncertain, Event }
- $\Lambda = \begin{bmatrix} -100 & 1000 \\ 50 & -500 \\ 100 & -1000 \end{bmatrix}$
    - Columns: Hypotheses
    - Rows: Actions
    - How to select these values?

Minimax Sensor Fusion: Results



Conclusion

Bound on performance

Conclusion

Bound on performance

- Minimax solution finds worst-case event distribution

Conclusion

Bound on performance

- Minimax solution finds worst-case event distribution
- Leveraged past work to define:
  - Observable metric - Mahalanobis distance
  - Distribution on observable - χ^2 distribution
  - Metric combines information from multiple sensors

Conclusion

Bound on performance

- Minimax solution finds worst-case event distribution
- Leveraged past work to define:
  - Observable metric - Mahalanobis distance
  - Distribution on observable - χ^2 distribution
  - Metric combines information from multiple sensors
- Determine decision policy to minimize worst-case effects

Conclusion

Bound on performance

- Minimax solution finds worst-case event distribution
- Leveraged past work to define:
  - Observable metric - Mahalanobis distance
  - Distribution on observable - χ^2 distribution
  - Metric combines information from multiple sensors
- Determine decision policy to minimize worst-case effects
- Flexible constraints

Conclusion

Bound on performance

- Minimax solution finds worst-case event distribution
- Leveraged past work to define:
  - Observable metric - Mahalanobis distance
  - Distribution on observable - χ^2 distribution
  - Metric combines information from multiple sensors
- Determine decision policy to minimize worst-case effects
- Flexible constraints

Issues:

Conclusion

Bound on performance

- Minimax solution finds worst-case event distribution
- Leveraged past work to define:
  - Observable metric - Mahalanobis distance
  - Distribution on observable - χ^2 distribution
  - Metric combines information from multiple sensors
- Determine decision policy to minimize worst-case effects
- Flexible constraints

Issues:

- Large observable support
  - Hard for optimization tools to handle

Conclusion

Bound on performance

- Minimax solution finds worst-case event distribution
- Leveraged past work to define:
    - Observable metric - Mahalanobis distance
    - Distribution on observable - χ^2 distribution
    - Metric combines information from multiple sensors
- Determine decision policy to minimize worst-case effects
- Flexible constraints

Issues:

- Large observable support
    - Hard for optimization tools to handle
- Cost definition
    - Subjective in nature

Conclusion

Bound on performance

- Minimax solution finds worst-case event distribution
- Leveraged past work to define:
    - Observable metric - Mahalanobis distance
    - Distribution on observable - χ^2 distribution
    - Metric combines information from multiple sensors
- Determine decision policy to minimize worst-case effects
- Flexible constraints

Issues:

- Large observable support
    - Hard for optimization tools to handle
- Cost definition
    - Subjective in nature
- Appropriate constraints

Conclusion

Thank You!

Any Questions?