Strategic methods in adversarial classifier combination

Anshuman Singh and Arun Lakhotia

University of Louisiana at Lafayette

CRW'10 (11/15/10)

Outline

1. Motivation
2. Related work
3. Background
4. Configuration of primitive combinations
5. Summary

Outline

1 Motivation

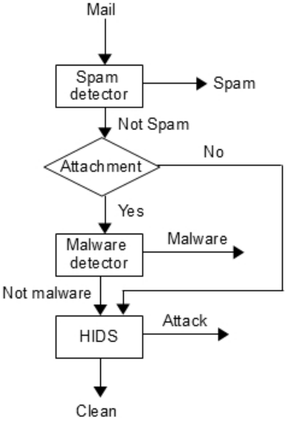
2 Related work

3 Background

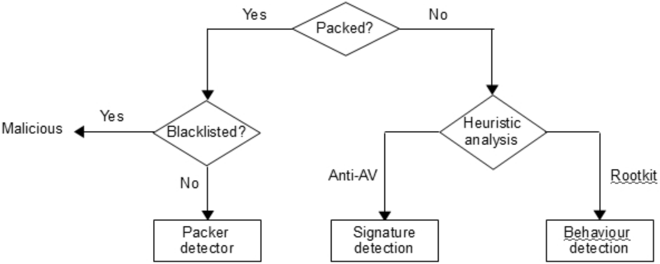
4 Configuration of primitive combinations

5 Summary

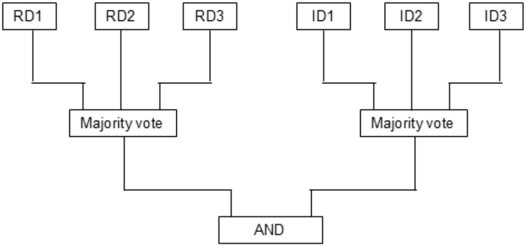
Complex detection systems: An Email Security system



Complex detection systems: A Malware detection system



Complex detection systems: A Multibiometric system



Strategic design of complex detection systems

- Strategic design involves choosing detectors for cost sensitive defenders.
- Strategic decision making considered an art and based on experience.
- Strategic design is important because:
  - Focus on computational issues is effective only in the short run.
  - In the long run, adversary is ahead and the defender is forced to take a reactive approach.

Strategic design of complex detection systems

- Strategic design involves choosing detectors for cost sensitive defenders.
- Strategic decision making considered an art and based on experience.
- Strategic design is important because:
  - Focus on computational issues is effective only in the short run.
  - In the long run, adversary is ahead and the defender is forced to take a reactive approach.

Strategic design of complex detection systems

- Strategic design involves choosing detectors for cost sensitive defenders.
- Strategic decision making considered an art and based on experience.
- Strategic design is important because:
  - Focus on computational issues is effective only in the short run.
  - In the long run, adversary is ahead and the defender is forced to take a reactive approach.

Strategic design of complex detection systems

- Strategic design involves choosing detectors for cost sensitive defenders.
- Strategic decision making considered an art and based on experience.
- Strategic design is important because:
  - Focus on computational issues is effective only in the short run.
  - In the long run, adversary is ahead and the defender is forced to take a reactive approach.

Strategic design of complex detection systems

- Strategic design involves choosing detectors for cost sensitive defenders.
- Strategic decision making considered an art and based on experience.
- Strategic design is important because:
  - Focus on computational issues is effective only in the short run.
  - In the long run, adversary is ahead and the defender is forced to take a reactive approach.

Cost-sensitive classification

- Classifiers designed to increase detection rates.
- Costs due to a miss different from costs due to false alarm.
- Design a cost-sensitive classifier that minimizes expected costs.

Cost-sensitive classification

- Classifiers designed to increase detection rates.
- Costs due to a miss different from costs due to false alarm.
- Design a cost-sensitive classifier that minimizes expected costs.

Cost-sensitive classification

- Classifiers designed to increase detection rates.
- Costs due to a miss different from costs due to false alarm.
- Design a cost-sensitive classifier that minimizes expected costs.

Strategic design of classifiers

- Determine performance parameters that minimize expected costs against adversarial inputs.
- Adversarial inputs - inputs modified for misclassification by the detection system.
- Set the performance parameters using required level of training.

Strategic design of classifiers

- Determine performance parameters that minimize expected costs against adversarial inputs.
- Adversarial inputs - inputs modified for misclassification by the detection system.
- Set the performance parameters using required level of training.

Strategic design of classifiers

- Determine performance parameters that minimize expected costs against adversarial inputs.
- Adversarial inputs - inputs modified for misclassification by the detection system.
- Set the performance parameters using required level of training.

Multiple classifier systems

- Classifiers combined to increase detection rates.
- Two weak classifiers can be combined to give a strong classifier.
- Combinations may be more vulnerable than individual classifiers.
- Some classifier in the combination more vulnerable than others due the to structure of combination.

Multiple classifier systems

- Classifiers combined to increase detection rates.
- Two weak classifiers can be combined to give a strong classifier.
- Combinations may be more vulnerable than individual classifiers.
- Some classifier in the combination more vulnerable than others due the to structure of combination.

Multiple classifier systems

- Classifiers combined to increase detection rates.
- Two weak classifiers can be combined to give a strong classifier.
- Combinations may be more vulnerable than individual classifiers.
- Some classifier in the combination more vulnerable than others due the to structure of combination.

Multiple classifier systems

- Classifiers combined to increase detection rates.
- Two weak classifiers can be combined to give a strong classifier.
- Combinations may be more vulnerable than individual classifiers.
- Some classifier in the combination more vulnerable than others due the to structure of combination.

Strategic design of multiple classifier systems

- In an adversarial setting, optimal performance parameters of component classifiers depend on the choice of evasion or obfuscation method used by the adversary.
- So far, statistical decision theory used to design optimal classifier systems.
- Strategic interdependence can be modeled using game theory.
- Needed: Theory of adversarial classifier combination using strategic methods from game theory.

Strategic design of multiple classifier systems

- In an adversarial setting, optimal performance parameters of component classifiers depend on the choice of evasion or obfuscation method used by the adversary.
- So far, statistical decision theory used to design optimal classifier systems.
- Strategic interdependence can be modeled using game theory.
- Needed: Theory of adversarial classifier combination using strategic methods from game theory.

Strategic design of multiple classifier systems

- In an adversarial setting, optimal performance parameters of component classifiers depend on the choice of evasion or obfuscation method used by the adversary.
- So far, statistical decision theory used to design optimal classifier systems.
- Strategic interdependence can be modeled using game theory.
- Needed: Theory of adversarial classifier combination using strategic methods from game theory.

Strategic design of multiple classifier systems

- In an adversarial setting, optimal performance parameters of component classifiers depend on the choice of evasion or obfuscation method used by the adversary.
- So far, statistical decision theory used to design optimal classifier systems.
- Strategic interdependence can be modeled using game theory.
- Needed: Theory of adversarial classifier combination using strategic methods from game theory.

Outline

1. Motivation
2. Related work
3. Background
4. Configuration of primitive combinations
5. Summary

Adversarial classification

- Dalvi *et al.* Adversarial classification. *Proc. ACM KDD*, 2004).

- Interaction between adversary and classifier is modeled as an extensive game.
  - Classifier and adversary are cost-sensitive.
  - Classification function and feature change function are assumed to be public information.

- Adversary first decides the minimum-cost feature change strategy followed by the classifier adapting the classification function to the possibility of feature change by the adversary.

- Nash equilibrium is computed which constitutes
  - the optimal classification function for the classifier; and
  - the minimum cost feature change function for adversary

Adversarial classification

- Dalvi *et al.* Adversarial classification. *Proc. ACM KDD*, 2004).
- Interaction between adversary and classifier is modeled as an extensive game.
  - Classifier and adversary are cost-sensitive.
  - Classification function and feature change function are assumed to be public information.
- Adversary first decides the minimum-cost feature change strategy followed by the classifier adapting the classification function to the possibility of feature change by the adversary.
- Nash equilibrium is computed which constitutes
  - the optimal classification function for the classifier; and
  - the minimum cost feature change function for adversary

Adversarial learning

- D. Lowd and C. Meek. Adversarial learning. *Proc. ACM KDD*, 2005.
- Adversary doesn't know the classification function in advance.
- New learning paradigm suitable for adversarial problems, called adversarial classifier reverse engineering (ACRE) is introduced.
- The goal is not to learn the entire decision surface.
- The adversary tries to learn instances that are not labeled malicious in polynomial number of queries.

Adversarial learning

- D. Lowd and C. Meek. Adversarial learning. *Proc. ACM KDD*, 2005.
- Adversary doesn't know the classification function in advance.
- New learning paradigm suitable for adversarial problems, called adversarial classifier reverse engineering (ACRE) is introduced.
- The goal is not to learn the entire decision surface.
- The adversary tries to learn instances that are not labeled malicious in polynomial number of queries.

Two doctoral dissertations

- Benjamin Rubinstein. *Secure Learning and Learning for Security: Research in Intersection*. University of California, Berkeley, 2009.
  - How learners can be manipulated by poisoning the training data is studied.
  - A case study of evasion using minimum queries of already trained classifier is presented.
- Battista Biggio. *Adversarial Pattern Classification*. University of Cagliari, 2010.
  - An analysis of different attacks on different stages of a pattern recognition systems (data preprocessing, feature extraction, model training etc.) is given.
  - A methodology of evaluating the robustness of a classifier at design phase is proposed

Two doctoral dissertations

- Benjamin Rubinstein. *Secure Learning and Learning for Security: Research in Intersection*. University of California, Berkeley, 2009.
  - How learners can be manipulated by poisoning the training data is studied.
  - A case study of evasion using minimum queries of already trained classifier is presented.
- Battista Biggio. *Adversarial Pattern Classification*. University of Cagliari, 2010.
  - An analysis of different attacks on different stages of a pattern recognition systems (data preprocessing, feature extraction, model training etc.) is given.
  - A methodology of evaluating the robustness of a classifier at design phase is proposed

Outline

1. Motivation
2. Related work
3. Background
4. Configuration of primitive combinations
5. Summary

Classifier performance parameters

- **True positive rate**. Also called *hit rate*, *recall* and *sensitivity*.

$$\text{tp rate} = \frac{\text{Positives correctly classified}}{\text{Total positives}} = \frac{TP}{TP + FN} = \frac{TP}{P}$$

- **False positive rate**. Also called *false alarm rate*.

$$\text{fp rate} = \frac{\text{Negatives incorrectly classified}}{\text{Total Negatives}} = \frac{FP}{FP + TN} = \frac{FP}{N}$$

- **Cost Matrix**. Given a classifier $C : X \to \Omega$, where $\Omega = \{1, 2, \ldots, m\}$ is the class space, the performance of C can be described using an $m \times m$ matrix $C_{cost} = [c_{ij}]$ where c_{ij} is the cost of assigning class j to an instance of input with true class i for $i, j = 1, 2, \ldots, m$. The cost of correct classification is zero, i.e. $c_{ii} = 0, i = 1, 2, \ldots, m$.

Classifier performance parameters

- **True positive rate**. Also called *hit rate*, *recall* and *sensitivity*.

$$\text{tp rate} = \frac{\text{Positives correctly classified}}{\text{Total positives}} = \frac{TP}{TP + FN} = \frac{TP}{P}$$

- **False positive rate**. Also called *false alarm rate*.

$$\text{fp rate} = \frac{\text{Negatives incorrectly classified}}{\text{Total Negatives}} = \frac{FP}{FP + TN} = \frac{FP}{N}$$

- **Cost Matrix**. Given a classifier $C : X \to \Omega$, where $\Omega = \{1, 2, \ldots, m\}$ is the class space, the performance of C can be described using an $m \times m$ matrix $C_{cost} = [c_{ij}]$ where c_{ij} is the cost of assigning class j to an instance of input with true class i for $i, j = 1, 2, \ldots, m$. The cost of correct classification is zero, i.e. $c_{ii} = 0, i = 1, 2, \ldots, m$.

Classifier performance parameters

- **True positive rate**. Also called *hit rate*, *recall* and *sensitivity*.

$$\text{tp rate} = \frac{\text{Positives correctly classified}}{\text{Total positives}} = \frac{TP}{TP + FN} = \frac{TP}{P}$$

- **False positive rate**. Also called *false alarm rate*.

$$\text{fp rate} = \frac{\text{Negatives incorrectly classified}}{\text{Total Negatives}} = \frac{FP}{FP + TN} = \frac{FP}{N}$$

- **Cost Matrix**. Given a classifier $C : X \rightarrow \Omega$, where $\Omega = \{1, 2, \ldots, m\}$ is the class space, the performance of C can be described using an $m \times m$ matrix $C_{cost} = [c_{ij}]$ where c_{ij} is the cost of assigning class j to an instance of input with true class i for $i, j = 1, 2, \ldots, m$. The cost of correct classification is zero, i.e. $c_{ii} = 0, i = 1, 2, \ldots, m$.

Types of costs in classification

1. **Operational Cost**: involves the computing resources needed to make the classification decision.
2. **Damage Cost**: characterizes the damage done when the classifier misses an attack.
3. **Response Cost**: cost of responding when there is a positive classification (or an alarm) by the classifier irrespective of whether it is correct or not.

Types of costs in classification

1. **Operational Cost**: involves the computing resources needed to make the classification decision.
2. **Damage Cost**: characterizes the damage done when the classifier misses an attack.
3. **Response Cost**: cost of responding when there is a positive classification (or an alarm) by the classifier irrespective of whether it is correct or not.

Types of costs in classification

1. **Operational Cost**: involves the computing resources needed to make the classification decision.
2. **Damage Cost**: characterizes the damage done when the classifier misses an attack.
3. **Response Cost**: cost of responding when there is a positive classification (or an alarm) by the classifier irrespective of whether it is correct or not.

Multiple classifier systems: Fusion

- Classifiers are connected together in parallel so that for any given input all classifiers are run, and the outputs are combined using some decision function.
- Decisions of individual classifiers are fused when
  - entire feature space is the input to all classifiers
  - error rate of all classifiers are almost identical
- The most commonly used fusion function is the majority vote.

Multiple classifier systems: Fusion

- Classifiers are connected together in parallel so that for any given input all classifiers are run, and the outputs are combined using some decision function.
- Decisions of individual classifiers are fused when
  - entire feature space is the input to all classifiers
  - error rate of all classifiers are almost identical
- The most commonly used fusion function is the majority vote.

Multiple classifier systems: Fusion

- Classifiers are connected together in parallel so that for any given input all classifiers are run, and the outputs are combined using some decision function.
- Decisions of individual classifiers are fused when
  - entire feature space is the input to all classifiers
  - error rate of all classifiers are almost identical
- The most commonly used fusion function is the majority vote.

Multiple classifier systems: Fusion

- Classifiers are connected together in parallel so that for any given input all classifiers are run, and the outputs are combined using some decision function.
- Decisions of individual classifiers are fused when
  - entire feature space is the input to all classifiers
  - error rate of all classifiers are almost identical
- The most commonly used fusion function is the majority vote.

Multiple classifier systems: Fusion

- Classifiers are connected together in parallel so that for any given input all classifiers are run, and the outputs are combined using some decision function.
- Decisions of individual classifiers are fused when
  - entire feature space is the input to all classifiers
  - error rate of all classifiers are almost identical
- The most commonly used fusion function is the majority vote.

Multiple classifier systems: Selection

- Classifiers connected together using lightweight classifiers called selectors.
- Each classifier is an expert on a part of the feature space.
- Selector decides which part of the feature space needs to be used for classification and the input is directed to the corresponding classifier.

Multiple classifier systems: Selection

- Classifiers connected together using lightweight classifiers called selectors.
- Each classifier is an expert on a part of the feature space.
- Selector decides which part of the feature space needs to be used for classification and the input is directed to the corresponding classifier.

Multiple classifier systems: Selection

- Classifiers connected together using lightweight classifiers called selectors.
- Each classifier is an expert on a part of the feature space.
- Selector decides which part of the feature space needs to be used for classification and the input is directed to the corresponding classifier.

Cost sensitive adversary

- **Feature-change cost (Obfuscation cost)**: The cost of making changes to the features in the input for evading classifiers.
- **Learning cost**: The cost of making changes to the feature change function when a modified input gets correctly classified.

Cost sensitive adversary

- **Feature-change cost (Obfuscation cost)**: The cost of making changes to the features in the input for evading classifiers.
- **Learning cost**: The cost of making changes to the feature change function when a modified input gets correctly classified.

Games

“Game theory concerns the behaviour of decision makers whose decisions affect each other”. Game theory is a generalization of decision theory. Decision theory is essentially one person game theory.

In general, any game involves:

- **Players**: An individual or a group of individuals can be considered a player.
- **Actions(Strategies)**: The set of moves available to choose from for each player .
- **Outcomes**: An outcome in a game is the act of each player choosing a move from its action set.
- **Preferences**: Each player prefers some outcomes over others based on payoffs or utilities associated with these outcomes.

Games

"Game theory concerns the behaviour of decision makers whose decisions affect each other". Game theory is a generalization of decision theory. Decision theory is essentially one person game theory.

In general, any game involves:

- **Players**: An individual or a group of individuals can be considered a player.
- **Actions(Strategies)**: The set of moves available to choose from for each player .
- **Outcomes**: An outcome in a game is the act of each player choosing a move from its action set.
- **Preferences**: Each player prefers some outcomes over others based on payoffs or utilities associated with these outcomes.

Games

"Game theory concerns the behaviour of decision makers whose decisions affect each other". Game theory is a generalization of decision theory. Decision theory is essentially one person game theory.

In general, any game involves:

- **Players**: An individual or a group of individuals can be considered a player.
- **Actions(Strategies)**: The set of moves available to choose from for each player .
- **Outcomes**: An outcome in a game is the act of each player choosing a move from its action set.
- **Preferences**: Each player prefers some outcomes over others based on payoffs or utilities associated with these outcomes.

Games

"Game theory concerns the behaviour of decision makers whose decisions affect each other". Game theory is a generalization of decision theory. Decision theory is essentially one person game theory.

In general, any game involves:

- **Players**: An individual or a group of individuals can be considered a player.
- **Actions(Strategies)**: The set of moves available to choose from for each player .
- **Outcomes**: An outcome in a game is the act of each player choosing a move from its action set.
- **Preferences**: Each player prefers some outcomes over others based on payoffs or utilities associated with these outcomes.

Games

"Game theory concerns the behaviour of decision makers whose decisions affect each other". Game theory is a generalization of decision theory. Decision theory is essentially one person game theory.

In general, any game involves:

- **Players**: An individual or a group of individuals can be considered a player.
- **Actions(Strategies)**: The set of moves available to choose from for each player .
- **Outcomes**: An outcome in a game is the act of each player choosing a move from its action set.
- **Preferences**: Each player prefers some outcomes over others based on payoffs or utilities associated with these outcomes.

Games

"Game theory concerns the behaviour of decision makers whose decisions affect each other". Game theory is a generalization of decision theory. Decision theory is essentially one person game theory.

In general, any game involves:

- **Players**: An individual or a group of individuals can be considered a player.
- **Actions(Strategies)**: The set of moves available to choose from for each player .
- **Outcomes**: An outcome in a game is the act of each player choosing a move from its action set.
- **Preferences**: Each player prefers some outcomes over others based on payoffs or utilities associated with these outcomes.

Solution of games

- Nash equilibrium - Solution concept for normal form games. An action profile a^* with the property that no player i can do better by choosing an action different from a_i^* , given that every other player j adheres to a_j^*.

- Backward induction- Solution concept for extensive form games. Steps:
  - Determine the optimal choices in the final stage K for each history h^K.
  - Go back to stage $K-1$, and determine the optimal action for the player on the move there, given the optimal choice for stage K.
  - "Roll back" until the initial stage is reached.

Solution of games

- Nash equilibrium - Solution concept for normal form games. An action profile a^* with the property that no player i can do better by choosing an action different from a_i^* , given that every other player j adheres to a_j^*.
- Backward induction- Solution concept for extensive form games. Steps:
  - Determine the optimal choices in the final stage K for each history h^K.
  - Go back to stage $K-1$, and determine the optimal action for the player on the move there, given the optimal choice for stage K.
  - "Roll back" until the initial stage is reached.

Solution of games

- Nash equilibrium - Solution concept for normal form games. An action profile a^* with the property that no player i can do better by choosing an action different from a_i^* , given that every other player j adheres to a_j^*.
- Backward induction- Solution concept for extensive form games. Steps:
  - Determine the optimal choices in the final stage K for each history h^K.
  - Go back to stage $K-1$, and determine the optimal action for the player on the move there, given the optimal choice for stage K.
  - "Roll back" until the initial stage is reached.

Solution of games

- Nash equilibrium - Solution concept for normal form games. An action profile a^* with the property that no player i can do better by choosing an action different from a_i^* , given that every other player j adheres to a_j^*.
- Backward induction- Solution concept for extensive form games. Steps:
  - Determine the optimal choices in the final stage K for each history h^K.
  - Go back to stage $K-1$, and determine the optimal action for the player on the move there, given the optimal choice for stage K.
  - "Roll back" until the initial stage is reached.

Solution of games

- Nash equilibrium - Solution concept for normal form games. An action profile a^* with the property that no player i can do better by choosing an action different from a_i^* , given that every other player j adheres to a_j^*.
- Backward induction- Solution concept for extensive form games. Steps:
  - Determine the optimal choices in the final stage K for each history h^K.
  - Go back to stage $K-1$, and determine the optimal action for the player on the move there, given the optimal choice for stage K.
  - "Roll back" until the initial stage is reached.

Outline

1. Motivation
2. Related work
3. Background
4. Configuration of primitive combinations
5. Summary

Cost model

- d - damage cost when MCS misses an attack
- r - response cost when there is an alarm or a positive classification by MCS
- φ - feature change cost for the adversary
- λ - learning cost for the adversary when the classifier correctly detects an attack
- p_D - detection rate (true positive rate) of the classifier
- p_F - false positive rate
- ROC curve given by the power function $p_D = p_F^r$, with $0 < r < 1$

Cost matrix

		Detected class $-$	Detected class $+$
True class	$-$	0	r
	$+$	d	r

MCS

		Detected class $-$	Detected class $+$
True class	$-$	0	0
	$+$	φ	$\varphi + \lambda$

Adversary

Configuring selection combination

- Adversary can game the selector to evade the selection combination.
- Evading the selector will direct the input to the classifier that has greater probability of misclassifying.
- Assumption: Adversary has complete knowledge of the MCS's combination method and the cost matrix and defender has complete knowledge about the adversary's cost matrix.
- The defender has two options:
  1. Configure the selector by anticipating the optimum cost of input modification by the adversary. We analyze this option using extensive game.
  2. Randomize between selection combination and one of the classifiers. We analyze this option using static game in mixed strategies.

Configuring selection combination

- Adversary can game the selector to evade the selection combination.
- Evading the selector will direct the input to the classifier that has greater probability of misclassifying.
- Assumption: Adversary has complete knowledge of the MCS's combination method and the cost matrix and defender has complete knowledge about the adversary's cost matrix.
- The defender has two options:
  1. Configure the selector by anticipating the optimum cost of input modification by the adversary. We analyze this option using extensive game.
  2. Randomize between selection combination and one of the classifiers. We analyze this option using static game in mixed strategies.

Configuring selection combination

- Adversary can game the selector to evade the selection combination.
- Evading the selector will direct the input to the classifier that has greater probability of misclassifying.
- Assumption: Adversary has complete knowledge of the MCS's combination method and the cost matrix and defender has complete knowledge about the adversary's cost matrix.
- The defender has two options:
  1. Configure the selector by anticipating the optimum cost of input modification by the adversary. We analyze this option using extensive game.
  2. Randomize between selection combination and one of the classifiers. We analyze this option using static game in mixed strategies.

Configuring selection combination

- Adversary can game the selector to evade the selection combination.
- Evading the selector will direct the input to the classifier that has greater probability of misclassifying.
- Assumption: Adversary has complete knowledge of the MCS's combination method and the cost matrix and defender has complete knowledge about the adversary's cost matrix.
- The defender has two options:
  1. Configure the selector by anticipating the optimum cost of input modification by the adversary. We analyze this option using extensive game.
  2. Randomize between selection combination and one of the classifiers. We analyze this option using static game in mixed strategies.

Configuring selection combination

- Adversary can game the selector to evade the selection combination.
- Evading the selector will direct the input to the classifier that has greater probability of misclassifying.
- Assumption: Adversary has complete knowledge of the MCS's combination method and the cost matrix and defender has complete knowledge about the adversary's cost matrix.
- The defender has two options:
  1. Configure the selector by anticipating the optimum cost of input modification by the adversary. We analyze this option using extensive game.
  2. Randomize between selection combination and one of the classifiers. We analyze this option using static game in mixed strategies.

Configuring selection combination

- Adversary can game the selector to evade the selection combination.
- Evading the selector will direct the input to the classifier that has greater probability of misclassifying.
- Assumption: Adversary has complete knowledge of the MCS's combination method and the cost matrix and defender has complete knowledge about the adversary's cost matrix.
- The defender has two options:
  1. Configure the selector by anticipating the optimum cost of input modification by the adversary. We analyze this option using extensive game.
  2. Randomize between selection combination and one of the classifiers. We analyze this option using static game in mixed strategies.

Extensive game analysis

- MCS is configured before adversary attacks.
- Defender decides the method of classification for the selector which determines the detection rate p_D^S.
- Adversary decides the obfuscation method (feature change cost φ) to evade the MCS.
- Extensive game of complete information - can be solved using backward induction to give the equilibrium outcome.
- Equilibrium outcome - the pair (selector accuracy, cost of evasion).

Extensive game analysis

- MCS is configured before adversary attacks.
- Defender decides the method of classification for the selector which determines the detection rate p_D^S.
- Adversary decides the obfuscation method (feature change cost φ) to evade the MCS.
- Extensive game of complete information - can be solved using backward induction to give the equilibrium outcome.
- Equilibrium outcome - the pair (selector accuracy, cost of evasion).

Extensive game analysis

- MCS is configured before adversary attacks.
- Defender decides the method of classification for the selector which determines the detection rate p_D^S.
- Adversary decides the obfuscation method (feature change cost φ) to evade the MCS.
- Extensive game of complete information - can be solved using backward induction to give the equilibrium outcome.
- Equilibrium outcome - the pair (selector accuracy, cost of evasion).

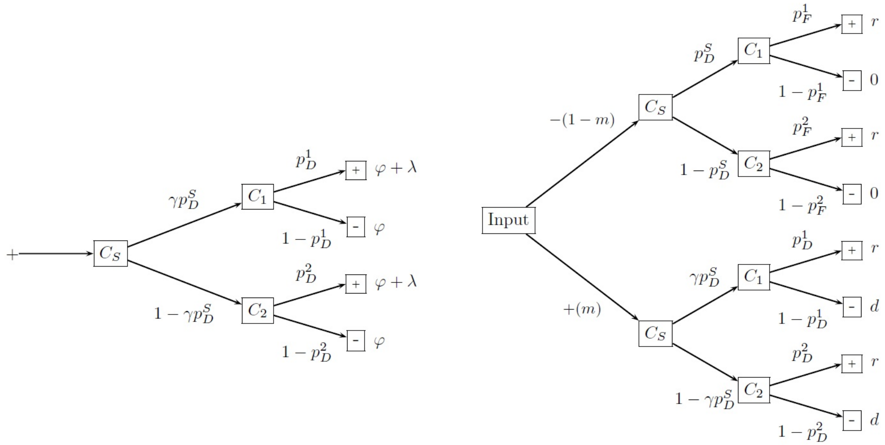
Extensive game analysis

- MCS is configured before adversary attacks.
- Defender decides the method of classification for the selector which determines the detection rate p_D^S.
- Adversary decides the obfuscation method (feature change cost φ) to evade the MCS.
- Extensive game of complete information - can be solved using backward induction to give the equilibrium outcome.
- Equilibrium outcome - the pair (selector accuracy, cost of evasion).

Extensive game analysis

- MCS is configured before adversary attacks.
- Defender decides the method of classification for the selector which determines the detection rate p_D^S.
- Adversary decides the obfuscation method (feature change cost φ) to evade the MCS.
- Extensive game of complete information - can be solved using backward induction to give the equilibrium outcome.
- Equilibrium outcome - the pair (selector accuracy, cost of evasion).

Extensive game analysis (Contd)



Extensive game analysis (Contd)

- Expected cost of the MCS:

$$
\begin{aligned}
E[c_{MCS}] =& m(p_D^S(p_F^1(r) + (1 - p_F^1)(0)) + \\
& (1 - p_D^S)(p_F^2(r) + (1 - p_F^2)(0))) + \\
& (1 - m)(\gamma p_D^S(p_D^1(r) + (1 - p_D^1)(d)) + \\
& (1 - \gamma p_D^S)(p_D^2(r) + (1 - p_D^2)(d)))
\end{aligned}
$$

- Expected cost of the adversary:

$$
\begin{aligned}
E[c_{Ad}] =& \gamma(p_D^S(p_D^1(\varphi + \lambda) + (1 - p_D^1)(\varphi)) + \\
& (1 - \gamma p_D^S)(p_D^2(\varphi + \lambda) + (1 - p_D^2)\varphi))
\end{aligned}
$$

Extensive game analysis (Contd)

- The equilibrium solution (p_D^S, φ) can be obtained by solving the following constrained optimization problem:

$$\min_{p_D^S} E[c_{MCS}]$$

subject to

$$\min_{\varphi} E[c_{Ad}]$$

- Expected cost of the adversary can be simplified to obtain

$$E[c_{Ad}] = \gamma p_D^S \lambda (p_D^1 - p_D^2) + p_D^2 \lambda + \varphi$$

Extensive game analysis (Contd)

- The equilibrium solution (p_D^S, φ) can be obtained by solving the following constrained optimization problem:

$$\min_{p_D^S} E[c_{MCS}]$$

subject to

$$\min_{\varphi} E[c_{Ad}]$$

- Expected cost of the adversary can be simplified to obtain

$$E[c_{Ad}] = \gamma p_D^S \lambda (p_D^1 - p_D^2) + p_D^2 \lambda + \varphi$$

Extensive game analysis (Contd)

- Selector's accuracy degradation factor γ depends on the feature change cost φ. If the selector is robust enough, then a small increase in γ will be obtained with a larger increase in φ. Assuming, $\gamma = \sqrt{\varphi}$,

$$\frac{d(\sqrt{\varphi} p_D^S \lambda (p_D^1 - p_D^2) + p_D^2 \lambda + \varphi)}{d\varphi} = 0$$

  yields

$$\varphi = \left[\frac{p_D^S \lambda (p_D^2 - p_D^1)}{2} \right]^2$$

- The value of φ computed above can be substituted in $\min_{p_D^S} E[c_{MCS}]$ using $\gamma = \sqrt{\varphi}$ to obtain the equilibrium value of p_D^S.
- The optimal p_F^S can be obtained using the ROC function.

Extensive game analysis (Contd)

- Selector's accuracy degradation factor γ depends on the feature change cost φ. If the selector is robust enough, then a small increase in γ will be obtained with a larger increase in φ. Assuming, $\gamma = \sqrt{\varphi}$,

$$\frac{d(\sqrt{\varphi} p_D^S \lambda (p_D^1 - p_D^2) + p_D^2 \lambda + \varphi)}{d\varphi} = 0$$

yields

$$\varphi = \left[\frac{p_D^S \lambda (p_D^2 - p_D^1)}{2} \right]^2$$

- The value of φ computed above can be substituted in $\min_{p_D^S} E[c_{MCS}]$ using $\gamma = \sqrt{\varphi}$ to obtain the equilibrium value of p_D^S.
- The optimal p_F^S can be obtained using the ROC function.

Extensive game analysis (Contd)

- Selector's accuracy degradation factor γ depends on the feature change cost φ. If the selector is robust enough, then a small increase in γ will be obtained with a larger increase in φ. Assuming, $\gamma = \sqrt{\varphi}$,

$$\frac{d(\sqrt{\varphi} p_D^S \lambda (p_D^1 - p_D^2) + p_D^2 \lambda + \varphi)}{d\varphi} = 0$$

yields

$$\varphi = \left[\frac{p_D^S \lambda (p_D^2 - p_D^1)}{2} \right]^2$$

- The value of φ computed above can be substituted in $\min_{p_D^S} E[c_{MCS}]$ using $\gamma = \sqrt{\varphi}$ to obtain the equilibrium value of p_D^S.
- The optimal p_F^S can be obtained using the ROC function.

Random probabilistic selection

- Defender decides whether to use selection combination or not.
- Adversary decides whether to game the combination or the single classifier.
- Use random probabilistic selection based on random primitive as a defense against gaming of the selector.
- Consider a static game in mixed strategies where both players randomize between the two options.
- Solve the game for optimal randomization probability for each player.

Random probabilistic selection

- Defender decides whether to use selection combination or not.
- Adversary decides whether to game the combination or the single classifier.
- Use random probabilistic selection based on random primitive as a defense against gaming of the selector.
- Consider a static game in mixed strategies where both players randomize between the two options.
- Solve the game for optimal randomization probability for each player.

Random probabilistic selection

- Defender decides whether to use selection combination or not.
- Adversary decides whether to game the combination or the single classifier.
- Use random probabilistic selection based on random primitive as a defense against gaming of the selector.
- Consider a static game in mixed strategies where both players randomize between the two options.
- Solve the game for optimal randomization probability for each player.

Random probabilistic selection

- Defender decides whether to use selection combination or not.
- Adversary decides whether to game the combination or the single classifier.
- Use random probabilistic selection based on random primitive as a defense against gaming of the selector.
- Consider a static game in mixed strategies where both players randomize between the two options.
- Solve the game for optimal randomization probability for each player.

Random probabilistic selection

- Defender decides whether to use selection combination or not.
- Adversary decides whether to game the combination or the single classifier.
- Use random probabilistic selection based on random primitive as a defense against gaming of the selector.
- Consider a static game in mixed strategies where both players randomize between the two options.
- Solve the game for optimal randomization probability for each player.

Random probabilistic selection

- α - the probability of adversary gaming the selector
- β - the probability of classifier using the selection combination
- Expected cost of the classifier:

$$\pi_C = \alpha\beta c_{11} + (1-\alpha)\beta c_{12} + \alpha(1-\beta)c_{21} + (1-\alpha)(1-\beta)c_{22}$$

- Expected cost of the adversary:

$$\pi_A = \alpha\beta a_{11} + (1-\alpha)\beta a_{12} + \alpha(1-\beta)a_{21} + (1-\alpha)(1-\beta)a_{22}$$

- The Nash equilibrium (α, β) can be obtained by solving the simultaneous equations $\frac{\partial \pi_C}{\partial \alpha}$ and $\frac{\partial \pi_A}{\partial \beta}$ for α and β.

Random probabilistic selection

- α - the probability of adversary gaming the selector
- β - the probability of classifier using the selection combination
- Expected cost of the classifier:

$$\pi_C = \alpha\beta c_{11} + (1-\alpha)\beta c_{12} + \alpha(1-\beta)c_{21} + (1-\alpha)(1-\beta)c_{22}$$

- Expected cost of the adversary:

$$\pi_A = \alpha\beta a_{11} + (1-\alpha)\beta a_{12} + \alpha(1-\beta)a_{21} + (1-\alpha)(1-\beta)a_{22}$$

- The Nash equilibrium (α, β) can be obtained by solving the simultaneous equations $\frac{\partial \pi_C}{\partial \alpha}$ and $\frac{\partial \pi_A}{\partial \beta}$ for α and β.

Random probabilistic selection

- α - the probability of adversary gaming the selector
- β - the probability of classifier using the selection combination
- Expected cost of the classifier:

$$\pi_C = \alpha\beta c_{11} + (1-\alpha)\beta c_{12} + \alpha(1-\beta)c_{21} + (1-\alpha)(1-\beta)c_{22}$$

- Expected cost of the adversary:

$$\pi_A = \alpha\beta a_{11} + (1-\alpha)\beta a_{12} + \alpha(1-\beta)a_{21} + (1-\alpha)(1-\beta)a_{22}$$

- The Nash equilibrium (α, β) can be obtained by solving the simultaneous equations $\frac{\partial \pi_C}{\partial \alpha}$ and $\frac{\partial \pi_A}{\partial \beta}$ for α and β.

Majority voting as a combination of boolean AND and OR

- Majority voting - most commonly used decision combination method for fusion.

- If there are three classifiers, majority voting can be given using boolean AND ($\wedge$) and OR ($\vee$) as follows:

$$(C_1 \wedge C_2) \vee (C_2 \wedge C_3) \vee (C_1 \wedge C_3)$$

- This can be generalized to n classifiers as:

$$\mathcal{C}_1 \vee \mathcal{C}_2 \vee \ldots \mathcal{C}_k$$

where

$$\mathcal{C}_1 = C_1 \wedge C_2 \wedge \ldots C_l$$

and so on, $k = \binom{n}{l}$, and $l = (n/2) + 1$ when n is even or $l = (n+1)/2$ when n is odd.

Majority voting as a combination of boolean AND and OR

- Majority voting - most commonly used decision combination method for fusion.

- If there are three classifiers, majority voting can be given using boolean AND ($\wedge$) and OR ($\vee$) as follows:

$$(C_1 \wedge C_2) \vee (C_2 \wedge C_3) \vee (C_1 \wedge C_3)$$

- This can be generalized to n classifiers as:

$$\mathcal{C}_1 \vee \mathcal{C}_2 \vee \ldots \mathcal{C}_k$$

where

$$\mathcal{C}_1 = C_1 \wedge C_2 \wedge \ldots C_l$$

and so on, $k = \binom{n}{l}$, and $l = (n/2) + 1$ when n is even or $l = (n+1)/2$ when n is odd.

Majority voting as a combination of boolean AND and OR

- Majority voting - most commonly used decision combination method for fusion.
- If there are three classifiers, majority voting can be given using boolean AND ($\wedge$) and OR ($\vee$) as follows:

$$(C_1 \wedge C_2) \vee (C_2 \wedge C_3) \vee (C_1 \wedge C_3)$$

- This can be generalized to n classifiers as:

$$\mathcal{C}_1 \vee \mathcal{C}_2 \vee \ldots \mathcal{C}_k$$

where

$$\mathcal{C}_1 = C_1 \wedge C_2 \wedge \ldots C_l$$

and so on, $k = \binom{n}{l}$, and $l = (n/2) + 1$ when n is even or $l = (n+1)/2$ when n is odd.

Configuring OR combination

- Adversary will have to evade both the classifiers to evade the OR combination.
- Compute the optimum detection rates of the two classifiers (p_D^1, p_D^2) for the defender and the optimum cost of obfuscating the two classifiers (φ_1, φ_2) for the adversary.
- Degradation in the detection rate of classifiers C_1 and C_2 due to obfuscation is denoted by γ_1 and γ_2, respectively.
- Consider sequential game - defender configures before adversary attacks.

Configuring OR combination

- Adversary will have to evade both the classifiers to evade the OR combination.
- Compute the optimum detection rates of the two classifiers (p_D^1, p_D^2) for the defender and the optimum cost of obfuscating the two classifiers (φ_1, φ_2) for the adversary.
- Degradation in the detection rate of classifiers C_1 and C_2 due to obfuscation is denoted by γ_1 and γ_2, respectively.
- Consider sequential game - defender configures before adversary attacks.

Configuring OR combination

- Adversary will have to evade both the classifiers to evade the OR combination.
- Compute the optimum detection rates of the two classifiers (p_D^1, p_D^2) for the defender and the optimum cost of obfuscating the two classifiers (φ_1, φ_2) for the adversary.
- Degradation in the detection rate of classifiers C_1 and C_2 due to obfuscation is denoted by γ_1 and γ_2, respectively.
- Consider sequential game - defender configures before adversary attacks.

Configuring OR combination

- Adversary will have to evade both the classifiers to evade the OR combination.
- Compute the optimum detection rates of the two classifiers (p_D^1, p_D^2) for the defender and the optimum cost of obfuscating the two classifiers (φ_1, φ_2) for the adversary.
- Degradation in the detection rate of classifiers C_1 and C_2 due to obfuscation is denoted by γ_1 and γ_2, respectively.
- Consider sequential game - defender configures before adversary attacks.

Configuring OR combination

Input true class	C_1	$p(C_1)$	C_2	$p(C_2)$	C_1 OR C_2	$p(C_1$ OR $C_2)$	MCS's payoff
+	+	$\gamma_1 p_D^1$	+	$\gamma_2 p_D^2$	+	$\gamma_1\gamma_2 p_D^1 p_D^2$	r
+	+	$\gamma_1 p_D^1$	-	$1-\gamma_2 p_D^2$	+	$\gamma_1 p_D^1(1-\gamma_2 p_D^2)$	r
+	-	$1-\gamma_1 p_D^1$	+	$\gamma_2 p_D^2$	+	$(1-\gamma_1 p_D^1)\gamma_2 p_D^2$	r
+	-	$1-\gamma_1 p_D^1$	-	$1-\gamma_2 p_D^1$	-	$(1-\gamma_1 p_D^1)(1-\gamma_2 p_D^2)$	d
-	+	p_F^1	+	p_F^2	+	$p_F^1 p_F^2$	r
-	+	p_F^1	-	$1-p_F^2$	+	$p_F^1(1-p_F^2)$	r
-	-	$1-p_F^1$	+	p_F^2	+	$(1-p_F^1)p_F^2$	r
-	-	$1-p_F^1$	-	$1-p_F^1$	-	$(1-p_F^1)(1-p_F^2)$	0

Configuring OR combination

C_1	$p(C_1)$	C_2	$p(C_2)$	C_1 OR C_2	$p(C_1$ OR $C_2)$	Adversary's payoff
+	$\gamma_1 p_D^1$	+	$\gamma_2 p_D^2$	+	$\gamma_1\gamma_2 p_D^1 p_D^2$	$\varphi_1 + \varphi_2 + \lambda_1 + \lambda_2$
+	$\gamma_1 p_D^1$	-	$1 - \gamma_2 p_D^2$	+	$\gamma_1 p_D^1(1 - \gamma_2 p_D^2)$	$\varphi_1 + \varphi_2 + \lambda_1$
-	$1 - \gamma_1 p_D^1$	+	$\gamma_2 p_D^2$	+	$(1 - \gamma_1 p_D^1)\gamma_2 p_D^2$	$\varphi_1 + \varphi_2 + \lambda_2$
-	$1 - \gamma_1 p_D^1$	-	$1 - \gamma_2 p_D^2$	-	$(1 - \gamma_1 p_D^1)(1 - \gamma_2 p_D^2)$	$\varphi_1 + \varphi_2$

Configuring OR combination

Nash equilibrium $((p_D^1, p_D^2), (\varphi_1, \varphi_2))$ can be obtained by solving the following constrained optimization problem:

$$\min_{p_D^1, p_D^2} E[c_{MCS}]$$

subject to

$$\min_{\varphi_1, \varphi_2} E[c_{Ad}]$$

Configuring AND combination

- Adversary can evade the AND combination by evading the classifier for which the cost of evasion is lower.
- Defender randomizes between AND combination and the classifier with higher detection rate.
- Adversary - forced to randomize between evading the AND combination and evading the classifier with higher detection rate.
- Model this situation as a simultaneous (static) game in mixed strategies - analysis is similar to random probabilistic selection.

Configuring AND combination

- Adversary can evade the AND combination by evading the classifier for which the cost of evasion is lower.
- Defender randomizes between AND combination and the classifier with higher detection rate.
- Adversary - forced to randomize between evading the AND combination and evading the classifier with higher detection rate.
- Model this situation as a simultaneous (static) game in mixed strategies - analysis is similar to random probabilistic selection.

Configuring AND combination

- Adversary can evade the AND combination by evading the classifier for which the cost of evasion is lower.
- Defender randomizes between AND combination and the classifier with higher detection rate.
- Adversary - forced to randomize between evading the AND combination and evading the classifier with higher detection rate.
- Model this situation as a simultaneous (static) game in mixed strategies - analysis is similar to random probabilistic selection.

Configuring AND combination

- Adversary can evade the AND combination by evading the classifier for which the cost of evasion is lower.
- Defender randomizes between AND combination and the classifier with higher detection rate.
- Adversary - forced to randomize between evading the AND combination and evading the classifier with higher detection rate.
- Model this situation as a simultaneous (static) game in mixed strategies - analysis is similar to random probabilistic selection.

Outline

1 Motivation

2 Related work

3 Background

4 Configuration of primitive combinations

5 Summary

Summary

- Presented an analytical method of configuring performance parameters of individual classifiers in a multiple classifier system (MCS) such that system as a whole incurs minimum misclassification costs.

- Primitive combinations (OR, AND, SELECT) made "adversary-aware" using game-theoretic analysis.

- Probabilistic randomization as a defense strategy for selection combination.

- Backward induction based configuration of OR combination for expected cost minimization.

Summary

- Presented an analytical method of configuring performance parameters of individual classifiers in a multiple classifier system (MCS) such that system as a whole incurs minimum misclassification costs.
- Primitive combinations (OR, AND, SELECT) made "adversary-aware" using game-theoretic analysis.
- Probabilistic randomization as a defense strategy for selection combination.
- Backward induction based configuration of OR combination for expected cost minimization.

Summary

- Presented an analytical method of configuring performance parameters of individual classifiers in a multiple classifier system (MCS) such that system as a whole incurs minimum misclassification costs.
- Primitive combinations (OR, AND, SELECT) made "adversary-aware" using game-theoretic analysis.
- Probabilistic randomization as a defense strategy for selection combination.
- Backward induction based configuration of OR combination for expected cost minimization.

Summary

- Presented an analytical method of configuring performance parameters of individual classifiers in a multiple classifier system (MCS) such that system as a whole incurs minimum misclassification costs.
- Primitive combinations (OR, AND, SELECT) made "adversary-aware" using game-theoretic analysis.
- Probabilistic randomization as a defense strategy for selection combination.
- Backward induction based configuration of OR combination for expected cost minimization.

Questions ?