

Payment Rules through Discriminant-Based Classifiers

by Paul Dütting, Felix Fischer, Pichayut Jirapinyo,
John K. Lai, Benjamin Lubin and David C. Parkes

Nicolas Küchler

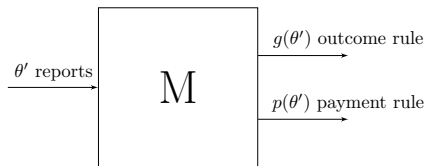
University of Zurich

nicolas.kuechler@uzh.ch

May 19, 2017

Classical Approach for Mechanism Design

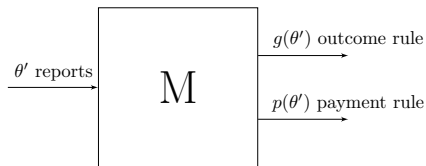
Classical Approach



Classical Approach for Mechanism Design

Classical Approach

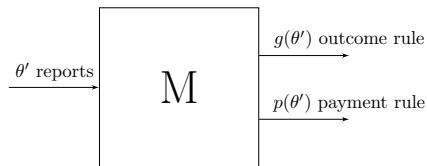
- 1 Impose incentive compatibility (IC) constraint (DSIC, BNIC)



Classical Approach for Mechanism Design

Classical Approach

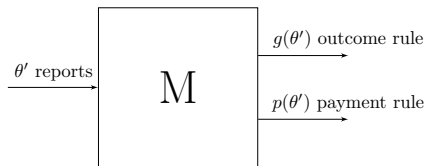
- 1 Impose incentive compatibility (IC) constraint (DSIC, BNIC)
- 2 Design outcome- and payment rule subject to IC constraint



Classical Approach for Mechanism Design

Classical Approach

- 1 Impose incentive compatibility (IC) constraint (DSIC, BNIC)
- 2 Design outcome- and payment rule subject to IC constraint

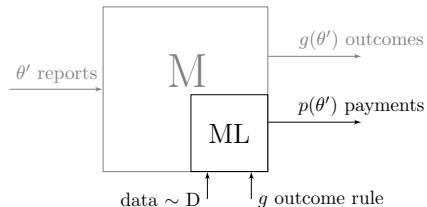


Challenges

- Analytical Complexity
- Exclusion of Mechanisms
- Computational Complexity

New Approach for Mechanism Design

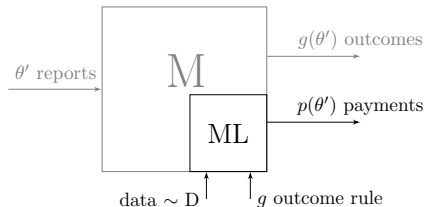
New Approach



New Approach for Mechanism Design

New Approach

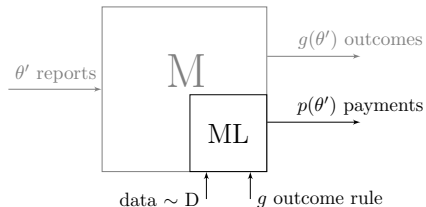
- 1 Define an outcome rule g (e.g. optimal outcome rule)



New Approach for Mechanism Design

New Approach

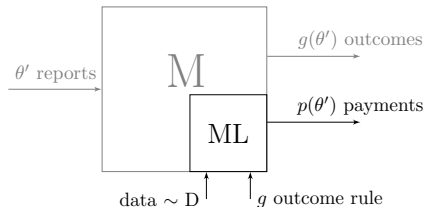
- 1 Define an outcome rule g (e.g. optimal outcome rule)
- 2 Sample data from a type distribution D



New Approach for Mechanism Design

New Approach

- 1 Define an outcome rule g (e.g. optimal outcome rule)
- 2 Sample data from a type distribution D
- 3 Use *Machine Learning (ML)* to find a payment rule p that minimizes ex-post regret



Ex-Post Regret

ex-post regret an agent has for truthfully reporting in a given instance is the amount by which its utility could be increased through a misreport.

$$rgt_i(\theta_i, \theta'_{-i}) = \max_{\theta'_i \in \Theta_i} \underbrace{u_i((\theta'_i, \theta'_{-i}), \theta_i)}_{\text{utility misreport}} - \underbrace{u_i((\theta_i, \theta'_{-i}), \theta_i)}_{\text{utility truthful report}}$$

Ex-Post Regret

ex-post regret an agent has for truthfully reporting in a given instance is the amount by which its utility could be increased through a misreport.

$$rgt_i(\theta_i, \theta'_{-i}) = \max_{\theta'_i \in \Theta_i} \underbrace{u_i((\theta'_i, \theta'_{-i}), \theta_i)}_{\text{utility misreport}} - \underbrace{u_i((\theta_i, \theta'_{-i}), \theta_i)}_{\text{utility truthful report}}$$

Properties

Ex-Post Regret

ex-post regret an agent has for truthfully reporting in a given instance is the amount by which its utility could be increased through a misreport.

$$rgt_i(\theta_i, \theta'_{-i}) = \max_{\theta'_i \in \Theta_i} \underbrace{u_i((\theta'_i, \theta'_{-i}), \theta_i)}_{\text{utility misreport}} - \underbrace{u_i((\theta_i, \theta'_{-i}), \theta_i)}_{\text{utility truthful report}}$$

Properties

- $rgt_i(\theta_i, \theta'_{-i}) = 0 \quad \forall \theta_i, \theta'_{-i} \quad \rightarrow \text{strategyproof}$

Ex-Post Regret

ex-post regret an agent has for truthfully reporting in a given instance is the amount by which its utility could be increased through a misreport.

$$rgt_i(\theta_i, \theta'_{-i}) = \max_{\theta'_i \in \Theta_i} \underbrace{u_i((\theta'_i, \theta'_{-i}), \theta_i)}_{\text{utility misreport}} - \underbrace{u_i((\theta_i, \theta'_{-i}), \theta_i)}_{\text{utility truthful report}}$$

Properties

- $rgt_i(\theta_i, \theta'_{-i}) = 0 \quad \forall \theta_i, \theta'_{-i} \quad \rightarrow \text{strategyproof}$
- $rgt_i(\theta_i, \theta'_{-i}) \neq 0 \quad \exists \theta_i, \theta'_{-i} \quad \rightarrow \text{no direct implications}$

Ex-Post Regret

ex-post regret an agent has for truthfully reporting in a given instance is the amount by which its utility could be increased through a misreport.

$$rgt_i(\theta_i, \theta'_{-i}) = \max_{\theta'_i \in \Theta_i} \underbrace{u_i((\theta'_i, \theta'_{-i}), \theta_i)}_{\text{utility misreport}} - \underbrace{u_i((\theta_i, \theta'_{-i}), \theta_i)}_{\text{utility truthful report}}$$

Properties

- $rgt_i(\theta_i, \theta'_{-i}) = 0 \quad \forall \theta_i, \theta'_{-i}$ \rightarrow *strategyproof*
- $rgt_i(\theta_i, \theta'_{-i}) \neq 0 \quad \exists \theta_i, \theta'_{-i}$ \rightarrow *no direct implications*
- $\mathbb{E}(\text{gain}) < \text{cost}(\text{strategic behaviour})$ \rightarrow *agents are assumed to report truthfully*

Payment Rules from Multi-Class Classifiers

Payment Rules from Multi-Class Classifiers

Properties of a truthful Mechanism

Payment Rules from Multi-Class Classifiers

Properties of a truthful Mechanism

Agent-Independent Price: $p_i(\theta) = t_i(\theta_{-i}, g_i(\theta)) \quad \forall i$

Payment Rules from Multi-Class Classifiers

Properties of a truthful Mechanism

Agent-Independent Price: $p_i(\theta) = t_i(\theta_{-i}, g_i(\theta)) \quad \forall i$

Agent-Optimizing Outcome: $g_i(\theta) \in \arg \max_{o_i \in \Omega_i} v_i(\theta_i, o_i) - t_i(\theta_{-i}, o_i) \quad \forall i$

Payment Rules from Multi-Class Classifiers

Properties of a truthful Mechanism

Agent-Independent Price: $p_i(\theta) = t_i(\theta_{-i}, g_i(\theta)) \quad \forall i$

Agent-Optimizing Outcome: $g_i(\theta) \in \arg \max_{o_i \in \Omega_i} v_i(\theta_i, o_i) - t_i(\theta_{-i}, o_i) \quad \forall i$

Multi-Class Classifier

$$h_w(x) \in \arg \max_{y \in Y} f_w(x, y)$$

Payment Rules from Multi-Class Classifiers

Properties of a truthful Mechanism

Agent-Independent Price: $p_i(\theta) = t_i(\theta_{-i}, g_i(\theta)) \quad \forall i$

Agent-Optimizing Outcome: $g_i(\theta) \in \arg \max_{o_i \in \Omega_i} v_i(\theta_i, o_i) - t_i(\theta_{-i}, o_i) \quad \forall i$

Multi-Class Classifier

$$h_w(x) \in \arg \max_{y \in Y} f_w(x, y)$$

in Mechanism Design:

$$h_w(\theta) \in \arg \max_{o_i \in \Omega_i} f_w(\theta, o_i)$$

Payment Rules from Multi-Class Classifiers

Properties of a truthful Mechanism

Agent-Independent Price: $p_i(\theta) = t_i(\theta_{-i}, g_i(\theta)) \quad \forall i$

Agent-Optimizing Outcome: $g_i(\theta) \in \arg \max_{o_i \in \Omega_i} v_i(\theta_i, o_i) - t_i(\theta_{-i}, o_i) \quad \forall i$

Connection

Multi-Class Classifier

$$h_w(x) \in \arg \max_{y \in Y} f_w(x, y)$$

in Mechanism Design:

$$h_w(\theta) \in \arg \max_{o_i \in \Omega_i} f_w(\theta, o_i)$$

Payment Rules from Multi-Class Classifiers

Properties of a truthful Mechanism

Agent-Independent Price: $p_i(\theta) = t_i(\theta_{-i}, g_i(\theta)) \quad \forall i$

Agent-Optimizing Outcome: $g_i(\theta) \in \arg \max_{o_i \in \Omega_i} v_i(\theta_i, o_i) - t_i(\theta_{-i}, o_i) \quad \forall i$

Connection

Discriminant-Based Classifier: $h_w(\theta) \in \arg \max_{o_i \in \Omega_i} w_i v_i(\theta_i, o_i) + w_{-i}^T \psi(\theta_{-i}, o_i)$

Multi-Class Classifier

$$h_w(x) \in \arg \max_{y \in Y} f_w(x, y)$$

in Mechanism Design: $h_w(\theta) \in \arg \max_{o_i \in \Omega_i} f_w(\theta, o_i)$

Payment Rules from Multi-Class Classifiers

Agent-Independent Price: $p_1(\theta) = t_1(\theta_{-1}, g_1(\theta))$

Agent-Optimizing Outcome: $g_1(\theta) \in \arg \max_{o_1 \in \Omega_1} v_1(\theta_1, o_1) - t_1(\theta_{-1}, o_1)$

Discriminant-Based Classifier: $h_w(\theta) \in \arg \max_{o_1 \in \Omega_1} w_1 v_1(\theta_1, o_1) + w_{-1}^T \psi(\theta_{-1}, o_1)$

Payment Rules from Multi-Class Classifiers

Agent-Independent Price: $p_1(\theta) = t_1(\theta_{-1}, g_1(\theta))$

Agent-Optimizing Outcome: $g_1(\theta) \in \arg \max_{o_1 \in \Omega_1} v_1(\theta_1, o_1) - t_1(\theta_{-1}, o_1)$

Discriminant-Based Classifier: $h_w(\theta) \in \arg \max_{o_1 \in \Omega_1} w_1 v_1(\theta_1, o_1) + w_{-1}^T \psi(\theta_{-1}, o_1)$

Associated Price Function

$$t_w(\theta_{-1}, o_1) = -\frac{1}{w_1} w_{-1}^T \psi(\theta_{-1}, o_1)$$

Payment Rules from Multi-Class Classifiers

Agent-Independent Price: $p_1(\theta) = t_1(\theta_{-1}, g_1(\theta))$

Agent-Optimizing Outcome: $g_1(\theta) \in \arg \max_{o_1 \in \Omega_1} v_1(\theta_1, o_1) - t_1(\theta_{-1}, o_1)$

Discriminant-Based Classifier: $h_w(\theta) \in \arg \max_{o_1 \in \Omega_1} w_1 v_1(\theta_1, o_1) + w_{-1}^T \psi(\theta_{-1}, o_1)$

Associated Price Function

$$t_w(\theta_{-1}, o_1) = -\frac{1}{w_1} w_{-1}^T \psi(\theta_{-1}, o_1)$$

Payment Rule

$$p_w(\theta) = (t_w(\theta_{-1}, g_1(\theta)), t_w(\theta_{-2}, g_2(\theta)), \dots, t_w(\theta_{-n}, g_n(\theta)))$$

Truthful Mechanisms with a Perfect Classifier

Theorem (3.2)

Let g be an agent symmetric outcome rule, h_w an admissible classifier, and p_w the payment rule corresponding to h_w . If h_w is a perfect classifier for the partial outcome rule g_1 , then the mechanism (g, p_w) is strategyproof.

Truthful Mechanisms with a Perfect Classifier

Theorem (3.2)

Let g be an agent symmetric outcome rule, h_w an admissible classifier, and p_w the payment rule corresponding to h_w . If h_w is a perfect classifier for the partial outcome rule g_1 , then the mechanism (g, p_w) is strategyproof.

h_w is a perfect classifier for $g_1 \Rightarrow$ mechanism (g, p_w) is strategyproof

Truthful Mechanisms with a Perfect Classifier

Theorem (3.2)

Let g be an agent symmetric outcome rule, h_w an admissible classifier, and p_w the payment rule corresponding to h_w . If h_w is a perfect classifier for the partial outcome rule g_1 , then the mechanism (g, p_w) is strategyproof.

h_w is a perfect classifier for $g_1 \Rightarrow$ mechanism (g, p_w) is strategyproof

But what happens when h_w is not a perfect classifier?

Truthful Mechanisms with a Perfect Classifier

Theorem (3.2)

Let g be an agent symmetric outcome rule, h_w an admissible classifier, and p_w the payment rule corresponding to h_w . If h_w is a perfect classifier for the partial outcome rule g_1 , then the mechanism (g, p_w) is strategyproof.

h_w is a perfect classifier for $g_1 \Rightarrow$ mechanism (g, p_w) is strategyproof

But what happens when h_w is not a perfect classifier?

minimizing *generalization error* of the classifier \Rightarrow
 minimizing *expected ex-post regret* of the mechanism.

Example: Single Item Auction

Discriminant-Based Classifier: $h_w(\theta) \in \arg \max_{o_1 \in \Omega_1} w_1 v_1(\theta_1, o_1) + w_{-1}^T \psi(\theta_{-1}, o_1)$

Example: Single Item Auction

Discriminant-Based Classifier: $h_w(\theta) \in \arg \max_{o_1 \in \Omega_1} w_1 v_1(\theta_1, o_1) + w_{-1}^T \psi(\theta_{-1}, o_1)$

What outcome o_1 has the classifier to select?

Example: Single Item Auction

Discriminant-Based Classifier: $h_w(\theta) \in \arg \max_{o_1 \in \Omega_1} w_1 v_1(\theta_1, o_1) + w_{-1}^T \psi(\theta_{-1}, o_1)$

What outcome o_1 has the classifier to select?

the outcome matching the outcome rule of the *Single Item Auction*

$$g_1(\theta) = \begin{cases} \text{allocate} & \text{if agent 1 has the highest bid} \\ \text{not allocate} & \text{else} \end{cases}$$

Example: Single Item Auction

Discriminant-Based Classifier: $h_w(\theta) \in \arg \max_{o_1 \in \Omega_1} w_1 v_1(\theta_1, o_1) + w_{-1}^T \psi(\theta_{-1}, o_1)$

What outcome o_1 has the classifier to select?

the outcome matching the outcome rule of the *Single Item Auction*

$$g_1(\theta) = \begin{cases} \text{allocate} & \text{if agent 1 has the highest bid} \\ \text{not allocate} & \text{else} \end{cases}$$

How can this be achieved with the classifier?

Example: Single Item Auction

Discriminant-Based Classifier: $h_w(\theta) \in \arg \max_{o_1 \in \Omega_1} w_1 v_1(\theta_1, o_1) + w_{-1}^T \psi(\theta_{-1}, o_1)$

What outcome o_1 has the classifier to select?

the outcome matching the outcome rule of the *Single Item Auction*

$$g_1(\theta) = \begin{cases} \text{allocate} & \text{if agent 1 has the highest bid} \\ \text{not allocate} & \text{else} \end{cases}$$

How can this be achieved with the classifier?

$$w_1 = 1 \quad \text{and} \quad w_{-1}^T \psi(\theta_{-1}, o_1) = \begin{cases} -\max(\theta_{-1}) & \text{if } o_1 = 1 \\ 0 & \text{if } o_1 = 0 \end{cases}$$

Example: Single Item Auction

Discriminant-Based Classifier: $h_w(\theta) \in \arg \max_{o_1 \in \Omega_1} w_1 v_1(\theta_1, o_1) + w_{-1}^T \psi(\theta_{-1}, o_1)$

What outcome o_1 has the classifier to select?

the outcome matching the outcome rule of the *Single Item Auction*

$$g_1(\theta) = \begin{cases} \text{allocate} & \text{if agent 1 has the highest bid} \\ \text{not allocate} & \text{else} \end{cases}$$

How can this be achieved with the classifier?

$$w_1 = 1 \quad \text{and} \quad w_{-1}^T \psi(\theta_{-1}, o_1) = \begin{cases} -\max(\theta_{-1}) & \text{if } o_1 = 1 \\ 0 & \text{if } o_1 = 0 \end{cases}$$

plugging this into the associated price function \rightarrow second price payment rule

$$t_w(\theta_{-1}, o_1) = -\frac{1}{w_1} w_{-1}^T \psi(\theta_{-1}, o_1) = \max(\theta_{-1})$$

Example: Single Item Auction Instance

$$h_w(\theta) = \arg \max_{o_i} \underbrace{1 * v_i(\theta_i, o_i)}_{\text{left part}} + \underbrace{\begin{cases} -\max(\theta_{-1}) & \text{if } o_i = 1 \\ 0 & \text{if } o_i = 0 \end{cases}}_{\text{right part}}$$

Example: Single Item Auction Instance

$$h_w(\theta) = \arg \max_{o_i} \underbrace{1 * v_i(\theta_i, o_i)}_{\text{left part}} + \underbrace{\begin{cases} -\max(\theta_{-1}) & \text{if } o_i = 1 \\ 0 & \text{if } o_i = 0 \end{cases}}_{\text{right part}}$$

	Outcome	Left Part	Right Part	Classifier	Optimal
Agent 1	$o_1 = 1$				
	$o_1 = 0$				
Agent 2	$o_2 = 1$				
	$o_2 = 0$				

Example: Single Item Auction Instance

$$h_w(\theta) = \arg \max_{o_i} \underbrace{1 * v_i(\theta_i, o_i)}_{\text{left part}} + \underbrace{\begin{cases} -\max(\theta_{-1}) & \text{if } o_i = 1 \\ 0 & \text{if } o_i = 0 \end{cases}}_{\text{right part}}$$

	Outcome	Left Part	Right Part	Classifier	Optimal
Agent 1	$o_1 = 1$	$v_1(\theta_1, 1) = 6$			
	$o_1 = 0$	$v_1(\theta_1, 0) = 0$			
Agent 2	$o_2 = 1$	$v_2(\theta_2, 1) = 8$			
	$o_2 = 0$	$v_2(\theta_2, 0) = 0$			

Example: Single Item Auction Instance

$$h_w(\theta) = \arg \max_{o_i} \underbrace{1 * v_i(\theta_i, o_i)}_{\text{left part}} + \underbrace{\begin{cases} -\max(\theta_{-1}) & \text{if } o_i = 1 \\ 0 & \text{if } o_i = 0 \end{cases}}_{\text{right part}}$$

	Outcome	Left Part	Right Part	Classifier	Optimal
Agent 1	$o_1 = 1$	$v_1(\theta_1, 1) = 6$	8		
	$o_1 = 0$	$v_1(\theta_1, 0) = 0$	0		
Agent 2	$o_2 = 1$	$v_2(\theta_2, 1) = 8$	6		
	$o_2 = 0$	$v_2(\theta_2, 0) = 0$	0		

Example: Single Item Auction Instance

$$h_w(\theta) = \arg \max_{o_i} \underbrace{1 * v_i(\theta_i, o_i)}_{\text{left part}} + \underbrace{\begin{cases} -\max(\theta_{-1}) & \text{if } o_i = 1 \\ 0 & \text{if } o_i = 0 \end{cases}}_{\text{right part}}$$

	Outcome	Left Part	Right Part	Classifier	Optimal
Agent 1	$o_1 = 1$	$v_1(\theta_1, 1) = 6$	8	$6 - 8 = -2$	
	$o_1 = 0$	$v_1(\theta_1, 0) = 0$	0	$0 - 0 = 0$	
Agent 2	$o_2 = 1$	$v_2(\theta_2, 1) = 8$	6	$8 - 6 = 2$	
	$o_2 = 0$	$v_2(\theta_2, 0) = 0$	0	$0 - 0 = 0$	

Example: Single Item Auction Instance

$$h_w(\theta) = \arg \max_{o_i} \underbrace{1 * v_i(\theta_i, o_i)}_{\text{left part}} + \underbrace{\begin{cases} -\max(\theta_{-1}) & \text{if } o_i = 1 \\ 0 & \text{if } o_i = 0 \end{cases}}_{\text{right part}}$$

	Outcome	Left Part	Right Part	Classifier	Optimal
Agent 1	$o_1 = 1$	$v_1(\theta_1, 1) = 6$	8	$6 - 8 = -2$	
	$o_1 = 0$	$v_1(\theta_1, 0) = 0$	0	$0 - 0 = 0$	X
Agent 2	$o_2 = 1$	$v_2(\theta_2, 1) = 8$	6	$8 - 6 = 2$	X
	$o_2 = 0$	$v_2(\theta_2, 0) = 0$	0	$0 - 0 = 0$	

Structural Support Vector Machine (SSVM)

SSVM

Learn weight vector w by training SSVM using:

training data: $\{(\theta^1, o_1^1), (\theta^2, o_1^2), \dots, (\theta^\ell, o_1^\ell)\}$

Structural Support Vector Machine (SSVM)

SSVM

Learn weight vector w by training SSVM using:

training data: $\{(\theta^1, o_1^1), (\theta^2, o_1^2), \dots, (\theta^\ell, o_1^\ell)\}$

SSVM Optimization Problem

$$\begin{aligned}
 &\underset{w, \xi}{\text{minimize}} && \frac{1}{2} \|w\|^2 + \frac{C}{\ell} \sum_{k=1}^{\ell} \xi^k \\
 &\text{subject to} && (w_1 v_1(\theta_1^k, o_1^k) + w_{-1}^T \psi'(\theta_{-1}^k, o_1^k)) - (w_1 v_1(\theta_1^k, o_1) + w_{-1}^T \psi'(\theta_{-1}^k, o_1)) \\
 &&& \geq \mathcal{L}(o_1^k, o_1) - \xi^k, \forall k = 1, \dots, \ell, \quad o_1 \in \Omega_1 \\
 &&& \xi^k \geq 0, \forall k = 1, \dots, \ell
 \end{aligned}$$

Structural Support Vector Machine (SSVM)

SSVM

Learn weight vector w by training SSVM using:

$$\text{training data: } \{(\theta^1, o_1^1), (\theta^2, o_1^2), \dots, (\theta^\ell, o_1^\ell)\}$$

SSVM Optimization Problem

$$\begin{aligned} & \underset{w, \xi}{\text{minimize}} && \frac{1}{2} \|w\|^2 + \frac{C}{\ell} \sum_{k=1}^{\ell} \xi^k \\ & \text{subject to} && (w_1 v_1(\theta_1^k, o_1^k) + w_{-1}^T \psi'(\theta_{-1}^k, o_1^k)) - (w_1 v_1(\theta_1^k, o_1) + w_{-1}^T \psi'(\theta_{-1}^k, o_1)) \\ & && \geq \mathcal{L}(o_1^k, o_1) - \xi^k, \forall k = 1, \dots, \ell, \quad o_1 \in \Omega_1 \\ & && \xi^k \geq 0, \forall k = 1, \dots, \ell \end{aligned}$$

Remarks

Valid Price Function if $w_1 > 0$

Flexible Payment Structure with ψ

Problems of the computed Payment Rule

Negative Payments

computed payments p_w can be negative \rightarrow normalize payments

Violation of Individual Rationality

truthful report leads to $utility < 0$ \rightarrow

- introduce payment offsets
- adjust the loss function \mathcal{L}
- introduce deallocation

Multi-Minded Combinatorial Auctions

Multi-Minded CAs

- r items $\{A, B, \dots\}$
- n agents $\{1, 2, \dots, n\}$
- express valuation for bundles
- each agent interested in at most b bundles

Multi-Minded Combinatorial Auctions

Multi-Minded CAs

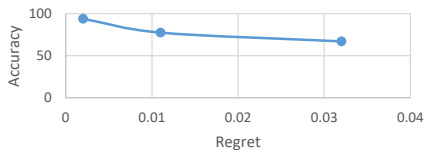
- r items $\{A, B, \dots\}$
- n agents $\{1, 2, \dots, n\}$
- express valuation for bundles
- each agent interested in at most b bundles

In Terms of the Framework

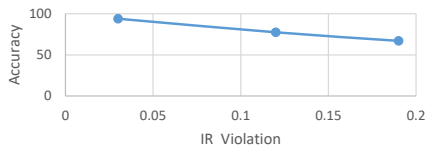
- type $\theta_i = (v_i(\emptyset), v_i(A), v_i(B), v_i(C), v_i(AB), v_i(AC), v_i(BC), v_i(ABC))$
- type profile $\theta = (\theta_1, \theta_2, \dots, \theta_n)$
- outcome $o_1 = 101$
- type distribution D with parameter to control correlation and complementarity between items

Correlation between Accuracy and Regret / IR Violation

Negative Correlation between Accuracy and Regret

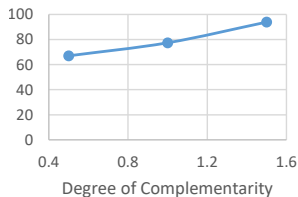


Negative Correlation between Accuracy and IR Violation

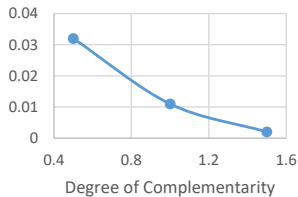


Degree of Complementarity

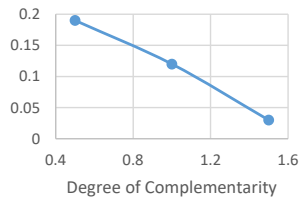
Accuracy



Regret



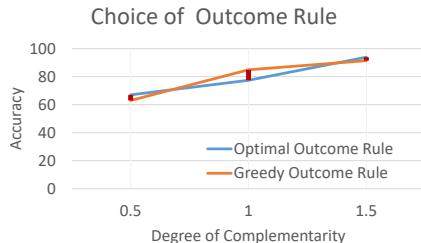
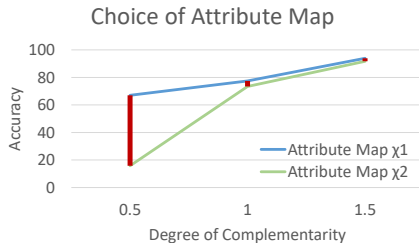
IR Violation



low degree of complementarity: $v(AB) \approx v(A) + v(B)$

high degree of complementarity: $v(AB) \gg v(A) + v(B)$

Choice of Outcome Rule and Attribute Map



Training Set Size and IR Fixes

Training Set Size: more training data leads to better results

IR Fixes:

- payment offset
- adjusting loss function
- introducing deallocation

} IR - Violation ↓ Regret ↑

Conclusion

Challenges of Classical Mechanism Design

- Analytical Complexity
- Exclusion of Mechanisms
- Computational Complexity

Conclusion

Challenges of Classical Mechanism Design

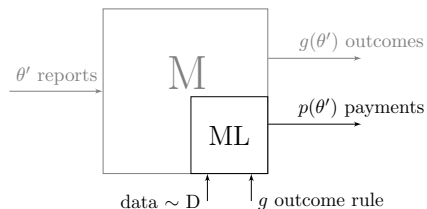
- Analytical Complexity
- Exclusion of Mechanisms
- Computational Complexity

Conclusion

- introduce new paradigm for computational mechanism design
- shown encouraging experimental results
- further directions of interest that have to be investigated in the future

Discussion - Overview

New Approach



Remarks

- train a classifier for the outcome
- use special structure of the classifier to extract a payment rule
- the better the classifier for the given outcome rule, the less incentive an agent has for not reporting truthfully