On AI, Markets and Machine Learning

David C. Parkes

Computer Science
John A. Paulson School of Engineering and Applied Sciences
Harvard University

May 11, 2017
Architecting the “outside” to make the “inside” easier, and promote good outcomes.

Designing the rules of the game to bring AIs, and AIs and people, together in useful ways.

Myriad applications: smart grid, participatory democracy, fair division, resource allocation, matching, coalition formation, ...
An AI-mediated society that strikes a balance between being fair, utilitarian, representative, market-driven, ... 

Yikes! Clear that our field needs to be interdisciplinary (political science, economic science, social psychology, ...)

And once we’ve understood all of this, we need to make our systems scale.
An AI-mediated society that strikes a balance between being fair, utilitarian, representative, market-driven, ... 

Yikes! Clear that our field needs to be interdisciplinary (political science, economic science, social psychology, ...)

And once we’ve understood all of this, we need to make our systems scale.
The Backdrop: ‘Value-aligned AI’

- An AI-mediated society that strikes a balance between being fair, utilitarian, representative, market-driven, . . .

- Yikes! Clear that our field needs to be interdisciplinary (political science, economic science, social psychology, ...)

- And once we’ve understood all of this, we need to make our systems scale.
To put a finer point on it...

There is concern about Silicon Valley “exporting its values to the world,” with programs needing to encode values.

I think our community has a response, and it’s the coupling of:

- **Strong Agents**: Representing individual preferences, individual values.

- **Normative System Design**: Aggregate inputs and make decisions in a fair, representative, and utilitarian way.
There is concern about Silicon Valley “exporting its values to the world,” with programs needing to encode values.

I think our community has a response, and it’s the coupling of:

- **Strong Agents**: Representing individual preferences, individual values.

- **Normative System Design**: Aggregate inputs and make decisions in a fair, representative, and utilitarian way.
An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations

Tuomas Sandholm

Computer Science Department
University of Massachusetts
Where I started...

An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations
Tuomas Sandholm
Computer Science Department
University of Massachusetts

Designing Conventions for Automated Negotiation
Jeffrey S. Rosenschein and Gilad Zlotkin
Where I started...

Already autonomy, normative design, and connections w/ economics!
What I liked

- Outside/inside: Design outside to promote simplicity inside, avoid “wasteful counterspeculation.”

- Precision: clearly stated design goals (norms), impossibility and possibility results.

- Apparent applicability, connection to practice.
(Aside) What I didn’t like

“Design an iterative combinatorial auction”

Regression testing by paper and pen.

Lucky break 1: Pat Harker suggested “go read Bertsekas’ work on auction algorithms.”

Lucky break 2: Bikhchandani and Ostroy’98. An LP hierarchy, allowing primal-dual formulations of CAs.

After a lot of sweat, iterative CAs as primal-dual algorithms.
( Aside ) What I didn’t like

“Design an iterative combinatorial auction”

Regression testing by paper and pen.

Lucky break 1: Pat Harker suggested “go read Bertsekas’ work on auction algorithms.”

Lucky break 2: Bikhchandani and Ostroy’98. An LP hierarchy, allowing primal-dual formulations of CAs.

After a lot of sweat, iterative CAs as primal-dual algorithms.
(Aside) What I didn’t like

“Design an iterative combinatorial auction”

Regression testing by paper and pen.

Lucky break 1: Pat Harker suggested “go read Bertsekas’ work on auction algorithms.”

Lucky break 2: Bikhchandani and Ostroy’98. An LP hierarchy, allowing primal-dual formulations of CAs.

After a lot of sweat, iterative CAs as primal-dual algorithms.
(Aside) What I didn’t like

“Design an iterative combinatorial auction”

Regression testing by paper and pen.

Lucky break 1: Pat Harker suggested “go read Bertsekas’ work on auction algorithms.”

Lucky break 2: Bikhchandani and Ostroy’98. An LP hierarchy, allowing primal-dual formulations of CAs.

After a lot of sweat, iterative CAs as primal-dual algorithms.
(Aside) What I didn’t like

“Design an iterative combinatorial auction”

Regression testing by paper and pen.

Lucky break 1: Pat Harker suggested “go read Bertsekas’ work on auction algorithms.”

Lucky break 2: Bikhchandani and Ostroy’98. An LP hierarchy, allowing primal-dual formulations of CAs.

After a lot of sweat, iterative CAs as primal-dual algorithms.
Fast forward a few years…
Rui Dong, a precocious freshman. It’s also my first week at Harvard.

We talk.

“But why is this computer science?”

To explain: let’s take the framework of mechanism design as an example of the kinds of things that we do.
“I want to do research”

- Rui Dong, a precocious freshman. It’s also my first week at Harvard.

- We talk.

- “But why is this computer science?”

- To explain: let’s take the framework of mechanism design as an example of the kinds of things that we do.
Rui Dong, a precocious freshman. It’s also my first week at Harvard.

We talk.

“But why is this computer science?”

To explain: let’s take the framework of mechanism design as an example of the kinds of things that we do.
“I want to do research”

Rui Dong, a precocious freshman. It’s also my first week at Harvard.

We talk.

“But why is this computer science?”

To explain: let’s take the framework of mechanism design as an example of the kinds of things that we do.
Mechanism Design

- $n$ agents, set $A$ of alternatives
- Agent $i$’s value $v_i(a)$, for $a \in A$, with $v_i \in V_i$.
- Let $V = V_1 \times \ldots \times V_n$
- A mechanism:
  \[ f : V \mapsto A \]
- Agents are self-interested. Properties of $f$ studied in equilibrium:
  \[ g(v_1, \ldots, v_n) = f(s_1^*(v_1), \ldots, s_n^*(v_n)) \]
Mechanism Design

- $n$ agents, set $A$ of alternatives
- Agent $i$'s value $v_i(a)$, for $a \in A$, with $v_i \in V_i$.
- Let $V = V_1 \times \ldots \times V_n$
- A mechanism:
  
  $f : V \rightarrow A$

- Agents are self-interested. Properties of $f$ studied in equilibrium:
  
  $g(v_1, \ldots, v_n) = f(s_1^*(v_1), \ldots, s_n^*(v_n))$
Mechanism Design

- $n$ agents, set $A$ of alternatives
- Agent $i$’s value $v_i(a)$, for $a \in A$, with $v_i \in V_i$.
- Let $V = V_1 \times \ldots \times V_n$
- A mechanism:

$$f : V \mapsto A$$

- Agents are self-interested. Properties of $f$ studied in equilibrium:

$$g(v_1, \ldots, v_n) = f(s^*_1(v_1), \ldots, s^*_n(v_n))$$
The magical Revelation principle

- An incentive compatible (IC) mechanism has a truthful equilibrium. Simplifies!

- The revelation principle: imposing IC is WLOG on mechanism design.
The magical Revelation principle

- An incentive compatible (IC) mechanism has a truthful equilibrium. Simplifies!

- The revelation principle: imposing IC is WLOG on mechanism design.
An incentive compatible (IC) mechanism has a truthful equilibrium. Simplifies!

The revelation principle: imposing IC is WLOG on mechanism design.
IC can be tricky to work with...

It's a global property of a function.

Suppose \( v_1 = w \). Then for mechanism \( f \) to be IC, we need:

\[
\begin{align*}
    v_1(f(w, v_{-1})) & \geq v_1(f(x, v_{-1})) \\
    v_1(f(w, v_{-1})) & \geq v_1(f(y, v_{-1})) \\
    v_1(f(w, v_{-1})) & \geq v_1(f(z, v_{-1})) \\
    v_1(f(w, v_{-1})) & \geq \cdots
\end{align*}
\]

Also need this for all agents, for all true valuations, and for all valuations of others.
Another great idea: The VCG mechanism

- Suppose agents other than 1 are truthful, let \( \hat{v}_1 \) denote report of 1.

- Define \( f \) to solve

\[
a^* = \arg \max_a \left( \hat{v}_1(a) + \sum_{j \neq 1} v_j(a) \right)
\]

- Charge agent 1 the amount \(-\sum_{j \neq 1} v_j(a^*)\). Utility:

\[
v_1(a^*) + \sum_{j \neq 1} v_j(a^*)
\]

- This provides IC! Can also modify payment, charge cost on others:

\[
\sum_{j \neq 1} v_j(a_{-1}) - \sum_{j \neq 1} v_j(a^*)
\]
Another great idea: The VCG mechanism

- Suppose agents other than 1 are truthful, let \( \hat{v}_1 \) denote report of 1.

- Define \( f \) to solve

\[
a^* = \arg \max_a (\hat{v}_1(a) + \sum_{j \neq 1} v_j(a))
\]

- Charge agent 1 the amount \(-\sum_{j \neq 1} v_j(a^*)\). Utility:

\[
v_1(a^*) + \sum_{j \neq 1} v_j(a^*)
\]

- This provides IC! Can also modify payment, charge cost on others:

\[
\sum_{j \neq 1} v_j(a_{-1}) - \sum_{j \neq 1} v_j(a^*)
\]
Another great idea: The VCG mechanism

- Suppose agents other than 1 are truthful, let $\hat{v}_1$ denote report of 1.
- Define $f$ to solve
  $$a^* = \arg \max_a (\hat{v}_1(a) + \sum_{j \neq 1} v_j(a))$$
- Charge agent 1 the amount $-\sum_{j \neq 1} v_j(a^*)$. Utility:
  $$v_1(a^*) + \sum_{j \neq 1} v_j(a^*)$$
- This provides IC! Can also modify payment, charge cost on others:
  $$\sum_{j \neq 1} v_j(a_{-1}) - \sum_{j \neq 1} v_j(a^*)$$
It’s not really that strange a mechanism...

- **Equilibrium prices**: [70, 100] (balance supply and demand)
- VCG charges cost on others: 70 - 0 = 70
- This the min equil. price. Unaffected by winner’s bid.
Equilibrium prices: [70,100] (balance supply and demand)

VCG charges cost on others: 70-0=70

This the min equil. price. Unaffected by winner’s bid.
It’s not really that strange a mechanism...

- Equilibrium prices: [70, 100] (balance supply and demand)
- VCG charges cost on others: 70-0=70
- This the min equil. price. Unaffected by winner’s bid.
What can be done in principle?
Auctioning London Bus Routes
How can we design real systems using ideas from mechanism design?

- Need to scale-up: efficient preference elicitation, decentralized computation, handle dynamics.
- Lots more research—winner determination, design of bidding languages, pricing algorithms, etc.
Can we elicit just enough information to compute the ‘right outcome’?

Modular architecture. Proxy agents query agents, continue until have correct market outcome. (With S. Lahaie, F. Constantin)
I: Elicitation via Learning

- monotone Bool functions
- Representation class $L$
- Target $h : \{0, 1\}^m \mapsto \mathbb{R}$
- Membership query: $h(x)$
- Equivalence query: $\hat{h}$?
- Learn $\text{poly}(m, \text{size}(h))$
- free disposal
- Bidding language $L$
- Target $v_i : \{0, 1\}^m \mapsto \mathbb{R}$
- Value query: $v_i(x)$
- Demand query: $(p, x)$?
- Solve $\text{poly}(n, m, \text{size}(v_1, v_n))$
I: Elicitation via Learning

- monotone Bool functions
- Representation class $L$
- Target $h : \{0, 1\}^m \rightarrow \mathbb{R}$
- Membership query: $h(x)$
- Equivalence query: $\hat{h}$?
- Learn $poly(m, \text{size}(h))$
- free disposal
- Bidding language $L$
- Target $v_i : \{0, 1\}^m \rightarrow \mathbb{R}$
- Value query: $v_i(x)$
- Demand query: $(p, x)$?
- Solve $poly(n, m, \text{size}(v_1, v_n))$
I: Elicitation via Learning

- monotone Bool functions
- Representation class $L$
- Target $h : \{0, 1\}^m \rightarrow \mathbb{R}$
- Membership query: $h(x)$
- Equivalence query: $\hat{h}$?
- Learn $\text{poly}(m, \text{size}(h))$
- free disposal
- Bidding language $L$
- Target $v_i : \{0, 1\}^m \rightarrow \mathbb{R}$
- Value query: $v_i(x)$
- Demand query: $(p, x)$?
- Solve $\text{poly}(n, m, \text{size}(v_1, v_n))$

Positive results for XOR, for polynomial language (substitutes).
II: Removing the ‘Center’

Can we distribute the computation involved in $f : V \mapsto A$?

What could possibly go wrong?
II: Removing the ‘Center’

(w/ Jeff Shneidman, Boi Faltings, Adrian Petcu.)

- Distribute computation of outcome rule and payments across agents

- Need faithfulness: bring suggested strategy (= algorithm) into an ex post Nash equilibrium

- M-DPOP: adapted DPOP, a complete and optimal distr. opt alg to make it faithful (via VCG ideas)

- Crucial idea: The “partitioning principle”. Need ‘without i’ outcome to be computed correctly whatever action of i.
II: Removing the ‘Center’

(w/ Jeff Shneidman, Boi Faltings, Adrian Petcu.)

- Distribute computation of outcome rule and payments across agents

- Need **faithfulness**: bring suggested strategy (＝ algorithm) into an ex post Nash equilibrium

- M-DPOP: adapted DPOP, a complete and optimal distr. opt alg to make it faithful (via VCG ideas)

- Crucial idea: The “partitioning principle”. Need ‘without i’ outcome to be computed correctly whatever action of i.
II: Removing the ‘Center’

(w/ Jeff Shneidman, Boi Faltings, Adrian Petcu.)

- Distribute computation of outcome rule and payments across agents
- Need **faithfulness**: bring suggested strategy (= algorithm) into an ex post Nash equilibrium
- **M-DPOP**: adapted DPOP, a complete and optimal distr. opt alg to make it faithful (via VCG ideas)

*Crucial idea*: The “partitioning principle”. Need ‘without $i$’ outcome to be computed correctly whatever action of $i$. 
II: Removing the ‘Center’

(w/ Jeff Shneidman, Boi Faltings, Adrian Petcu.)

- Distribute computation of outcome rule and payments across agents

- Need **faithfulness**: bring suggested strategy (≡ algorithm) into an ex post Nash equilibrium

- **M-DPOP**: adapted DPOP, a complete and optimal distr. opt alg to make it faithful (via VCG ideas)

- **Crucial idea**: The “partitioning principle”. Need ‘without *i*’ outcome to be computed correctly whatever action of *i*. 
I remember teaching a class in around 2004. Someone asked about a sequence of Vickrey auctions. Not IC!

Can we handle dynamic agent population, dynamically changing information?
I remember teaching a class in around 2004. Someone asked about a sequence of Vickrey auctions. Not IC!

Can we handle dynamic agent population, dynamically changing information?
With a monotone rule, can change the pricing rule. (Charge $40 in this example.) Recover IC.

In a probabilistic model, follow optimal MDP policy, and use online VCG mechanism, charge agent $i$

$$\text{value} = (V^*(s_t) - V^*(s_t, -i))$$

e.g., if $v_3 \sim U(40, 60)$ then charge $50 to each of agents 1 and 2.
With a monotone rule, can change the pricing rule. (Charge $40 in this example.) Recover IC.

In a probabilistic model, follow optimal MDP policy, and use online VCG mechanism, charge agent $i$

$$\text{value} - (V^*(s_t) - V^*(s_{t,-i}))$$

E.g., if $v_3 \sim U(40, 60)$ then charge $50 to each of agents 1 and 2.
Thus far, living in this (abstract!) world

\[ f : V \mapsto A \]

But what if the inputs are not preferences but actions, for example:

- Feedback on local places in a city
- Pick-ups on a ride-sharing platform
- Demand-response

A prototypical new design pattern is:

mechanism suggests \( x \), sets payment schedule, agents decide whether to do \( x \) or \( y \), mechanism observes \( z \), makes payment, ...
Thus far, living in this (abstract!) world

\[ f : V \leftrightarrow A \]

But what if the inputs are not preferences but actions, for example:

- Feedback on local places in a city
- Pick-ups on a ride-sharing platform
- Demand-response

A prototypical new design pattern is:

mechanism suggests \( x \), sets payment schedule, agents decide whether to do \( x \) or \( y \), mechanism observes \( z \), makes payment, ...
Thus far, living in this (abstract!) world

\[ f : V \mapsto A \]

But what if the inputs are not preferences but actions, for example:

- Feedback on local places in a city
- Pick-ups on a ride-sharing platform
- Demand-response

A prototypical new design pattern is:

mechanism suggests \( x \), sets payment schedule, agents decide whether to do \( x \) or \( y \), mechanism observes \( z \), makes payment, ...
Rock the vote

Help us crowdsourc those little details that really come in handy when deciding where to go.

Question: how can we promote effort, elicit useful information?
Can we use ‘Peer prediction’ mechanisms?

- Miller, Resnick and Zeckhauser, 2005

<table>
<thead>
<tr>
<th>peer</th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>no</td>
<td>0.1</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>score</th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>no</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Truthful reports maximize expected payment:

\[ E_{x_2|x_1}[t_1(x_1, x_2)] > E_{x_2|x_1}[t_1(x'_1, x_2)], \quad \text{for all } x_1, x'_1 \]

However: bad equilibria, need to know signal distribution.
(w/ J. Witkowski, D. Mandal, V. Shnayder, A. Agarwal, N. Shah, R. Frongillo)
Can we use ‘Peer prediction’ mechanisms?

- Miller, Resnick and Zeckhauser, 2005

<table>
<thead>
<tr>
<th>peer</th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>no</td>
<td>0.1</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>score</th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>no</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

- Truthful reports maximize expected payment:

\[
E_{x_2|x_1}[t_1(x_1, x_2)] > E_{x_2|x_1}[t_1(x'_1, x_2)], \text{ for all } x_1, x'_1
\]

- However: bad equilibria, need to know signal distribution.
  (w/ J. Witkowski, D. Mandal, V. Shnayder, A. Agarwal, N. Shah, R. Frongillo)
Can we use ‘Peer prediction’ mechanisms?

Miller, Resnick and Zeckhauser, 2005

<table>
<thead>
<tr>
<th></th>
<th>peer</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>yes 0.5 0.1</td>
<td>yes 1 0</td>
</tr>
<tr>
<td>no</td>
<td>no 0.1 0.3</td>
<td>no 0 2</td>
</tr>
</tbody>
</table>

Truthful reports maximize expected payment:

\[
E_{x_2|x_1}[t_1(x_1, x_2)] > E_{x_2|x_1}[t_1(x'_1, x_2)], \text{ for all } x_1, x'_1
\]

However: bad equilibria, need to know signal distribution.

(w/ J. Witkowski, D. Mandal, V. Shnayder, A. Agarwal, N. Shah, R. Frongillo)
We’re not all alike (surprise!)

Zooniverse (exploding stars). E. Simpson et al. (2012)

Five distinct groupings of users. Labels -1 “not supernova.” +1 “possible supernova.” +3 “likely supernova.”

- Group 1, clear in categorization of true 0s. Less sure about 1s.
- Group 2 are “extremists.” Group 3 are “pessimists.” Group 4 are “optimists.” Group 5 are “non-committals.”
Robust Peer Prediction

(w/ A. Agarwal, D. Mandal and N. Shah.)

Multi-task

Learn the correlation structure, gives scoring scheme:

\[
\text{sgn}(\Delta) = \begin{pmatrix}
+ & + & - \\
+ & + & - \\
- & - & + 
\end{pmatrix};
\text{score} = \begin{pmatrix}
1 & 1 & 0 \\
1 & 1 & 0 \\
0 & 0 & 1 
\end{pmatrix}
\]

Do this via clustering, and then estimate pairwise cluster correlations. Incentive aligned!
Robust Peer Prediction

(w/ A. Agarwal, D. Mandal and N. Shah.)

- Multi-task

- Learn the correlation structure, gives scoring scheme:

\[
\text{sgn}(\Delta) = \begin{pmatrix}
+ & + & - \\
+ & + & - \\
- & - & +
\end{pmatrix}; \quad \text{score} = \begin{pmatrix}
1 & 1 & 0 \\
1 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

- Do this via clustering, and then estimate pairwise cluster correlations. Incentive aligned!
Robust Peer Prediction

(w/ A. Agarwal, D. Mandal and N. Shah.)

- Multi-task

- Learn the correlation structure, gives scoring scheme:

  \[
  \text{sgn}(\Delta) = \begin{pmatrix}
  + & + & - \\
  + & + & - \\
  - & - & + 
\end{pmatrix};
  \text{score} = \begin{pmatrix}
  1 & 1 & 0 \\
  1 & 1 & 0 \\
  0 & 0 & 1
\end{pmatrix}
  \]

- Do this via clustering, and then estimate pairwise cluster correlations. Incentive aligned!
Eliciting Actions II: Ride-sharing platforms

Uber:
Market Oriented Programming Strikes Back!

(with Fei Fang and Hongyao Ma)

- Drivers and passengers are customers. Both ‘at will.’
- Can only optimize by suggesting matches, prices.
- Market-based optimization in practice (this time with real money!)

A challenge has been lack of smoothness of prices.

Can we design matching methods, and fair prices, so that drivers always want to accept matched trips?
Market Oriented Programming Strikes Back!

(with Fei Fang and Hongyao Ma)

- Drivers and passengers are customers. Both ‘at will.’
- Can only optimize by suggesting matches, prices.
- Market-based optimization in practice (this time with real money!)

A challenge has been lack of smoothness of prices.

*Can we design matching methods, and fair prices, so that drivers always want to accept matched trips?*
Drivers and passengers are customers. Both ‘at will.’

Can only optimize by suggesting matches, prices.

Market-based optimization in practice (this time with real money!)

A challenge has been lack of smoothness of prices.

Can we design matching methods, and fair prices, so that drivers always want to accept matched trips?
Voluntary participation

Efficiency. Note externality. A trip from $A$ to $B$ has a side effect. The car is now at $B$!

Fairness: any two drivers with the same (location,time) pair should have same utility

In particular, we suggest the following interpretation of ‘smooth prices’:

Smoothness $\equiv$ equilibrium prices (in time and space)

We obtain anonymous, time-based trip prices. Come full circle, back to primal/dual formulations...
Voluntary participation

Efficiency. Note externality. A trip from $A$ to $B$ has a side effect. The car is now at $B$!

Fairness: any two drivers with the same (location, time) pair should have same utility

In particular, we suggest the following interpretation of ‘smooth prices’:

$$\text{Smoothness} \equiv \text{equilibrium prices (in time and space)}$$

We obtain anonymous, time-based trip prices. Come full circle, back to primal/dual formulations...
Towards Closing: History of Optimal Auctions

W. Vickrey
Second Price Auction

1961

R. B. Myerson
Optimal Single-item Auction

1981

Rochet, 1987

1987

Hart & Nisan, 2012
Cai et al. 2012a, b
Daskalakis et al. 2012
Giannakopoulos & Koutsoupias, 2015
Daskalakis et al. 2016
Yao, 2016
Cai & Zhao, 2017

2012-2017

Optimal auction for 2 bidders and 2 items still unknown!
Can Deep Learning help?

- Renaissance in neural networks ("deep learning") for nonlinear function approximation.
- Robust tool chains (e.g., TensorFlow, GPUs).
- Beginning to get attention in economics/EconCS (e.g., Hartford, Wright, L-B’16a; Hartford, Lewis, L-B, Taddy’16b)
Automated (Data-driven) Mechanism Design

AMD first proposed by Conitzer and Sandholm, 2002.

- Given: distribution $v \sim F^n$
- Learn an optimal auction $(g, p)$
- Use neural networks for non-linear function approximation

\[
\begin{pmatrix}
(v^1) \\
(v^2) \\
. \\
. \\
. \\
(v^D)
\end{pmatrix}
\quad \rightarrow \quad \text{Training}
\quad \begin{align*}
\text{Tune weights to} \\
\text{maximize revenue s.t.} \\
\text{incentive constraints}
\end{align*}
\quad \rightarrow
\]

(w/ H. Narasimhan, S. Agarwal, Z. Feng, P. Dütting.)
Characterization-based architectures (MyersonNet, RochetNet), as well as ‘agnostic networks’:

(a) Allocation network $g$

(b) Payment network $p$
Example Allocation Rule (One buyer, multi-item)

(w/ Z. Feng, H. Narasimhan, P. Dütting.)

- Recover existing optimal auctions.
- Extend to new settings (e.g., 2 buyers, 2 items, continuous values.)

<table>
<thead>
<tr>
<th></th>
<th>Symmetric Unf</th>
<th></th>
<th>Asymmetric Unf</th>
<th></th>
<th>Unit-demand</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Revenue</td>
<td>Regret</td>
<td>Revenue</td>
<td>Regret</td>
<td>Revenue</td>
</tr>
<tr>
<td>RochetNet</td>
<td>0.548</td>
<td>0.001</td>
<td>9.65</td>
<td>0.0</td>
<td>2.12</td>
</tr>
<tr>
<td>Optimal</td>
<td>0.550</td>
<td>-</td>
<td>9.68</td>
<td>-</td>
<td>2.13</td>
</tr>
</tbody>
</table>
Looking ahead

- We’re at a turning point for multi-agent systems. Many systems will need to be designed.

- The promise of **agent-mediated electronic commerce** is about to be realized. *Machina economica* (w/ M. Wellman).

- Crucial for value-aligned AI will be
  - **strong agents**, that can effectively represent individuals’ preferences and values
  - **principled, normative design**, that makes good tradeoffs

- We will need to keep aware of other disciplines, collaborate.
Mentors

- Lyle Ungar
- Michael Wellman
- Barbara Grosz
[thankyou!]
All of my Collaborators

[thankyou!]
[you’re the best!]

- Rediet Abebe
- Hossein Azari
- Ruggiero Cavallo
- David Chen
- Florin Constantin
- Jacomo Corbo
- Zhe Feng
- Pavithra Harsha
- Shaili Jain
- Adam Juda
- Laura Kang
- Sebastien Lahaie
- John Lai
- Ben Lubin
- James Zou

- Hongyao Ma
- Debmalya Mandal
- Jolie Martin
- Katy Milkman
- Chaki Ng
- Malvika Rao
- Sven Seuken
- Victor Shnayder
- Jeffrey Shneidman
- Christopher Thorpe
- Panos Toulis
- Paul Tylkin
- Jens Witkowski
- Jason Woodard
- Haoqi Zhang

**Postdocs**

- Tanmoy Chakraborty
- Fei Fang
- Felix Fischer
- Ian Kash
- Reshef Meir
- Harikrishna Narasimhan
- Ariel Procaccia
- Goran Radanovic
- Lior Seeman
- Nisarg Shah
- Lirong Xia


Thank you