Mechanism Design through Statistical Machine Learning: Part II (Social Choice and Matching)

David C. Parkes

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

January 13, 2016

Mechanism Design without Money

The use of money is *not natural* in many multi-agent settings:

- *Matching* students to high schools, doctors to hospitals
- *Choosing* a location for a new firestation
- *Assigning* volunteers to evening shifts at a childcare co-op
- Meeting *scheduling*

Agents have *preference order* $a \succ_i a'$ on *alternatives* $a \in A$.

Example desiderata:

- Pareto-optimality, envy-free, non-dictatorial.

Mechanism Design without Money

The use of money is *not natural* in many multi-agent settings:

- *Matching* students to high schools, doctors to hospitals
- *Choosing* a location for a new firestation
- *Assigning* volunteers to evening shifts at a childcare co-op
- Meeting *scheduling*

Agents have *preference order* $a \succ_i a'$ on *alternatives* $a \in A$.

Example desiderata:

- Pareto-optimality, envy-free, non-dictatorial.

Mechanism Design without Money

The use of money is *not natural* in many multi-agent settings:

- *Matching* students to high schools, doctors to hospitals
- *Choosing* a location for a new firestation
- *Assigning* volunteers to evening shifts at a childcare co-op
- Meeting *scheduling*

Agents have *preference order* $a \succ_i a'$ on *alternatives* $a \in A$.

Example desiderata:

- Pareto-optimality, envy-free, non-dictatorial.

The Basic Model

- Agent i has a *preference order* $\succ_i \in P$, *preference profile* $\succ = (\succ_1, \ldots, \succ_n)$, sampled $\succ \sim D$
- Alternatives A. *Outcome rule* $f : P^n \mapsto A$
- *Incentive compatibility*. Given rule f, want

$$f(\succ_i, \hat{\succ}_{-i}) \succeq_i f(\hat{\succ}_i, \hat{\succ}_{-i}), \text{ for all } \succ_i, \text{ all } \hat{\succ}_i, \text{ all } \hat{\succ}_{-i}$$

- *Examples of IC Mechanisms*:
  - For assignment: Random serial dictatorship, top-trading cycles, ...
  - For social choice: Median mechanism:

The Basic Model

- Agent i has a *preference order* $\succ_i \in P$, *preference profile* $\succ = (\succ_1, \ldots, \succ_n)$, sampled $\succ \sim D$
- Alternatives A. *Outcome rule* $f : P^n \mapsto A$
- *Incentive compatibility*. Given rule f, want

$$f(\succ_i, \hat{\succ}_{-i}) \ \succeq_i \ f(\hat{\succ}_i, \hat{\succ}_{-i}), \text{ for all } \succ_i, \text{ all } \hat{\succ}_i, \text{ all } \hat{\succ}_{-i}$$

- *Examples of IC Mechanisms*:
  - For assignment: Random serial dictatorship, top-trading cycles, ...
  - For social choice: Median mechanism:

The Basic Model

- Agent i has a *preference order* $\succ_i \in P$, *preference profile* $\succ = (\succ_1, \ldots, \succ_n)$, sampled $\succ \sim D$
- Alternatives A. *Outcome rule* $f : P^n \mapsto A$
- *Incentive compatibility*. Given rule f, want

$$f(\succ_i, \hat{\succ}_{-i}) \;\succeq_i\; f(\hat{\succ}_i, \hat{\succ}_{-i}), \text{ for all } \succ_i, \text{ all } \hat{\succ}_i, \text{ all } \hat{\succ}_{-i}$$

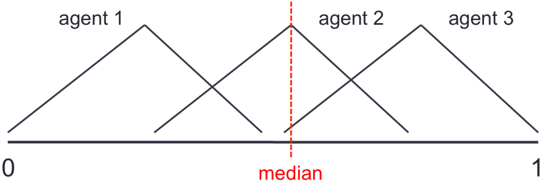
- *Examples of IC Mechanisms*:
  - For assignment: Random serial dictatorship, top-trading cycles, . . .
  - For social choice: Median mechanism:

The Basic Model

- Agent i has a *preference order* $\succ_i \in P$, *preference profile* $\succ = (\succ_1, \ldots, \succ_n)$, sampled $\succ \sim D$
- Alternatives A. *Outcome rule* $f : P^n \mapsto A$
- *Incentive compatibility*. Given rule f, want

$$f(\succ_i, \hat{\succ}_{-i}) \ \succeq_i \ f(\hat{\succ}_i, \hat{\succ}_{-i}), \text{ for all } \succ_i, \text{ all } \hat{\succ}_i, \text{ all } \hat{\succ}_{-i}$$

- *Examples of IC Mechanisms*:
  - For assignment: Random serial dictatorship, top-trading cycles, . . .
  - For social choice: Median mechanism:



State of the art (MD without money)

- Gibbard-Satterthwaite *impossibility result*
- *Characterization results* for specific problems
  - Often *axiomatic*, e.g., class of IC, onto, neutral rules for single-peaked setting is *generalized median mechanisms*
  - Impossibility results as well
- *No general design theory*, results for specific preference domains, axiomatic rather than optimization-based

State of the art (MD without money)

- Gibbard-Satterthwaite *impossibility result*
- *Characterization results* for specific problems
  - Often *axiomatic*, e.g., class of IC, onto, neutral rules for single-peaked setting is *generalized median mechanisms*
  - Impossibility results as well
- *No general design theory*, results for specific preference domains, axiomatic rather than optimization-based

State of the art (MD without money)

- Gibbard-Satterthwaite *impossibility result*
- *Characterization results* for specific problems
  - Often *axiomatic*, e.g., class of IC, onto, neutral rules for single-peaked setting is *generalized median mechanisms*
  - Impossibility results as well
- *No general design theory*, results for specific preference domains, axiomatic rather than optimization-based

The Machine Learning Framework

(Part 1) *Mechanism design with money*:

- Given *outcome rule* $f : X^n \mapsto Y$
- Want to *learn payment rule* t_w such that mechanism (f, t_w) is *maximally-IC*.

(Part 2) *Mechanism design without money*:

- Given *target outcome rule* $f : P^n \mapsto A$ (via training examples)
- Want to *learn outcome rule* f_w that is *IC* and solves

$$\min_{f_w \in \mathcal{F}_{ic}} \mathbb{E}_{\succ \sim D}[\ell(f(\succ), f_w(\succ), \succ)],$$

for IC rules $\mathcal{F}_{ic}$ and *loss function* $\ell(a, a', \succ) \geq 0$.

The Machine Learning Framework

(Part 1) *Mechanism design with money*:

- Given *outcome rule* $f : X^n \mapsto Y$
- Want to *learn payment rule* t_w such that mechanism (f, t_w) is *maximally-IC*.

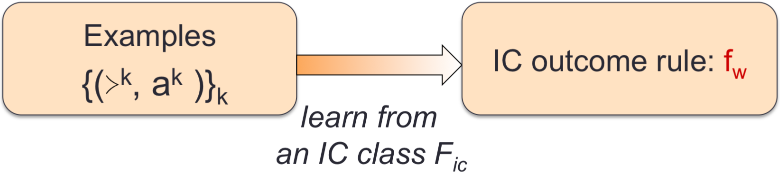
(Part 2) *Mechanism design without money*:

- Given *target outcome rule* $f : P^n \mapsto A$ (via training examples)
- Want to *learn outcome rule* f_w that is *IC* and solves

$$\min_{f_w \in \mathcal{F}_{ic}} \mathbb{E}_{\succ \sim D}[\ell(f(\succ), f_w(\succ), \succ)],$$

for IC rules $\mathcal{F}_{ic}$ and *loss function* $\ell(a, a', \succ) \geq 0$.

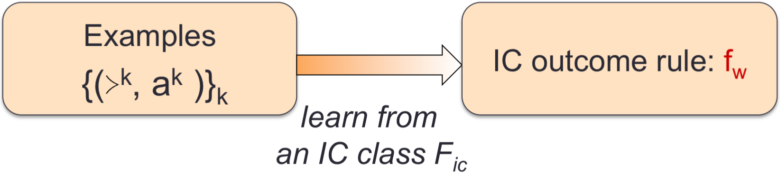
Overview: Learning Mechanisms without Money



Related work:

- Procaccia et al.'09: Learning non-IC voting rules
- Conitzer and Sandholm '02, Guo and Conitzer'10: Search through (parameterized) space of feasible mechanisms.

Overview: Learning Mechanisms without Money



Related work:

- Procaccia et al.'09: Learning non-IC voting rules
- Conitzer and Sandholm '02, Guo and Conitzer'10: Search through (parameterized) space of feasible mechanisms.

Outline

Configuration problems:

1. Single-Peaked Social choice
2. One-sided matching (assignment)
3. Stable,two-sided matching

Closing: towards a general framework (back to prices!), and a direction for 'with money' design.

Outline

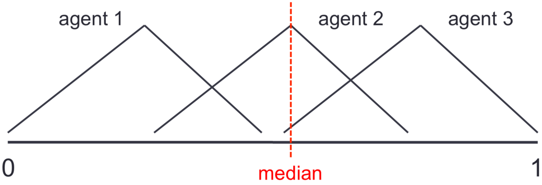
Configuration problems:

1. Single-Peaked Social choice
2. One-sided matching (assignment)
3. Stable,two-sided matching

Closing: towards a general framework (back to prices!), and a direction for ‘with money’ design.

Setting 1: Single-Peaked Social Choice

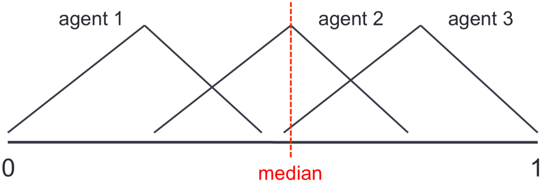
- Alternatives A, *preference order* $\succ_i$, with *peak* $o_i \in A$.
- Alternative a has *position* $z_a \in [0, 1]$.
  - $a < a'$ indicates $z_a < z_{a'}$.



- $\mathcal{F}_{ic}$: class of *weighted generalized median rules*, generalize GM rules

Setting 1: Single-Peaked Social Choice

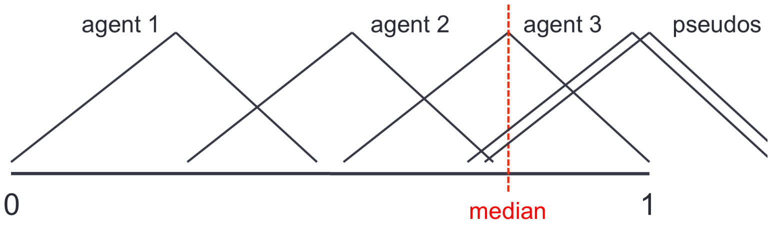
- Alternatives A, *preference order* $\succ_i$, with *peak* $o_i \in A$.
- Alternative a has *position* $z_a \in [0, 1]$.
  - $a < a'$ indicates $z_a < z_{a'}$.



- $\mathcal{F}_{ic}$: class of *weighted generalized median rules*, generalize GM rules

Generalized Median Rule

(Moulin'80)



$$rank(\succ, a) = \sum_i \mathbb{1}(o_i \leq a) + \sum_{i'} \mathbb{1}(o_{i'} \leq a)$$

$$f(\succ) = \arg\min_a \{z_a \mid rank(\succ, a) \geq \frac{(n + n_p)}{2}\}$$

Weighted Generalized-Median Rule

- *Weights* $\alpha \in \mathbb{R}^n_{\geq 0}$ and $\beta \in \mathbb{R}^m_{\geq 0}$. Define:

$$rank_w(\succ, a) = \sum_i \alpha_i \cdot \mathbb{1}(o_i \leq a) + \sum_j \beta_j \cdot \mathbb{1}(j \leq a)$$

- Given *threshold* $t \geq 0$, select:

$$f_w(\succ) = \arg\min_a \{z_a \mid rank_w(\succ, a) \geq t\}$$

Example

3 agents $\alpha = (1, 2, 3)$, 5 choices $\beta = (1.5, 0.5, 0, 0, 1.5)$, and $t = 3$. Agent peaks (a, c, d). Ranks: $2.5, 3, 5, \ldots$; $f_w(\succ) = b$.

Weighted Generalized-Median Rule

- *Weights* $\alpha \in \mathbb{R}^n_{\geq 0}$ and $\beta \in \mathbb{R}^m_{\geq 0}$. Define:

$$rank_w(\succ, a) = \sum_i \alpha_i \cdot \mathbb{1}(o_i \leq a) + \sum_j \beta_j \cdot \mathbb{1}(j \leq a)$$

- Given *threshold* $t \geq 0$, select:

$$f_w(\succ) = \arg\min_a \{z_a \mid rank_w(\succ, a) \geq t\}$$

Example

3 agents $\alpha = (1, 2, 3)$, 5 choices $\beta = (1.5, 0.5, 0, 0, 1.5)$, and $t = 3$. Agent peaks (a, c, d). Ranks: $2.5, 3, 5, \ldots$; $f_w(\succ) = b$.

Learning the optimal GWM rule

1. $w \in \mathbb{R}^{n+m+1}$. Adopt (continuous) *discriminant function*:

$$H_w(\succ, a) = -(rank_w(\succ, a) - t)^2.$$

   Not IC; but use learned w to instantiate a WGM rule.

2. Incorporate *loss function* $\ell(a, a', \succ) = |z_{a'} - z_a|$, via a continuous *surrogate* ℓ', obtaining *training problem*:

$$\min_w \frac{1}{2} w^\top w + C \sum_k \ell'(a^k, f_w(\succ^k), \succ^k).$$

   Need not be convex, solve via gradient-descent, restarts.

Learning the optimal GWM rule

1. $w \in \mathbb{R}^{n+m+1}$. Adopt (continuous) *discriminant function*:

$$H_w(\succ, a) = -(rank_w(\succ, a) - t)^2.$$

   Not IC; but use learned w to instantiate a WGM rule.

2. Incorporate *loss function* $\ell(a, a', \succ) = |z_{a'} - z_a|$, via a continuous *surrogate* ℓ', obtaining *training problem*:

$$\min_w \frac{1}{2} w^\top w + C \sum_k \ell'(a^k, f_w(\succ^k), \succ^k).$$

   Need not be convex, solve via gradient-descent, restarts.

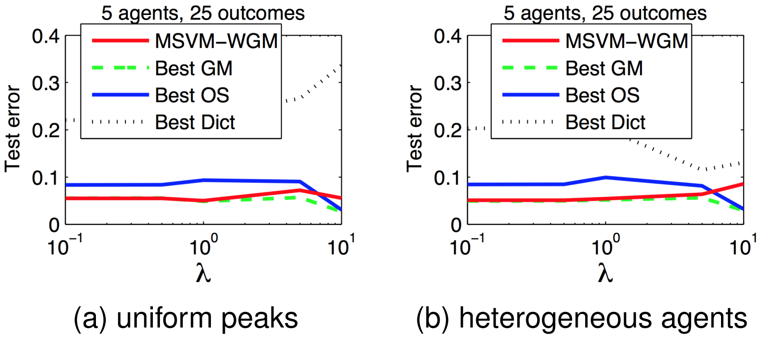
Results: Single-Peaked Social Choice

- *Target outcome rule*: priorities $C(z) = e^{-\lambda z}$ for $\lambda \geq 0$, and $f(\succ) \in \arg\min_a \sum_i C(z_{o_i}) \cdot (z_a - z_{o_i})^2$
- Compare with best *GM rule*, best *order-statistic rule*, and best *dictatorial rule*. Loss is distance from target.

(a) uniform peaks (b) heterogeneous agents

Results: Single-Peaked Social Choice

- *Target outcome rule*: priorities $C(z) = e^{-\lambda z}$ for $\lambda \geq 0$, and $f(\succ) \in \arg\min_a \sum_i C(z_{o_i}) \cdot (z_a - z_{o_i})^2$
- Compare with best *GM rule*, best *order-statistic rule*, and best *dictatorial rule*. Loss is distance from target.



Comparing run time

- $m = 25$, time in minutes.
- - indicates did not complete in 24 hours

	$n = 5$	$n = 7$	$n = 9$	$n = 11$
MSVM-WGM	27.65	29.97	29.95	30.00
Best GM	0.33	9.33	168.27	-
Best Percentile	3E-5	5E-5	6E-5	7E-5

Setting 2: One-Sided Matching (Assignment)

- Alternative $a \in A$ assigns items $\{1, \ldots, n\}$ to agents
- *Preferences* $\succ_i$ on items
- $\mathcal{F}_{ic}$: class of *Adaptive Serial-Dictator rules*

Example

Serial dictator rule. 3 agents, 3 items. Priority order 1 > 2 > 3. Reports:

$$\succ_1: b\,c\,a \quad \succ_2: c\,a\,b \quad \succ_3: b\,a\,c$$

1 gets b, 2 gets c, 3 gets a.

Setting 2: One-Sided Matching (Assignment)

- Alternative $a \in A$ assigns items $\{1, \ldots, n\}$ to agents
- *Preferences* $\succ_i$ on items
- $\mathcal{F}_{ic}$: class of *Adaptive Serial-Dictator rules*

Example

Serial dictator rule. 3 agents, 3 items. Priority order $1 > 2 > 3$. Reports:

$$\succ_1 : b\,c\,a \quad \succ_2 : c\,a\,b \quad \succ_3 : b\,a\,c$$

1 gets b, 2 gets c, 3 gets a.

Adaptive Serial-Dictator rules

Adaptive Serial-Dictator (Bade 2015, Pápai 2001):

- Priorities determined adaptively based on current assignment
- Use a *priority tree*. Start at root. Node specifies highest-priority agent, next node depends on selected item.

Tree with *one-level* of adaptation:

Example

3 agents, 3 items. Reports:

$\succ_1$: bca $\succ_2$: cab $\succ_3$: bac

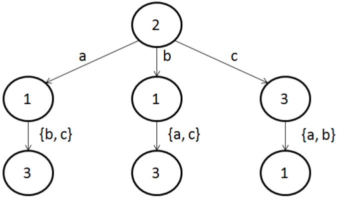
2 gets c, 3 gets b, and 1 gets a.

Adaptive Serial-Dictator rules

Adaptive Serial-Dictator (Bade 2015, Pápai 2001):

- Priorities determined adaptively based on current assignment
- Use a *priority tree*. Start at root. Node specifies highest-priority agent, next node depends on selected item.

Tree with *one-level* of adaptation:



Example

3 agents, 3 items. Reports:

$\succ_1$: bca $\succ_2$: cab $\succ_3$: bac

2 gets c, 3 gets b, and 1 gets a.

Learning Adaptive SD Rules

- Challenge: combinatorially large number of priority trees
- Use a *greedy* approach:
  - *Tree-splitting step*: assign the agent who is top-priority in optimal SD at subproblem rooted at a node
  - Branch on each item, and *recurse*.
  - *Stop at a desired level*; adopt optimal SD for rest of economy.

Learning Adaptive SD Rules

- Challenge: combinatorially large number of priority trees
- Use a *greedy* approach:
  - *Tree-splitting step*: assign the agent who is top-priority in optimal SD at subproblem rooted at a node
  - Branch on each item, and *recurse*.
  - *Stop at a desired level*; adopt optimal SD for rest of economy.

Results: One-sided Matching (Assignment)

- *Target outcome rule*: Hungarian assignment, with higher obj. value to agents who prefer particular items.
- *Loss function*: total absolute change in rank.
- Vary *correlation parameter* δ (higher, more concentration.)

Results: One-sided Matching (Assignment)

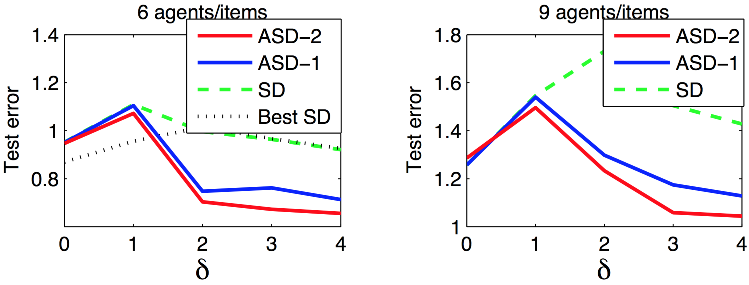
- *Target outcome rule*: Hungarian assignment, with higher obj. value to agents who prefer particular items.
- *Loss function*: total absolute change in rank.
- Vary *correlation parameter* δ (higher, more concentration.)

Results: One-sided Matching (Assignment)

- *Target outcome rule*: Hungarian assignment, with higher obj. value to agents who prefer particular items.
- *Loss function*: total absolute change in rank.
- Vary *correlation parameter* δ (higher, more concentration.)

Results: One-sided Matching (Assignment)

- *Target outcome rule*: Hungarian assignment, with higher obj. value to agents who prefer particular items.
- *Loss function*: total absolute change in rank.
- Vary *correlation parameter* δ (higher, more concentration.)



Setting 3: Stable, Two-Sided Matching

- *Bipartite graph* I and J. *Alternative* a defines a *matching*.
  - For $i \in I$: preference order $\succ_i$ on J.
  - For $j \in J$: preference order $\succ_j$ on I
  - For example, medical residency matching.
- Focus on *Stability*, not IC.

Example

Let $D = \{d_1, d_2, d_3\}$ and $H = \{h_1, h_2, h_3\}$. Consider the following:

$$\succ_{d_1}: \; h_2 \; h_1 \; h_3 \qquad \succ_{h_1}: \; d_1 \; d_2 \; d_3$$
$$\succ_{d_2}: \; h_1 \; h_2 \; h_3 \qquad \succ_{h_2}: \; d_3 \; d_2 \; d_1$$
$$\succ_{d_3}: \; h_2 \; h_3 \; h_1 \qquad \succ_{h_3}: \; d_1 \; d_3 \; d_2$$

Matching $((d_1, h_1), (d_2, h_3), (d_3, h_2))$ is *stable*.
Matching $((d_1, h_1), (d_2, h_2), (d_3, h_3))$ is *unstable*. (d_3, h_2) blocking.

Setting 3: Stable, Two-Sided Matching

- *Bipartite graph* I and J. *Alternative* a defines a *matching*.
  - For $i \in I$: preference order $\succ_i$ on J.
  - For $j \in J$: preference order $\succ_j$ on I
  - For example, medical residency matching.
- Focus on *Stability*, not IC.

Example

Let $D = \{d_1, d_2, d_3\}$ and $H = \{h_1, h_2, h_3\}$. Consider the following:

$$\succ_{d_1}: \; h_2 \; h_1 \; h_3 \qquad \succ_{h_1}: \; d_1 \; d_2 \; d_3$$
$$\succ_{d_2}: \; h_1 \; h_2 \; h_3 \qquad \succ_{h_2}: \; d_3 \; d_2 \; d_1$$
$$\succ_{d_3}: \; h_2 \; h_3 \; h_1 \qquad \succ_{h_3}: \; d_1 \; d_3 \; d_2$$

Matching $((d_1, h_1), (d_2, h_3), (d_3, h_2))$ is *stable*.
Matching $((d_1, h_1), (d_2, h_2), (d_3, h_3))$ is *unstable*. (d_3, h_2) blocking.

Stable Matching Mechanisms

Deferred Acceptance (Gale and Shapley'62):

- Doctors propose to hospitals, hospitals hold onto best offer so far, and doctors move down their list.
- *Stable*.

Weighted LP polytope (Roth et al.'93):

- Matchings are extreme points in *polytope* $\mathcal{P}(\succ)$
- Given obj. coeff. $\lambda(\succ) \in \mathbb{R}^{n \times n}$, can solve

$$\max_{a \in \mathcal{P}(\succ)} \sum_i \sum_j \lambda_{ij}(\succ) \cdot a_{ij}$$

- Use *weights* $w = (\alpha, \beta, \gamma)$ to define:

$$\lambda_{ij}(\succ) = \alpha_{ij} \cdot rank_i(h_j) + \beta_{ij} \cdot rank_j(d_i) + \gamma_{ij}.$$

Stable Matching Mechanisms

Deferred Acceptance (Gale and Shapley'62):

- Doctors propose to hospitals, hospitals hold onto best offer so far, and doctors move down their list.
- *Stable*.

Weighted LP polytope (Roth et al.'93):

- Matchings are extreme points in *polytope* $\mathcal{P}(\succ)$
- Given obj. coeff. $\lambda(\succ) \in \mathbb{R}^{n \times n}$, can solve

$$\max_{a \in \mathcal{P}(\succ)} \sum_i \sum_j \lambda_{ij}(\succ) \cdot a_{ij}$$

- Use *weights* $w = (\alpha, \beta, \gamma)$ to define:

$$\lambda_{ij}(\succ) = \alpha_{ij} \cdot rank_i(h_j) + \beta_{ij} \cdot rank_j(d_i) + \gamma_{ij}.$$

Stable Matching Mechanisms

Deferred Acceptance (Gale and Shapley'62):

- Doctors propose to hospitals, hospitals hold onto best offer so far, and doctors move down their list.
- *Stable*.

Weighted LP polytope (Roth et al.'93):

- Matchings are extreme points in *polytope* $\mathcal{P}(\succ)$
- Given obj. coeff. $\lambda(\succ) \in \mathbb{R}^{n \times n}$, can solve

$$\max_{\alpha \in \mathcal{P}(\succ)} \sum_i \sum_j \lambda_{ij}(\succ) \cdot \alpha_{ij}$$

- Use *weights* $w = (\alpha, \beta, \gamma)$ to define:

$$\lambda_{ij}(\succ) = \alpha_{ij} \cdot rank_i(h_j) + \beta_{ij} \cdot rank_j(d_i) + \gamma_{ij}.$$

Stable Matching Mechanisms

Deferred Acceptance (Gale and Shapley'62):

- Doctors propose to hospitals, hospitals hold onto best offer so far, and doctors move down their list.
- *Stable*.

Weighted LP polytope (Roth et al.'93):

- Matchings are extreme points in *polytope* $\mathcal{P}(\succ)$
- Given obj. coeff. $\lambda(\succ) \in \mathbb{R}^{n \times n}$, can solve

$$\max_{a \in \mathcal{P}(\succ)} \sum_i \sum_j \lambda_{ij}(\succ) \cdot a_{ij}$$

- Use *weights* $w = (\alpha, \beta, \gamma)$ to define:

$$\lambda_{ij}(\succ) = \alpha_{ij} \cdot rank_i(h_j) + \beta_{ij} \cdot rank_j(d_i) + \gamma_{ij}.$$

Learning Stable, Weighted-Polytope Rules

1. $w \in \mathbb{R}^{3(n \times n)}$. Learned hypothesis:

$$f_w \in \arg\max_{a \in \mathcal{P}(\succ)} H_w(\succ, a),$$

with *discriminant* $H_w(\succ, a) = \sum_i \sum_j \lambda_{ij}(\succ) a_{ij}$.

2. Incorporate 0-1 loss via a continuous *surrogate* ℓ', obtain *training problem*:

$$\min_w \frac{1}{2} w^\top w + C \sum_k \ell'(a^k, f_w(\succ^k), \succ^k)$$

Convex problem.

Learning Stable, Weighted-Polytope Rules

1. $w \in \mathbb{R}^{3(n\times n)}$. Learned hypothesis:

$$f_w \in \arg\max_{\alpha \in \mathcal{P}(\succ)} H_w(\succ, \alpha),$$

with *discriminant* $H_w(\succ, \alpha) = \sum_i \sum_j \lambda_{ij}(\succ)\alpha_{ij}$.

2. Incorporate 0-1 loss via a continuous *surrogate* ℓ', obtain *training problem*:

$$\min_w \frac{1}{2} w^\top w + C \sum_k \ell'(\alpha^k, f_w(\succ^k), \succ^k)$$

Convex problem.

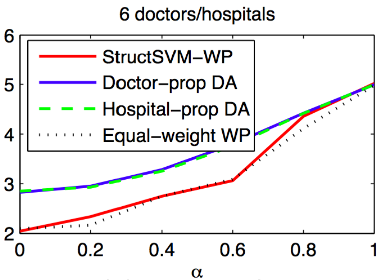
Results: Stable, two-sided matching

- *Target outcome rule*: Weighted, Hungarian assignment:
  - symmetric, equal weight to all
  - asymmetric, pref. to some doctors, hospitals
- Vary *corr. param.* α (higher, more concentration of prefs). 0-1 loss function.

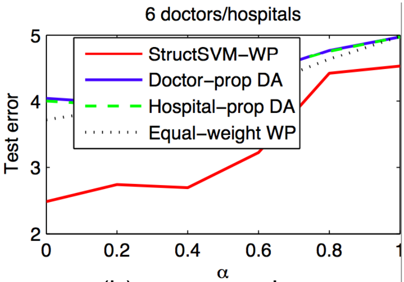
(a) symmetric (b) asymmetric

Results: Stable, two-sided matching

- *Target outcome rule*: Weighted, Hungarian assignment:
  - symmetric, equal weight to all
  - asymmetric, pref. to some doctors, hospitals
- Vary *corr. param.* α (higher, more concentration of prefs). 0-1 loss function.



(a) symmetric



(b) asymmetric

Back to Prices (and IC)

Goal: Given *training examples* $\{(\succ^k, a^k)\}_k$, and *loss function* $\ell(a, a', \succ)$, solve

$$\min_{f_w \in \mathcal{F}_{ic}} \mathbb{E}_{\succ \sim D}[\ell(f(\succ), f_w(\succ), \succ)]$$

Theorem 1

A rule f is IC if and only if, for *fixed budget* $b_i = 1$ (all i):

- *Agent-independence*: there are *virtual prices* $t_i(\hat{\succ}_{-i}, a)$
- *No-regret*: Let $A_i = \{a : t_i(\hat{\succ}_{-i}, a) \leq b_i\}$.

$$\forall i: \quad f(\hat{\succ}) \in top_i(A_i, \hat{\succ}_i)$$

Back to Prices (and IC)

Goal: Given *training examples* $\{(\succ^k, a^k)\}_k$, and *loss function* $\ell(a, a', \succ)$, solve

$$\min_{f_w \in \mathcal{F}_{ic}} \mathbb{E}_{\succ \sim D}[\ell(f(\succ), f_w(\succ), \succ)]$$

Theorem 1

A rule f is IC if and only if, for *fixed budget* $b_i = 1$ (all i):

- *Agent-independence*: there are *virtual prices* $t_i(\hat{\succ}_{-i}, a)$
- *No-regret*: Let $A_i = \{a : t_i(\hat{\succ}_{-i}, a) \leq b_i\}$.

$$\forall i : \quad f(\hat{\succ}) \in top_i(A_i, \hat{\succ}_i)$$

Back to Prices (and IC)

Goal: Given *training examples* $\{(\succ^k, a^k)\}_k$, and *loss function* $\ell(a, a', \succ)$, solve

$$\min_{f_w \in \mathcal{F}_{ic}} \mathbb{E}_{\succ \sim D}[\ell(f(\succ), f_w(\succ), \succ)]$$

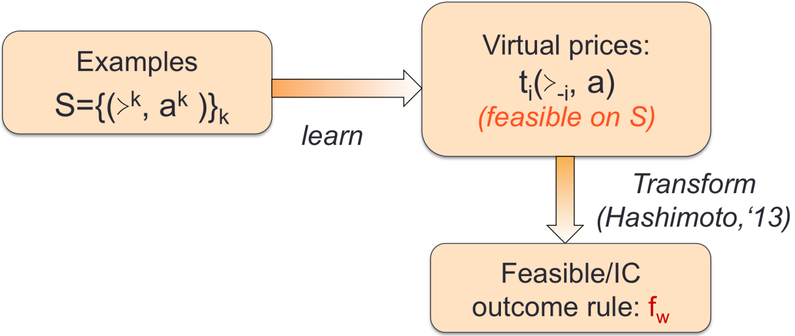
Theorem 1

A rule f is IC if and only if, for *fixed budget* $b_i = 1$ (all i):

- *Agent-independence*: there are *virtual prices* $t_i(\hat{\succ}_{-i}, a)$
- *No-regret*: Let $A_i = \{a : t_i(\hat{\succ}_{-i}, a) \leq b_i\}$.

$$\forall i: \quad f(\hat{\succ}) \in top_i(A_i, \hat{\succ}_i)$$

Overview: Learning Mechanisms via Virtual Prices



General Framework (Virtual Prices)

- Hypothesis represented by *virtual prices* $t_i(\hat{\succ}_{-i}, \alpha)$ that delineate available alternatives (from agent i's perspective)
- Challenge is to achieve *feasibility* (all point to same α)

$$\bigcap_i top_i(A_i, \succ_i) \neq \emptyset$$

- For assignment problems, can *transform* (Hashimoto 2013):
  - Allocate preferred choice in set A_i, unless rule is infeasible for $(\succ_i', \hat{\succ}_{-i})$, for some $\succ_i'$.
  - Feasible, and remains IC.
- Ongoing work. Allows *new mechanisms*.

General Framework (Virtual Prices)

- Hypothesis represented by *virtual prices* $t_i(\hat{\succ}_{-i}, \alpha)$ that delineate available alternatives (from agent i's perspective)
- Challenge is to achieve *feasibility* (all point to same α)

$$\bigcap_i top_i(A_i, \succ_i) \neq \emptyset$$

- For assignment problems, can *transform* (Hashimoto 2013):
  - Allocate preferred choice in set A_i, unless rule is infeasible for $(\succ_i', \hat{\succ}_{-i})$, for some $\succ_i'$.
  - Feasible, and remains IC.
- Ongoing work. Allows *new mechanisms*.

General Framework (Virtual Prices)

- Hypothesis represented by *virtual prices* $t_i(\hat{\succ}_{-i}, \alpha)$ that delineate available alternatives (from agent i's perspective)
- Challenge is to achieve *feasibility* (all point to same α)

$$\bigcap_i top_i(A_i, \succ_i) \neq \emptyset$$

- For assignment problems, can *transform* (Hashimoto 2013):
  - Allocate preferred choice in set A_i, unless rule is infeasible for $(\succ_i', \hat{\succ}_{-i})$, for some $\succ_i'$.
  - Feasible, and remains IC.
- Ongoing work. Allows *new mechanisms*.

General Framework (Virtual Prices)

- Hypothesis represented by *virtual prices* $t_i(\hat{\succ}_{-i}, a)$ that delineate available alternatives (from agent i's perspective)
- Challenge is to achieve *feasibility* (all point to same a)

$$\bigcap_i top_i(A_i, \succ_i) \neq \emptyset$$

- For assignment problems, can *transform* (Hashimoto 2013):
  - Allocate preferred choice in set A_i, unless rule is infeasible for $(\succ_i', \hat{\succ}_{-i})$, for some $\succ_i'$.
  - Feasible, and remains IC.
- Ongoing work. Allows *new mechanisms*.

Conclusions (1 of 2)

- *Avoid analytical bottleneck* by using statistical ML to design mechanisms specialized to a particular context.
- *With money*, learn a *payment rule to minimize expected regret*, coupled with outcome rule f.
  - Discriminant function provides the price rule, and the risk-optimal rule is maximally-IC
  - Applications to multi-minded CAs with greedy assignment, and to egalitarian assignment

Conclusions (1 of 2)

- *Avoid analytical bottleneck* by using statistical ML to design mechanisms specialized to a particular context.
- *With money*, learn a *payment rule to minimize expected regret*, coupled with outcome rule f.
  - Discriminant function provides the price rule, and the risk-optimal rule is maximally-IC
  - Applications to multi-minded CAs with greedy assignment, and to egalitarian assignment

Conclusions (2 of 2)

- *Without money*, learn an *IC (stable) outcome rule* to minimize expected loss relative to target outcome rule f.
  - Adopt parametric forms for single-peaked social choice, one-sided matching, and two-sided matching.
  - Learn via structural SVMs, greedy 'tree-splitting' algorithm.

- *Next steps*:
  - Build out the *general approach* for assignment problems, both with and without money (learn new, IC mechanisms)
  - Can a general approach be developed for *non-assignment* problems?
  - Understand the *sample complexity* of these problems

Conclusions (2 of 2)

- *Without money*, learn an *IC (stable) outcome rule* to minimize expected loss relative to target outcome rule f.
  - Adopt parametric forms for single-peaked social choice, one-sided matching, and two-sided matching.
  - Learn via structural SVMs, greedy 'tree-splitting' algorithm.
- *Next steps*:
  - Build out the *general approach* for assignment problems, both with and without money (learn new, IC mechanisms)
  - Can a general approach be developed for *non-assignment* problems?
  - Understand the *sample complexity* of these problems

Thank you

Learning Strategy-proof Mechanisms for Social Choice and Matching Problems, H. Narasimhan, S. Agarwal and D. C. Parkes, Working paper 2016.

Learning Strategy-proof Assignment Mechanisms without Money, H. Narasimhan and D. C. Parkes, Working paper 2016.