Payment rules through Discriminant-Based Classifiers

David C. Parkes
Harvard University

Indo-US Lectures Week in Machine Learning, Game Theory and Optimization

Classical Mechanism Design

- $X \equiv$ valuations; $Y \equiv$ alternatives
- Find $f\colon X^n \mapsto Y$, $t\colon X^n \mapsto R^n$ to maximize objective s.t. IC constraints

Classical Mechanism Design

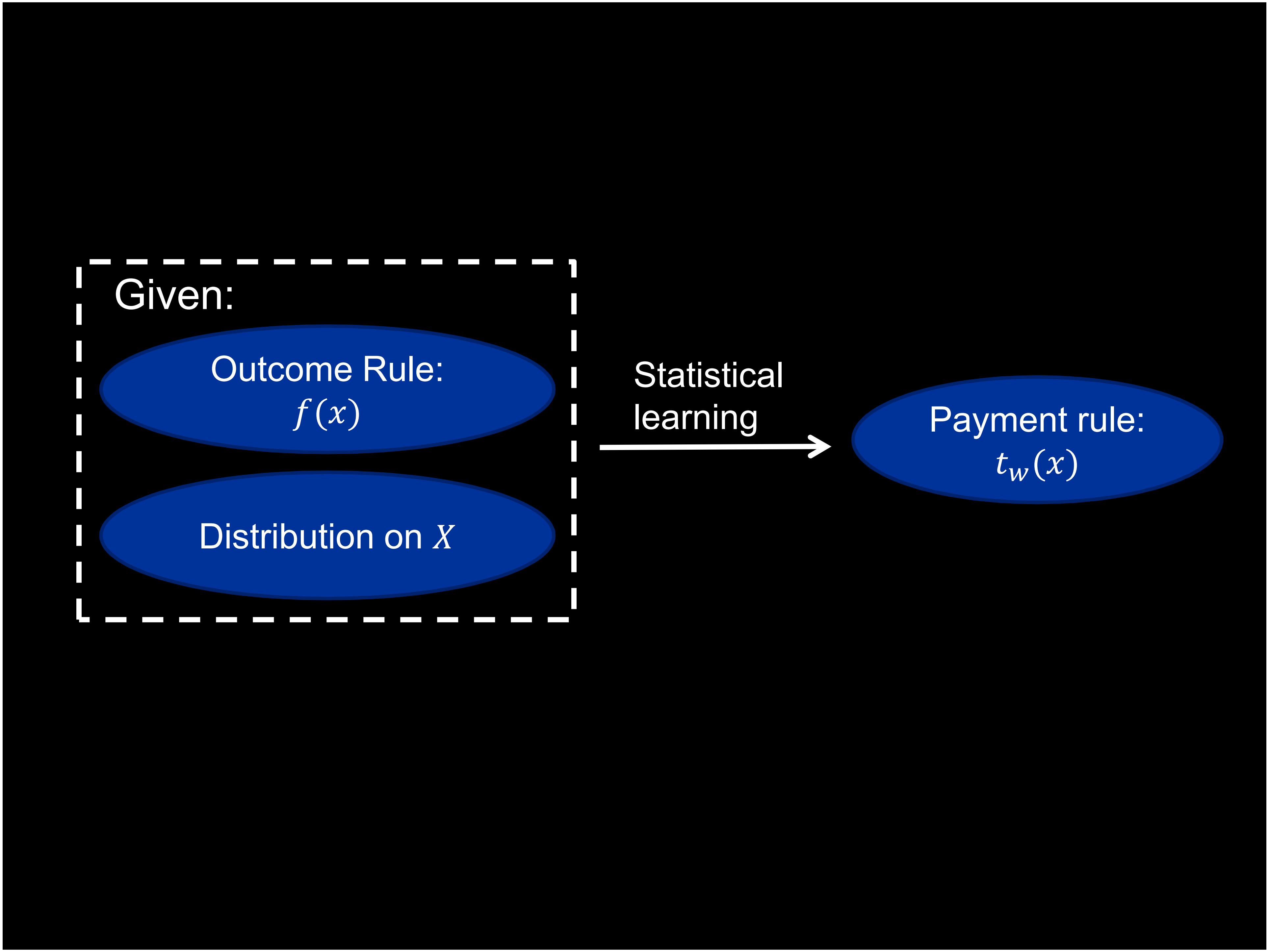
- $X \equiv$ valuations; $Y \equiv$ alternatives
- Find $f: X^n \mapsto Y$, $t: X^n \mapsto R^n$ to maximize objective s.t. IC constraints
- Difficulties:
  - Severe analytical difficulties; e.g., revenue-optimal auction for 2 items, 1 buyer is not known!
  - IC constraints can be too strong (e.g., problems with VCG in CAs)
  - No consideration to computational complexity

Our approach

- $X \equiv$ valuations; $Y \equiv$ alternatives
- Given $f: X \mapsto Y$; $X \sim D$
- $Learn\ t_w: X \mapsto R^n$ that is maximally IC
- Use mechanism (f, t_w)

Our approach

- $X \equiv$ valuations; $Y \equiv$ alternatives
- Given $f\colon X \mapsto Y$; $X \sim D$
- $Learn\ t_w\colon X \mapsto R^n$ that is maximally IC
- Use mechanism (f, t_w)
- Benefits:
  - From analytical bottleneck to Statistical learning
  - Relax IC constraints (good news/ bad news)
  - Learned mechanism tractable, uses rule f



Given:
Outcome Rule: $f(x)$
Distribution on X
Statistical learning
Payment rule: $t_w(x)$

What is "Maximally IC"?

- Given $f: X \mapsto Y$; $X \sim D$
- Learn $t_w: X \mapsto R^n$.

- Ex post regret: given reports of others, what is additional utility from best deviation?
- Maximally IC :: Payment rule t_w minimizes expected ex post regret.

(DSIC if zero)

(Low interim regret w.h.p.)

Useful characterization from MD

- Thm. Mechanism (f, t) is DSIC if (and only if)
  - Agent-independence: Conditioned on $f(x) = y$, then $t_i(x) = p_i(y, x_{-i})$
  - Maximizing: $f(x) \in \arg\max(x_i(y) - p(y, x_{-i}))$

Useful characterization from MD

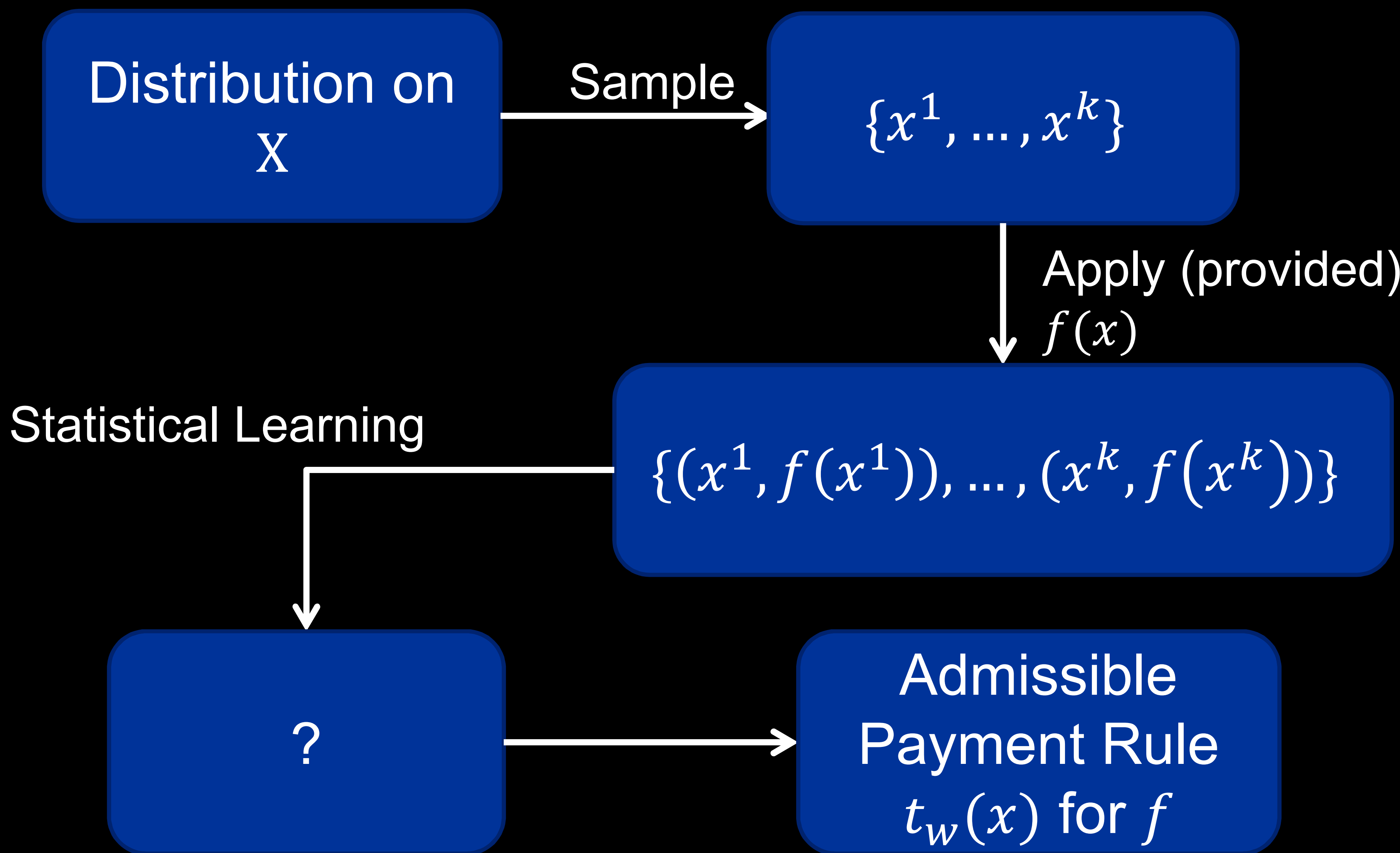
- Thm. Mechanism (f, t) is DSIC if (and only if)
  - Agent-independence: Conditioned on $f(x) = y$, then $t_i(x) = p_i(y, x_{-i})$
  - Maximizing: $f(x) \in \arg\max_y (x_i(y) - p(y, x_{-i}))$
- Example
  - Second price auction \$10, \$6, \$4.
  - $p_{1(win,x_{-1})} = 6$; $p_{2(win,x_{-2})} = 10, p_3(win, x_{-3}) = 10$

Useful characterization from MD

- Thm. Mechanism (f, t) is DSIC if (and only if)
  - Agent-independence: Conditioned on $f(x) = y$, then $t_i(x) = p_i(y, x_{-i})$
  - Maximizing: $f(x) \in \arg\max_y (x_i(y) - p(y, x_{-i}))$
- Example
  - Second price auction \$10, \$6, \$4.
  - $p_{1(win, x_{-1})} = 6$; $p_{2(win, x_{-2})} = 10, p_3(win, x_{-3}) = 10$
- We'll require that t_w is agent-independent.

If also maximizing, then DSIC.

High level approach



Desirable properties

1. DSIC when f is implementable
2. Minimize expected regret
3. Agent-independent prices
4. Scaleable training

Helpful property: Symmetry

- Assume $f(x)$ is symmetric, distribution on X is symmetric
- Wlog: payment rule $t_{w,i}(x_i, y)$ same all i
- Focus on agent 1, and allocation problems

Helpful property: Symmetry

- Assume $f(x)$ is symmetric, distribution on X is symmetric
- Wlog: payment rule $t_{w,i}(x_i, y)$ same all i
- Focus on agent 1, and allocation problems

- Going forward:
- $y = f(x)$: allocation to agent 1
- $t_w(x_{-1}, y)$: payment to agent 1, for outcome y

Basic idea

- Learn discriminant $F_w(x, y) = < w, \psi(x, y) >$

  feature vector

- Predict $h_w = y \in \arg\max_{y \in Y} F_w(x, y)$

Basic idea

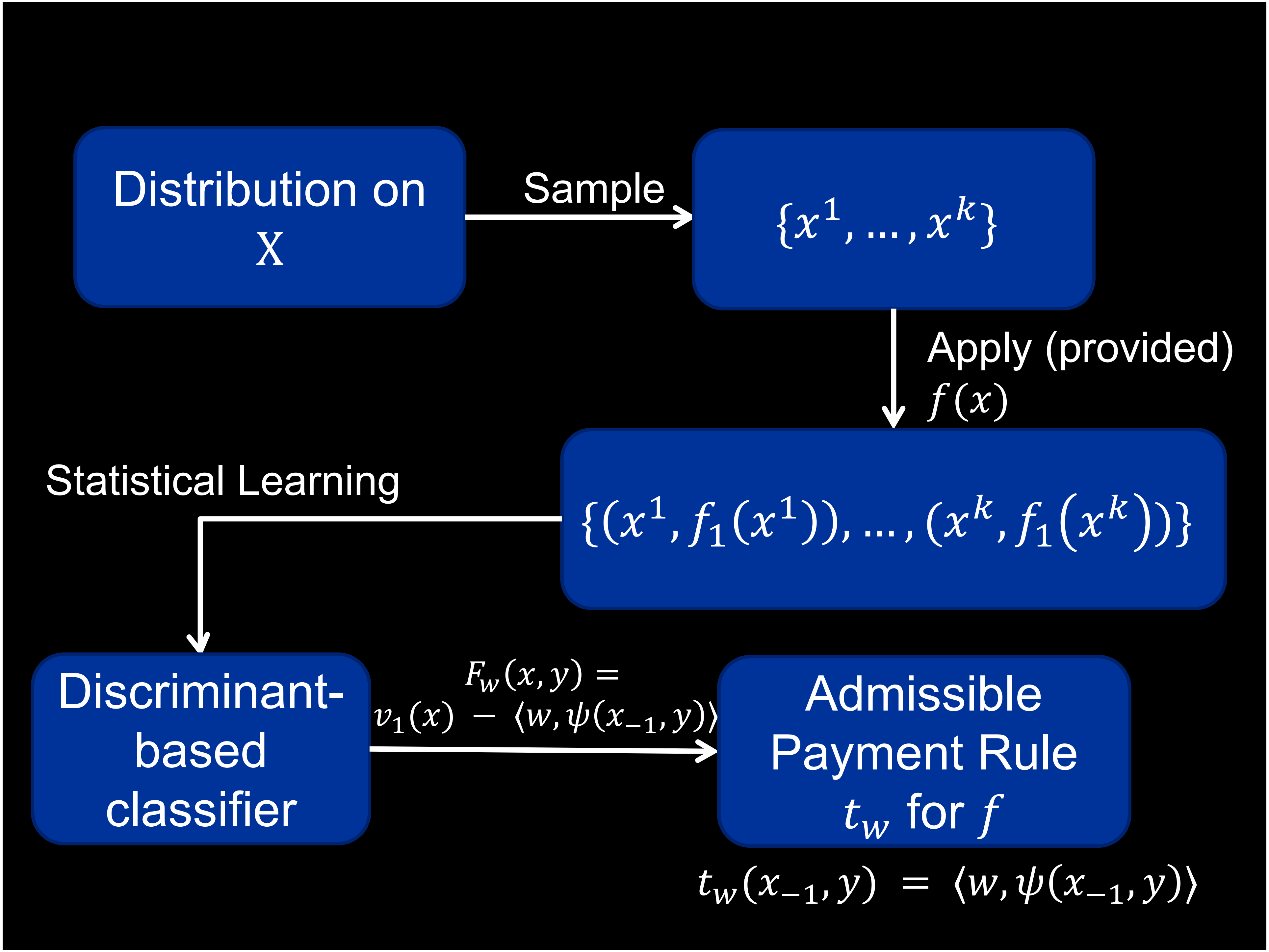
- Learn discriminant $F_w(x,y) \; = < w, \psi(x,y) >$

feature vector

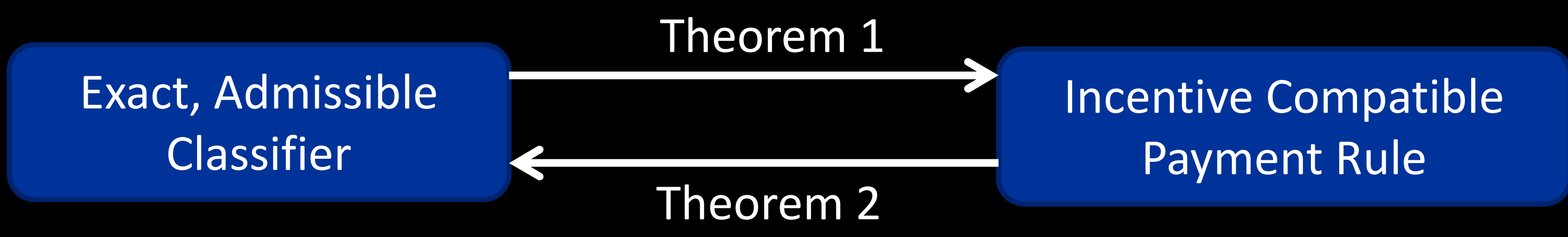
- Predict $h_w = y \in \arg\max_{y \in Y} F_w(x,y)$

				discr $F_w(x,y)$ $y=0$	$y=1$	$h_w(x)$
10	8	4	1	0 - 0	10 - 8	1
11	8	4	1	0 - 0	11 - 8	1
10	12	4	0	0 - 0	10 -12	0

Admissible: $F_w(x,y) \; = \; x_1(y) \; - \; \langle w, \psi(x_{-1}, y) \rangle$


Distribution on X
Sample
$\{x^1, \dots, x^k\}$
Apply (provided) $f(x)$
$\{(x^1, f_1(x^1)), \dots, (x^k, f_1(x^k))\}$
Statistical Learning
Discriminant-based classifier
$F_w(x, y) = v_1(x) - \langle w, \psi(x_{-1}, y)\rangle$
Admissible Payment Rule t_w for f
$t_w(x_{-1}, y) = \langle w, \psi(x_{-1}, y)\rangle$

Theoretical results



- Approximate incentive-compatibility
  - *regret*: Amount an agent can gain by misreporting

Exact classifier $\Rightarrow$ (f, t_w) is DSIC

- Proof. Consider agent 1, fix $x_{-1} = (x_2, \ldots, x_n)$.
- Let $y = f(x\ )$. Consider $x_1' \neq x_1$ for which $y' = f(x_1', x_{-1}) \neq y$

Exact classifier $\Rightarrow$ (f, t_w) is DSIC

- Proof. Consider agent 1, fix $x_{-1} = (x_2, \ldots, x_n)$.
- Let $y = f(x)$. Consider $x_1' \neq x_1$ for which $y' = f(x_1', x_{-1}) \neq y$
- By accuracy, $F_w(x, y) \geq F_w(x, y')$. Implies:

$$x_1(y) - t_w(x_{-1}, y) \geq x_1(y') - t_w(x_{-1}, y').$$

Minimize loss ⇒ min regret

$$R_D = \int_x \mathrm{L}\big(x, f(x), h_w(x)\big) g_D(x) \geq 0$$

= exp regret?

Minimize loss ⇒ min regret

$R_D = \int_x \mathrm{L}\big(x, f(x), h_w(x)\big) g_D(x) \geq 0$ = exp regret?

- Recall $F_w(x, y) = x_1(y) - t_w(x_{-1}, y)$
  - utility to agent 1 in (f, t_w) when outcome is y!

Minimize loss ⇒ min regret

$$R_D = \int_x \mathrm{L}\big(x, f(x), h_w(x)\big) g_D(x) \geq 0$$ = exp regret?

- Recall $F_w(x, y) = x_1(y) - t_w(x_{-1}, y)$
  - utility to agent 1 in (f, t_w) when outcome is y!
- Define $L(x, f(x), y') = F_w(x, y') - F_w(x, f(x))$
  - Non negative, zero when $y' = f(x)$.
  - Depends on x

Minimize loss ⇒ min regret

$$R_D = \int_x \mathrm{L}\big(x, f(x), h_w(x)\big) g_D(x) \geq 0$$ = exp regret?

- Recall $F_w(x, y) = x_1(y) - t_w(x_{-1}, y)$
  - utility to agent 1 in (f, t_w) when outcome is y!
- Define $L(x, f(x), y') = F_w(x, y') - F_w(x, f(x))$
  - Non negative, zero when $y' = f(x)$.
  - Depends on x
- By definition, $F_{w(x,y')} = \max_y [x_1(y) - t_w(x_{-1}, y)]$
- So, $L(.)$ is “max utility” – “actual utility”, regret.

Minimize loss ⇒ min regret

$R_D = \int_x \mathrm{L}\big(x, f(x), h_w(x)\big) g_D(x) \geq 0$ = exp regret

- $L(x, f(x), y') = F_w(x, y') - F_w(x, f(x))$
- Implication:

Minimize loss ⇒ min regret

$R_D = \int_x \mathrm{L}\big(x, f(x), h_w(x)\big) g\_D(x) \geq 0$ = exp regret

- $L(x, f(x), y') = F_w(x, y') - F_w(x, f(x))$
- Implication:
  - If $h_w(x)$ accurate @ x then $M = (f, t_w)$ no regret for agent 1 @ x

Minimize loss ⇒ min regret

$R_D = \int_x \mathrm{L}\big(x, f(x), h_w(x)\big) g\_D(x) \geq 0$ = exp regret

- $L(x, f(x), y') = F_w(x, y') - F_w(x, f(x))$
- Implication:
  - If $h_w(x)$ accurate @ x then $M = (f, t_w)$ no regret for agent 1 @ x
  - If $h_w(x)$ not accurate @ x, regret depends on amount by which disc $\max_y F_w(x, y) > F_w(x, f(x))$

Experimental results

- SVM^{struct} (Joachims et al.2009)
- Multi-minded combinatorial auctions
  - Generator: vary degree of complementarity ζ
  - Outcome rule: greedy algorithm to maximize welfare
- Egalitarian assignment
  - Generator: Item values are U[0,1]
  - Outcome rule: maximize egalitarian welfare
- Use RBF kernel
- Benchmark: VCG-based rules

Benchmark: VCG-based rules

- Given outcome rule $f(x)$, measure externality imposed by agent 1

$f(x)$:	{ 9AM }	{ 11AM }	{1PM}
Value:	\$5M	\$4M	\$3M

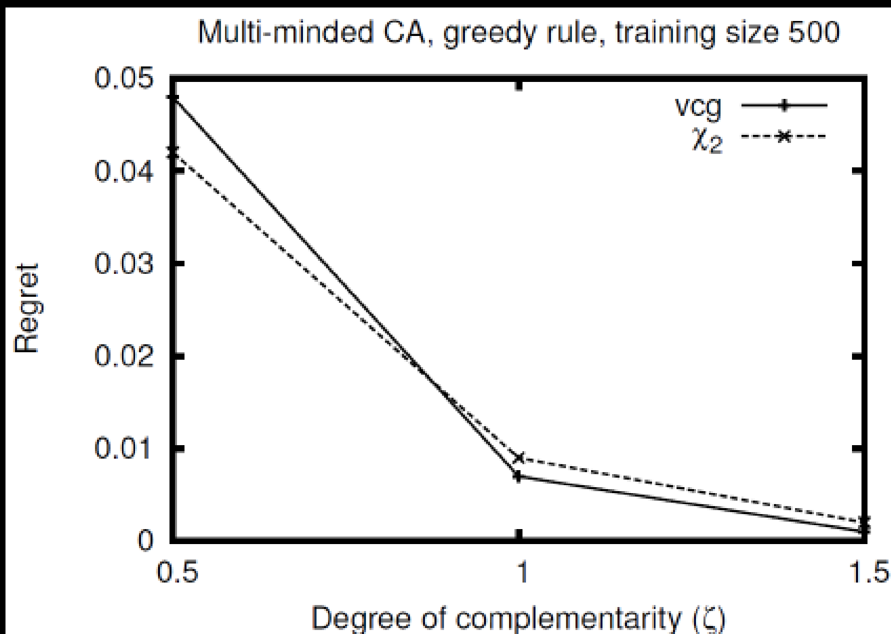
$f(x_{-1})$:	{ 11AM }	{ 9AM, 1PM }
Value:	\$4M	\$7M

Payment: \$11M – 7M = \$4M

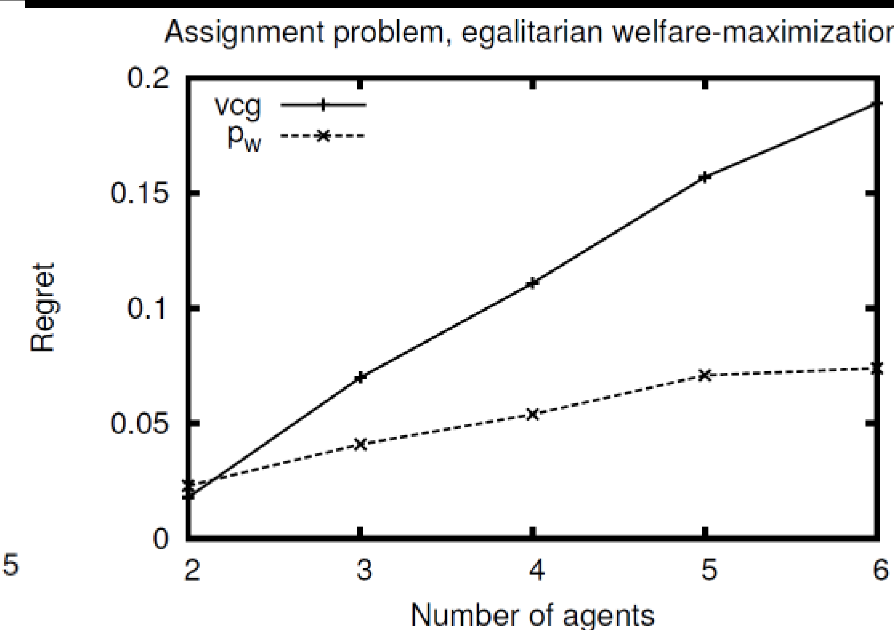
- Incentive compatible if $f(x)$ exactly maximizes welfare
- Not incentive compatible in either of our settings

Results (I)

Multi-minded CA



Egalitarian assignment



- Regret is normalized (1 = max value for a package)

CA training is bottleneck; limited to small instances, small amounts of training data

Training bottleneck

- Solve convex program of following form:

$$\min_{w,\xi \geq 0} \frac{1}{2} \|w\|^2 + C \sum_k \xi_k$$

s.t.

$$F_w\left(x^k, f\left(x^k\right)\right) \geq F_w\left(x^k, y\right) - \xi_k, \forall k,$$

Many constraints

$\forall y$

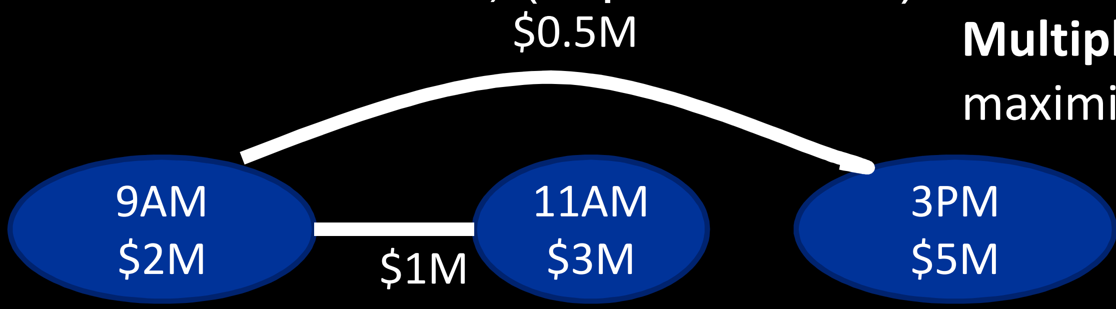
- For CA's, label space is exponential in # items
- Need efficient separation oracle (Joachims et al):

$$\max_y F_w(x, y) = \max_y x_1(y) - \langle w, \psi(x_{-1}, y) \rangle$$

Or: directly formulate separation as a compact LP, roll into primal convex program (Taskar et al. '04)

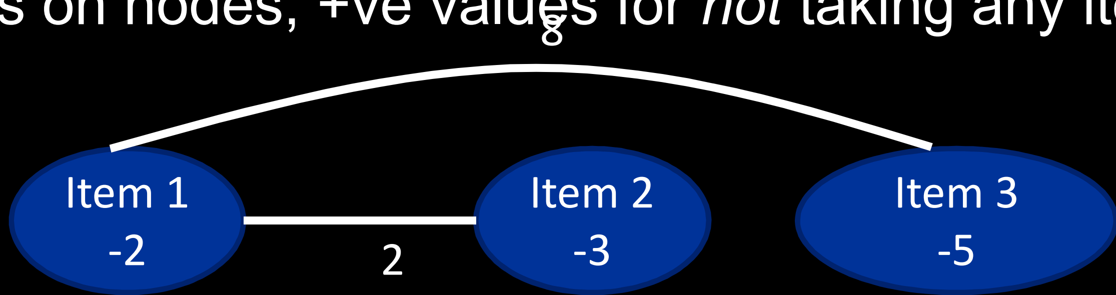
A tractable case

- (positive) k-wise dependent valuations (Conitzer and Sandholm, 2005; Abraham et al., 2012) (supermodular)



Multiple agent welfare-maximization still NP-hard

- Tractable for single-agent semi-positive 2-wise: allow –ve values on nodes, +ve values for *not* taking any item of a pair.



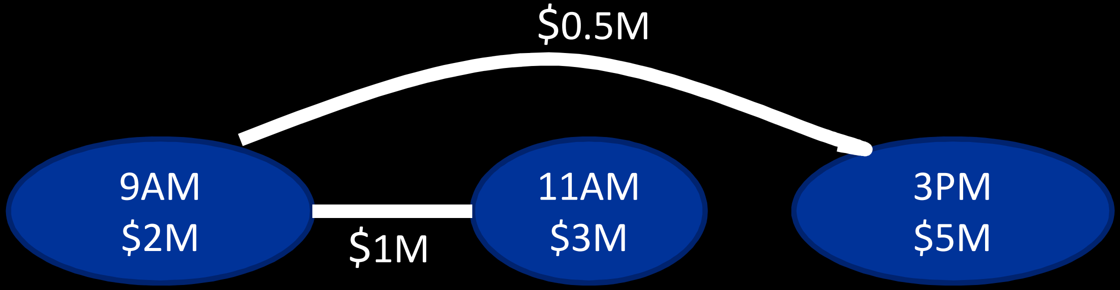
{1}: -2
{2}: -3
{1,2}:-3
{2,3}:-8
{1,3}:1
{1,2,3}:0

- By connection with MRFs, the separation problem can be solved via compact LP (Taskar et al., 2004)

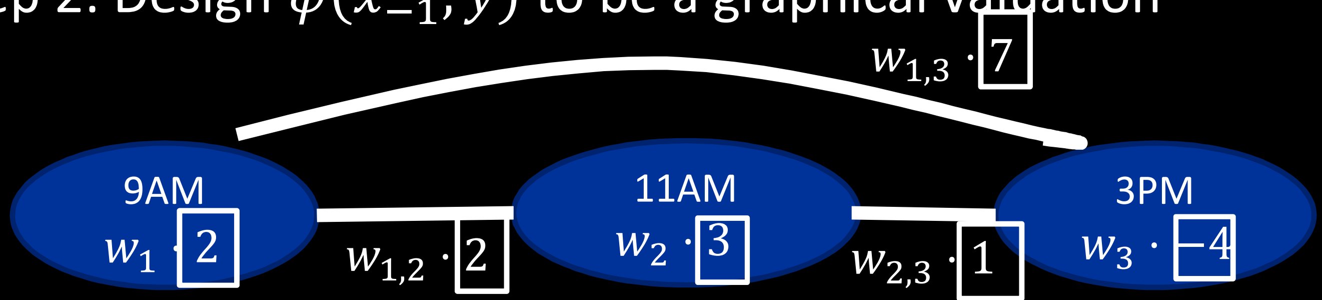
Applying this result

$$\max_y x_1(y) - \langle w, \psi(x_{-1}, y)\rangle$$

- Make this additive over nodes and edges, semi-positive
- Step 1: Assume x_1 is a positive k-wise dependent valuation



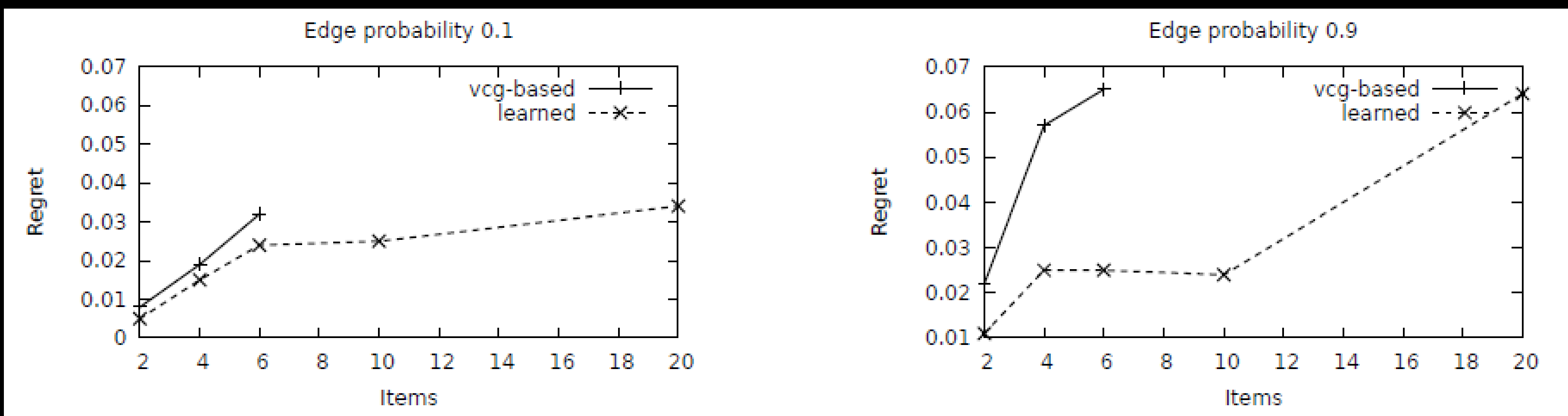
- Step 2: Design $\psi(x_{-1}, y)$ to be a graphical valuation



- Training is now tractable even though WDP is not

Results (II): k-wise dependent valuations

- Outcome rule: greedy welfare-maximization
- Edge prob. controls density of 2-valuation graph
- Regret is normalized (1 = expected value for bundle consisting of all items)



*vcg-based not available for {10, 20}, intractable to compute regret (our number is upper-bound there)

Conclusions

- Use statistical learning to find a payment rule that works well with a given allocation rule
- Connection with discriminant-based classifiers, and structural prediction
- Use of kernels allow for non-linear pricing rules (c.f., Lahaie 10)
- Future work: learn strategyproof mechanisms, in domains without money.
  - Challenge: the discriminant needs to support a feasible allocation.

Reference

- Payment Rules through Discriminant-Based Classifiers, Paul Duetting, Felix Fischer, Pichayut Jirapinyo, John K. Lai, Benjamin Lubin, and David C. Parkes, *ACM Transactions on Economics and Computation*, 2014 (forthcoming)