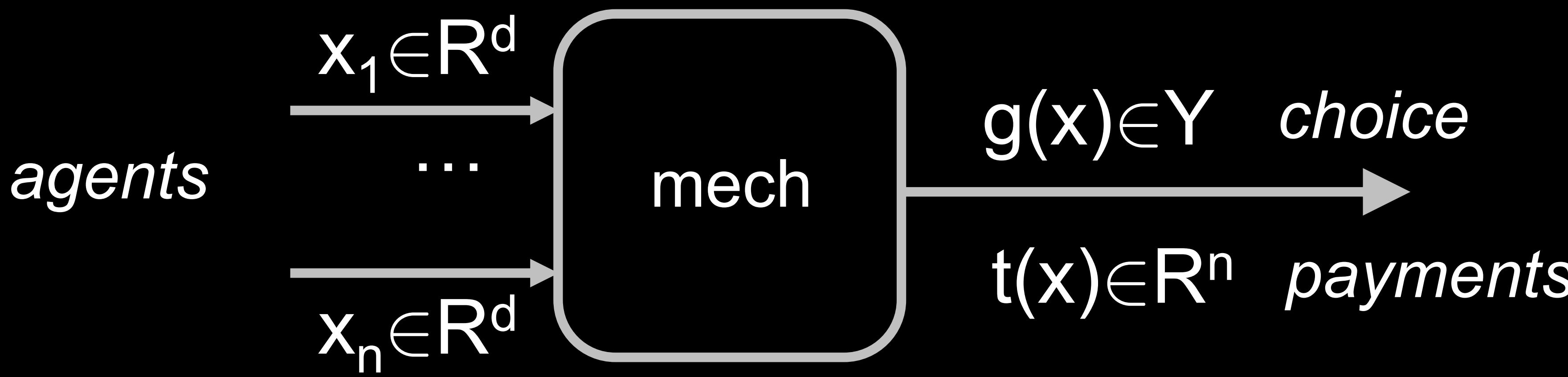
The Interplay of Machine Learning and Mechanism Design

David C. Parkes
Harvard University

learning a decision function given a distribution on inputs

make a decision given selfish inputs

A mechanism



value $v_i(y,x_i) = x_i(y) \in R$

$$x_i(g(x_i x_{-i})) - t_i(x_i,x_{-i}) \geq x_i(g(x_i',x_{-i})) - t_i(x_i',x_{-i})$$

$$\forall i, \forall x_i, \forall x_{-i}, \forall x_i'$$

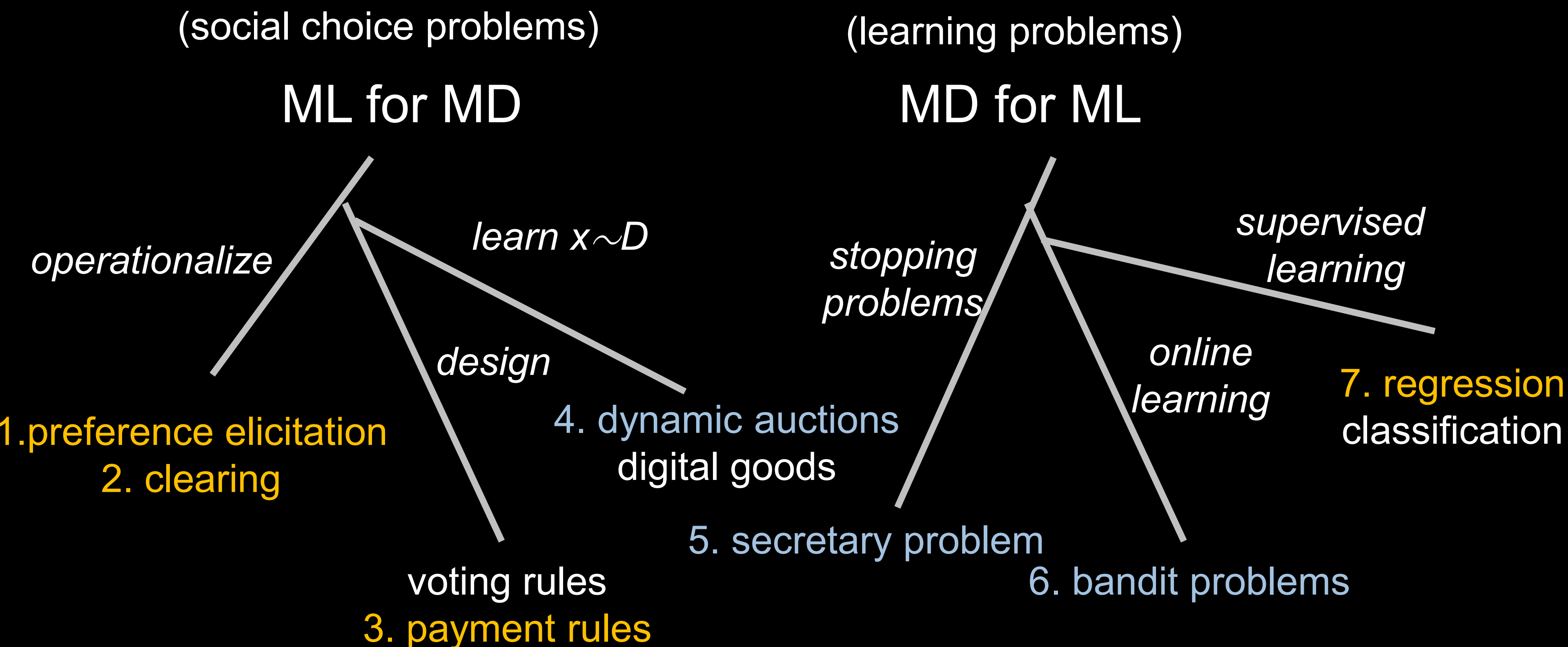
Example: Combinatorial Auction

- G goods ($|G|=m$), N agents
- Valuations x_i: $\{0,1\}^m \rightarrow R$
- Allocation $g(x) = (y_1, y_2, \ldots, y_n)$
  - feasible $y_i \subseteq G$

Example: Combinatorial Auction

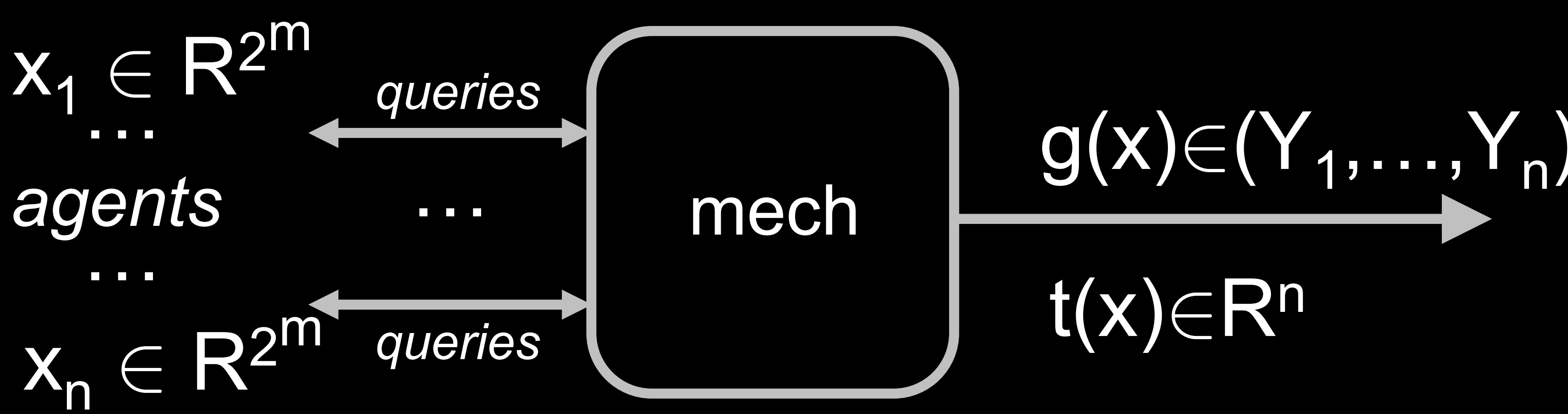
- G goods ($|G|=m$), N agents
- Valuations x_i: $\{0,1\}^m \rightarrow R$
- Allocation $g(x) = (y_1, y_2, \ldots, y_n)$
  - feasible $y_i \subseteq G$

- E.g., single-minded CA
- $G=\{A,B\}$
- $x_1=(10,0,10)$, $x_2=(0,0,19)$, $x_3=(0,8,8)$
- $g(x) = (\emptyset, AB, \emptyset)$, $t(x)=(0,18,0)$



(social choice problems)
ML for MD
operationalize
learn x∼D
design
1.preference elicitation
2. clearing
4. dynamic auctions
digital goods
voting rules
3. payment rules
(learning problems)
MD for ML
stopping problems
supervised learning
online learning
7. regression
classification
5. secretary problem
6. bandit problems

1. Preference Elicitation



Representation Languages

Representation Languages

- Atomic: { (A, 10), (B, 12), (BC, 20)}
  - set of (bundle, value) pairs
- L_{XOR}: $x_i(AB) = 12$; $x_i(ABC) = 20$
- L_{OR}: $x_i(AB) = 22$; $x_i(ABC) = 30$

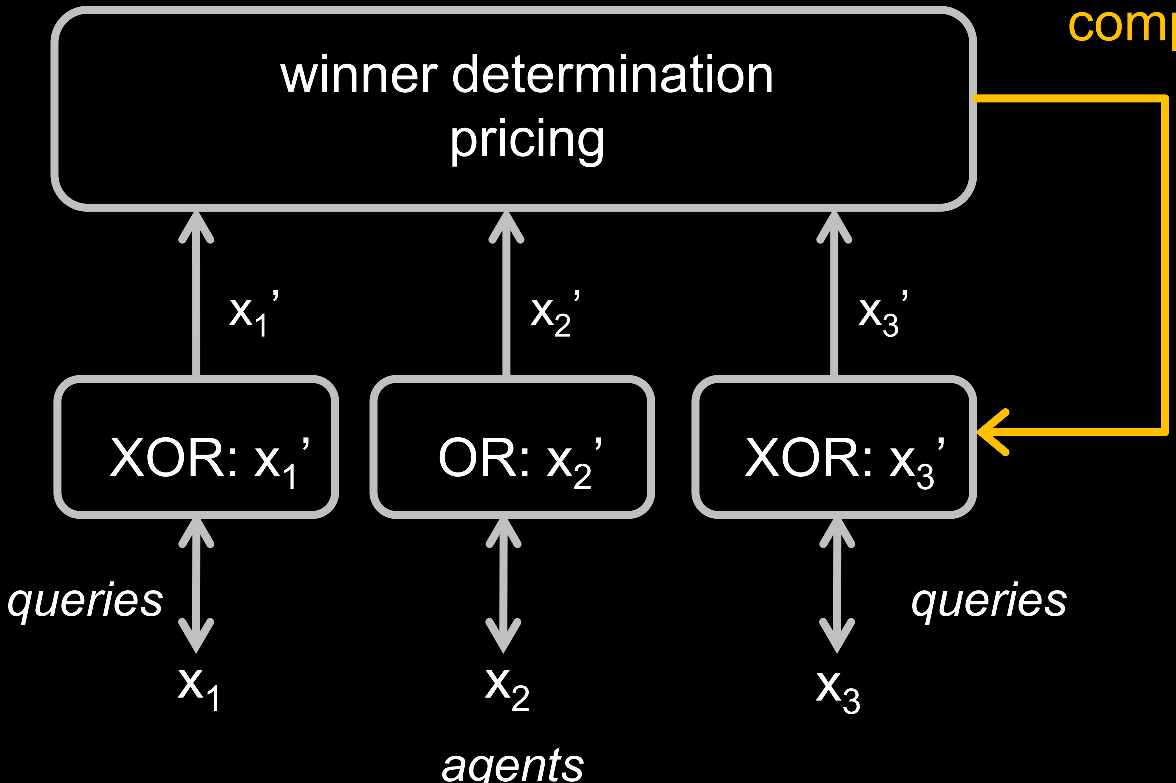
  … other languages
- *size*$_L(x_i)$: minimal |B| to represent x_i in L

Goal: Exact query learning with *value* and (linear) *demand* queries
$D_i(p) = \arg\max_s x_i(s) - \sum_j p_j s_j$
#queries poly in size, m and n

Efficient elicitation with *value* and *demand* queries
Goal: find g(x)
#queries poly in size, m, and n

Elicitation by Learning

(Lahaie & P. EC'04)



competitive equilibrium
(y,p) given x'

membership:
"is f(x) = y"?
Yes, or "no, y' is."
Value: "what is f(x)"?

Demand:
"does s maximize
utility given p"?
Yes, or "no, s' does."
Value: "what is $v_i(s)$"?

- Thm. Polynomial-query *elicitation* with value and demand queries when hypothesis class can be *polynomial-query exactly* learned with membership and equivalence queries.

Elicitation: Modularity

- Modular framework; e.g.,
  - polynomial $\equiv$ general valuations
  - monotone DNF $\equiv$ XOR
  - L_k (including OR)
- Can tune for particular setting

Extension: Incentives

(Constantin, Lahaie, P. AAAI'05)

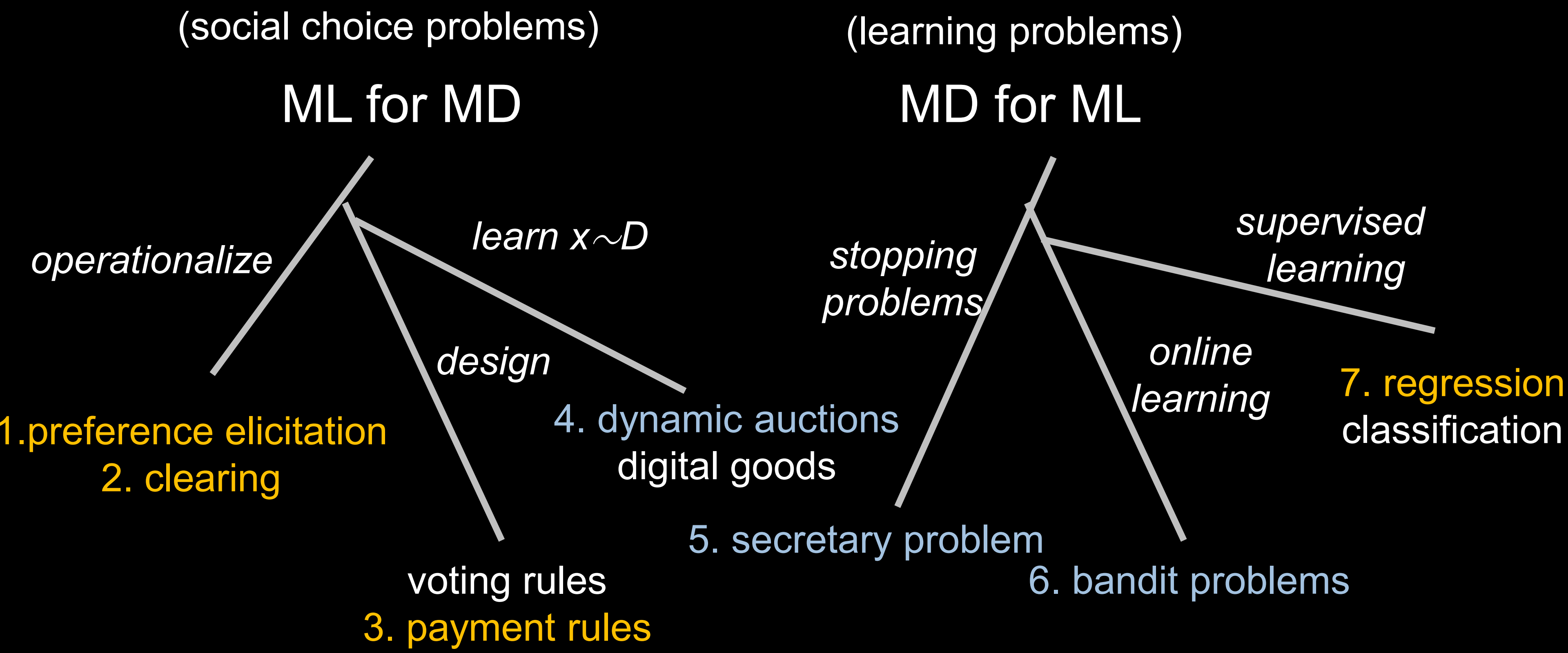
- Adopt *Universal* CE prices
  - CE (y,p): agents happy, seller happy
  - UCE (y,p): CE for {N} and $\{N_{-1}, \ldots, N_{-n}\}$
- Thm. Communication protocol for UCE $\Leftrightarrow$ Communication protocol for VCG
- Idea: simulate until get to UCE

Extension: Incentives

(Constantin, Lahaie, P. AAAI'05)

- Adopt *Universal* CE prices
  - CE (y,p): agents happy, seller happy
  - UCE (y,p): CE for {N} and $\{N_{-1}, \ldots, N_{-n}\}$
- Thm. Communication protocol for UCE ⇔ Communication protocol for VCG
- Idea: simulate until get to UCE
- Example: (10,0,10), (0,0,19), (0,8,8)

not CE	CE not UCE	UCE
(20,20)	(10.5,8.5)	(10,8)
(5,5)		


(social choice problems)
ML for MD
operationalize
learn x∼D
design
1.preference elicitation
2. clearing
4. dynamic auctions
digital goods
voting rules
3. payment rules
(learning problems)
MD for ML
stopping problems
supervised learning
online learning
7. regression
classification
5. secretary problem
6. bandit problems

2. Kernel Methods for Clearing

(Lahaie'09, Lahaie'10)

- Standard: $(x_1,\ldots,x_n) \rightarrow$ Solve n+1 problems $\rightarrow$ compute VCG
- (i) Miserable for large n; (ii) Maybe don't need exact feasibility

2. Kernel Methods for Clearing

(Lahaie'09, Lahaie'10)

- Standard: $(x_1,\ldots,x_n) \rightarrow$ Solve n+1 problems $\rightarrow$ compute VCG
- (i) Miserable for large n; (ii) Maybe don't need exact feasibility

- A kernel approach:
  - Use kernels to represent prices in high dimensional space, get new flexibility
  - Compute allocation and payments in one step
  - Connection between stability, UCE and VCG

Set-up: Kernels for Clearing

- Single minded bidders
- View non-linear prices as linear prices in high dimensional space;
- $S=\{0,1\}^m$, $\phi: S \rightarrow R^M$, $p(s) = <w,\phi(s)>$, $w \in R^M$
- Linear kernel $k(s_1,s_2) = <s_1,s_2>$
- Identity kernel $k(s_1,s_2) = 1$, if $s_1 = s_2$
  0, otherwise
  - All subsets, Gaussian,…

Primal formulation

(c.f. SVM dual)

$$\max_{\alpha\geq 0,\bar{\alpha}\geq 0} \quad \sum_{i=1}^{n} \alpha_i v_i(x_i) - \frac{1}{2\lambda}\left\|\sum_{i=1}^{n}\left(\alpha_i - \sum_{I\ni i}\bar{\alpha}_I\right)\phi(x_i)\right\|^2 \tag{1}$$

$$\text{s.t.} \quad \alpha_i \leq 1 \qquad i = 1,\ldots,n$$

$$\sum_{I\in\mathcal{F}} \bar{\alpha}_I \leq 1 \qquad I \in \mathcal{F}$$

λ small

k(.,.) complex

closer to feasible;
more integer solutions

Dual formulation

(c.f. SVM primal)

$$\min_{\pi \geq 0, \bar{\pi} \geq 0, p} \quad \sum_{i=1}^{n} \pi_i + \bar{\pi} + \frac{\lambda}{2} \|p\|^2$$

$$\text{s.t.} \quad \pi_i \geq v_i(x_i) - \langle p, \phi(x_i) \rangle \quad i = 1, \ldots, n$$

$$\bar{\pi} \geq \left\langle p, \sum_{i \in I} \phi(x_i) \right\rangle \quad I \in \mathcal{F}$$

$$p(x) = \frac{1}{\lambda} \sum_{i=1}^{n} \left(\alpha_i - \sum_{I \ni i} \bar{\alpha}_I \right) k(x_i, x)$$

Regularization: stability (Bousquet & Elisseef'02)

Higher λ, closer to UCE prices!

Incentive Analysis

(Lahaie AAAI'10)

Proposition

If p is an optimal dual solution with all buyers present, and p^{-h} is an optimal dual solution with buyer h removed, then

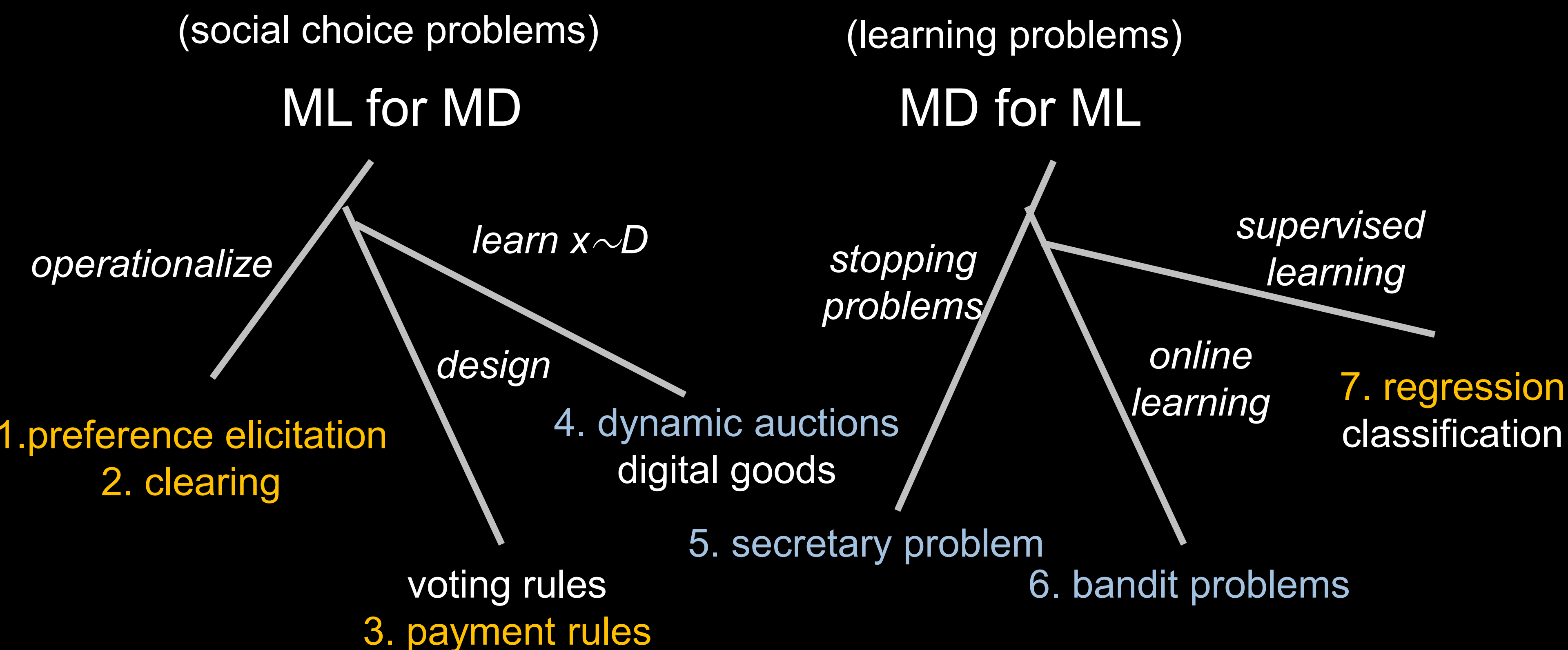
$$\left\| p - p^{-h} \right\| \leq \frac{\kappa}{\lambda}.$$

Theorem

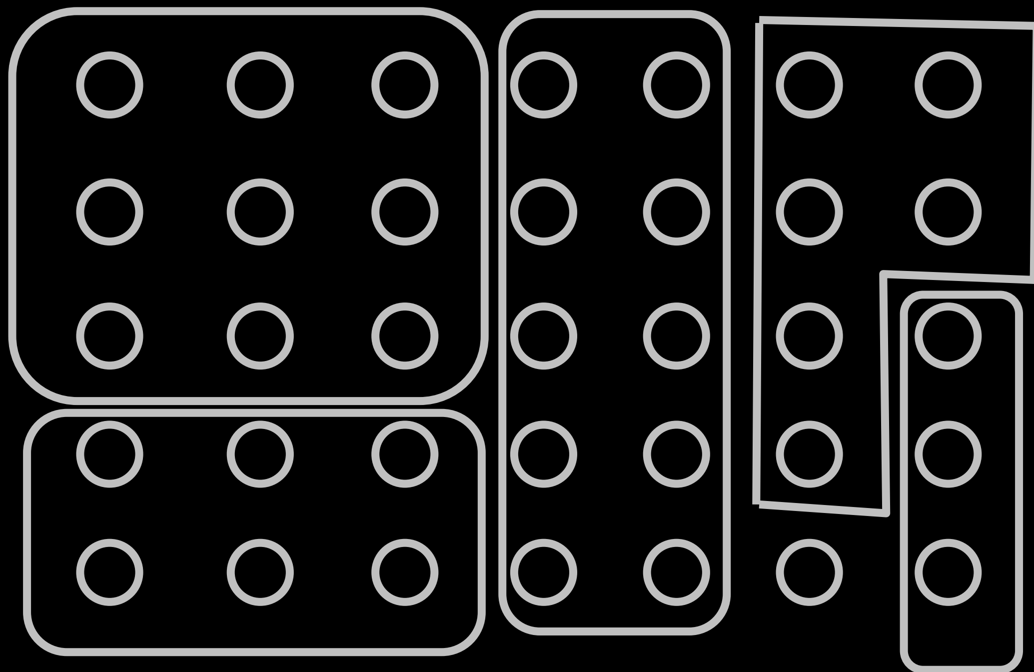
The auction that charges payments r is ϵ-incentive compatible for

$$\epsilon = \frac{2(n-1)\kappa^2}{\lambda}.$$

- k and λ tradeoff: fix k; find smallest λ s.t. feas OK


(social choice problems)
ML for MD
operationalize
learn x∼D
design
1.preference elicitation
2. clearing
4. dynamic auctions
digital goods
voting rules
3. payment rules
(learning problems)
MD for ML
stopping problems
supervised learning
online learning
7. regression
classification
5. secretary problem
6. bandit problems

3. Learning Mechanism Rules



idea: learn a pricing function for each i that "separates" demand sets and is independent of x_i

Classification problem

- Fix allocation rule $g: X \to Y$
- Data: $\{ (x^1, y^1), (x^2, y^2), \ldots\}$ from $(x,y) \sim D$
  - $x \in R^{2^m \times n}$; $y \in \{0,1\}^m$
- Learn: $g'(x): X \to Y$, as $g'(x) = \arg\max_{y'} f(x,y')$
- Derive payment rule $t_1(x)$ from $f(x,y)$

Example: Single-item allocation

- $X = R^n$; $Y = \{\pm 1\}$.
- Inputs: ((10,8,7), 1), ((5,8,7), -1), ((9,2,5), +1)
- Learn $f: R^n \rightarrow R$; $g'(x) = \mathrm{sgn}(f(x))$

Example: Single-item allocation

- $X = R^n$; $Y = \{\pm 1\}$.
- Inputs: ((10,8,7), 1), ((5,8,7), -1), ((9,2,5), +1)
- Learn $f: R^n \to R$; $g'(x) = \mathrm{sgn}(f(x))$
- Require: $f(x) = x_1 + \langle w', \phi(x_{-1}) \rangle = x_1 - p(x_{-1})$
- Exact classifier: $f(x) = x_1 - \max(x_{-1})$

Example: Single-item allocation

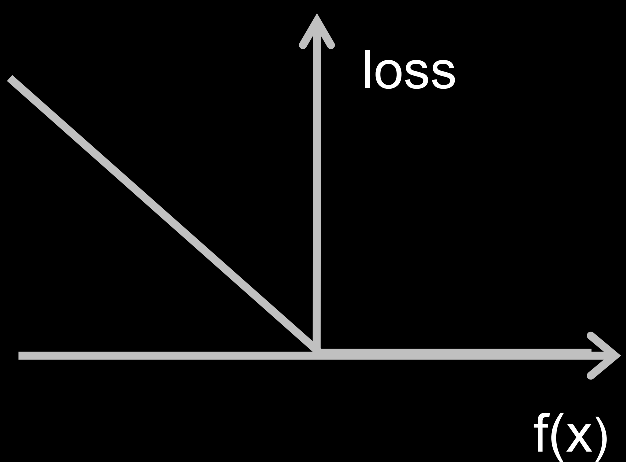
- $X = R^n$; $Y = \{\pm 1\}$.
- Inputs: ((10,8,7), 1), ((5,8,7), -1), ((9,2,5), +1)
- Learn $f: R^n \to R$; $g'(x) = \mathrm{sgn}(f(x))$
- Require: $f(x) = x_1 + <w', \phi(x_{-1})> = x_1 - p(x_{-1})$
- Exact classifier: $f(x) = x_1 - \max(x_{-1})$

- SP: If $x_1 - p(x_{-1}) = f(x) \geq 0$, then $g(x) = 1$

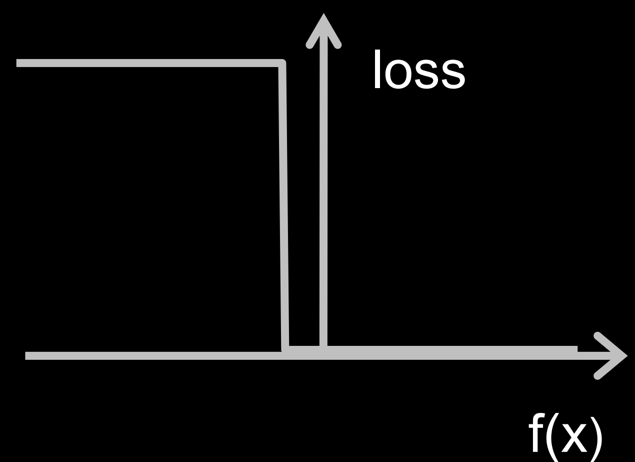
  $x_1 - p(x_{-1}) = f(x) < 0$, then $g(x) = -1$

Loss functions

- Regret = (-y f(x))+
- E.g., y=+1, but $x_1 - p_1(x_{-1}) < 0$



min expected regret on (x,y) $\sim$ D



min (prob regret $\geq \epsilon$) on (x,y) $\sim$ D

General problem

- $x \in R^{2^m \times n}$; $y \in \{0,1\}^m$
- Learn g: $X \rightarrow Y$; $g(x) = \arg\max_y f(x,y)$

General problem

- $x \in R^{2^m \times n}$; $y \in \{0,1\}^m$
- Learn $g: X \to Y$; $g(x) = \arg\max_y f(x,y)$
- Stipulate $f(x,y) = \langle w, \phi(x,y)\rangle =$
  $x_1(y) + \langle w_1', \phi(x_{-1}, y)\rangle$
- $p_1(x,y') = - \langle w', \phi(x_{-1}, y)\rangle$

General problem

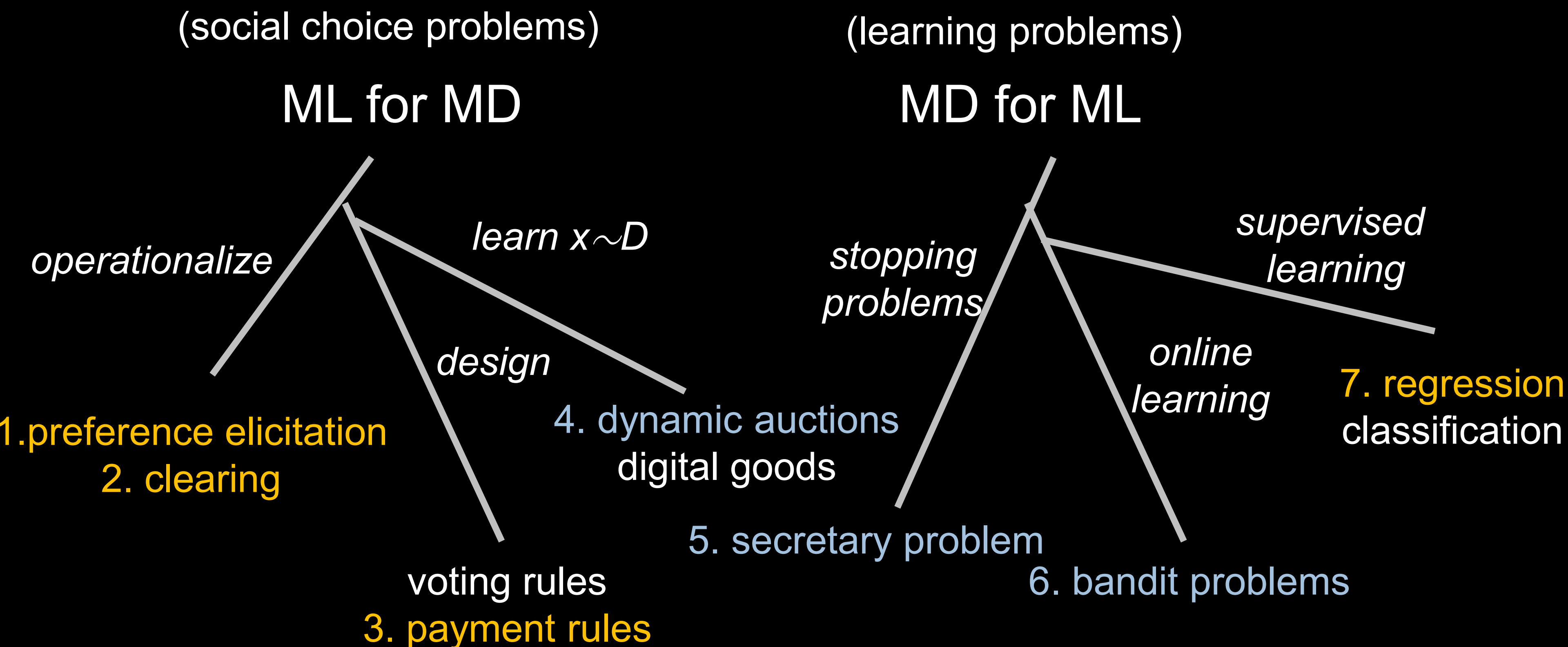
- $x \in R^{2^m \times n}$; $y \in \{0,1\}^m$
- Learn g: $X \rightarrow Y$; $g(x) = \arg\max_y f(x,y)$
- Stipulate $f(x,y) = \langle w, \phi(x,y) \rangle =$

  $x_1(y) + \langle w_1', \phi(x_{-1}, y) \rangle$
- $p_1(x,y') = - \langle w', \phi(x_{-1}, y) \rangle$
- Multiclass SVM with exponential |Y| but only small number of relevant labels for any input

General problem

- $x \in R^{2^m \times n}$; $y \in \{0,1\}^m$
- Learn g: $X \rightarrow Y$; $g(x) = \arg\max_y f(x,y)$
- Stipulate $f(x,y) = \langle w, \phi(x,y)\rangle =$
  $x_1(y) + \langle w_1', \phi(x_{-1}, y)\rangle$
- $p_1(x,y') = - \langle w', \phi(x_{-1}, y)\rangle$
- Multiclass SVM with exponential |Y| but only small number of relevant labels for any input

- **Theorem**. An exact classifier induces a strategyproof payment rule.


(social choice problems)
ML for MD
operationalize
learn x∼D
design
1.preference elicitation
2. clearing
4. dynamic auctions
digital goods
voting rules
3. payment rules
(learning problems)
MD for ML
stopping problems
supervised learning
online learning
7. regression
classification
5. secretary problem
6. bandit problems

4. Dynamic auctions

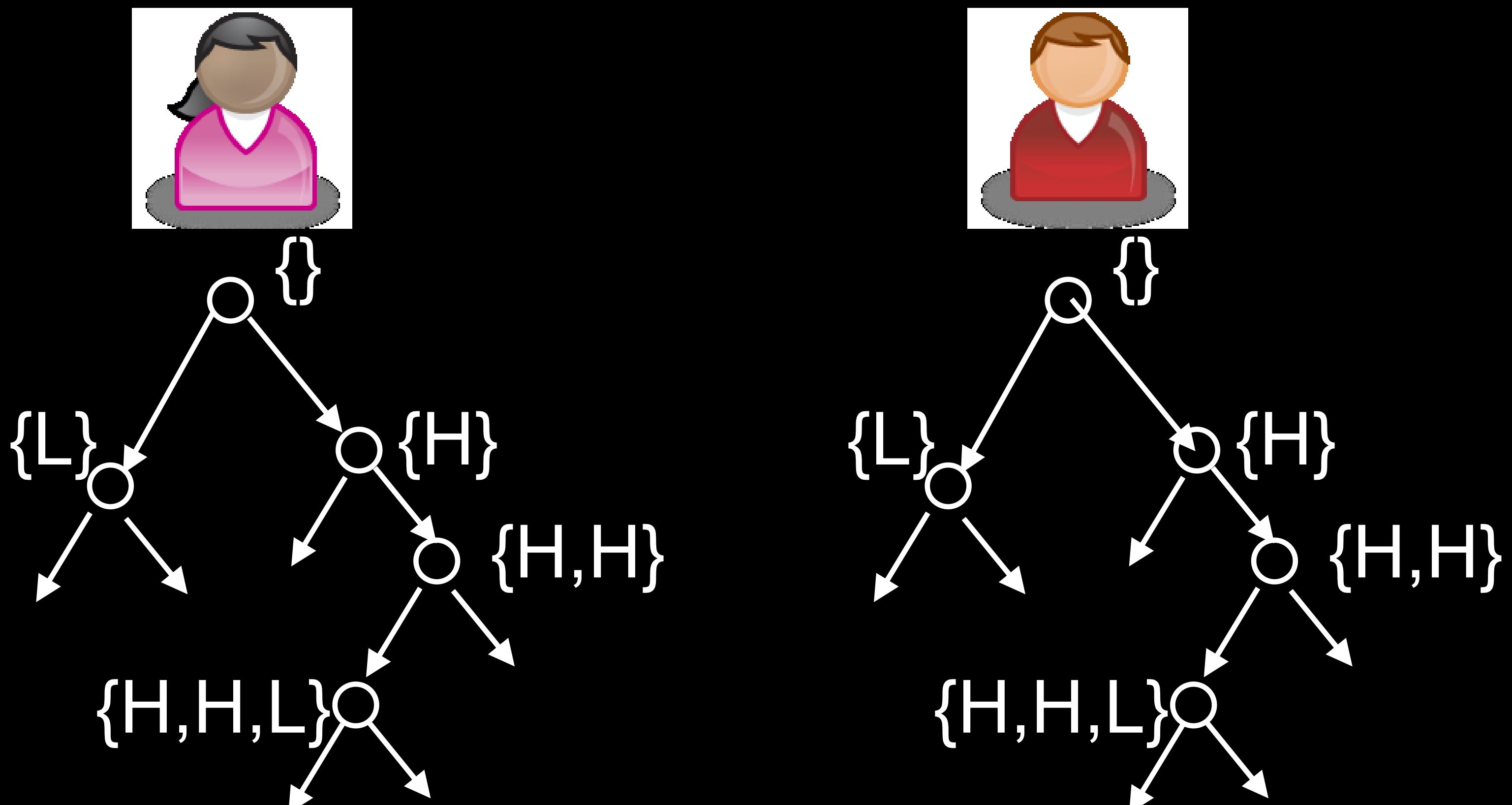
5. Secretary problems

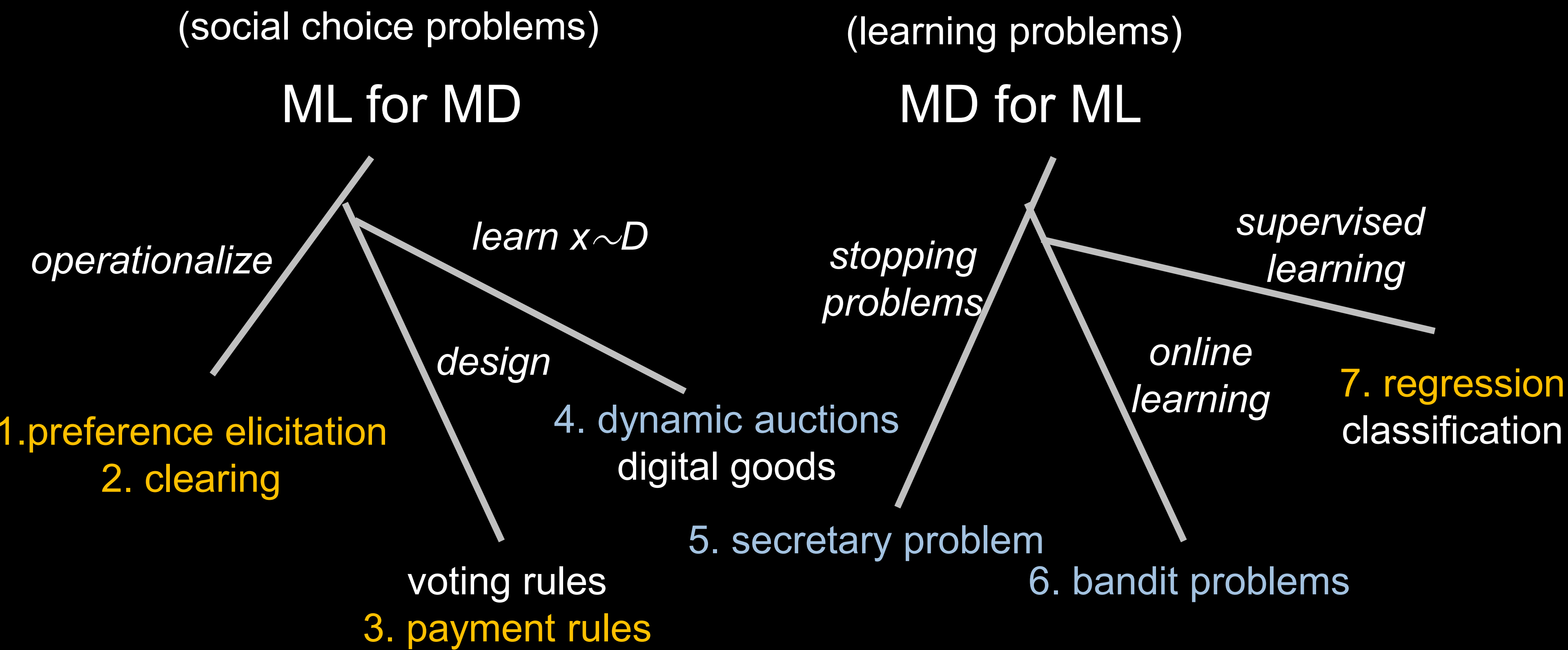
- Bids = secretaries
- Q: how to make the “1/e” online algorithm DSIC despite strategic inputs?
- A: Kleinberg, Mahdian & Parkes EC’03

6. Bandits problems

(Cavallo, Singh & P. UAI'06',
Bergemann & Valimaki'10)

- Bandits: arms = agents
- Q: how to make the agents report true reward and thus next state?




(social choice problems)
ML for MD
operationalize
learn x∼D
design
1.preference elicitation
2. clearing
4. dynamic auctions
digital goods
voting rules
3. payment rules
(learning problems)
MD for ML
stopping problems
supervised learning
online learning
7. regression
classification
5. secretary problem
6. bandit problems

7. IC Supervised Learning

- Learn $f: X \rightarrow R$
- Each agent i: $D_i \sim X$; $g_i: X \rightarrow R$
- $R_i(f) = E_{x \sim D_i} [\, \mathit{loss}(f(x), g_i(x)) \,]$
- Goal: min $R(f) = \sum_i R_i(f)$

- Rational agents, may misreport samples!

Framework

(Dekel, Fischer & Procaccia'08)

- Request m points $S_i = \{ (x_{ij}, y_{ij}) \}_{j=1}^{m}$
- Report $S'_i \neq S_i$
- One idea: select f' to be empirical risk minimizer
- Q: when will this be DSIC?

Warm-up: Special case

- m=1, D_i degenerate. Report y'_i

Warm-up: Special case

- m=1, D_i degenerate. Report y'_i
- Thm. For a linear |y-y'| loss function, convex hypothesis class F, then ERM is DSIC

Warm-up: Special case

- m=1, D_i degenerate. Report y'_i
- Thm. For a linear |y-y'| loss function, convex hypothesis class F, then ERM is DSIC
- E.g., {(1,6), (2,5), (3,1)}. Constant f(x)=c.
- ERM: select median. DSIC!

Warm-up: Special case

- m=1, D_i degenerate. Report y'_i
- Thm. For a linear |y-y'| loss function, convex hypothesis class F, then ERM is DSIC
- E.g., {(1,6), (2,5), (3,1)}. Constant f(x)=c.
- ERM: select median. DSIC!

- Fails for other loss functions.
- E.g., {(1,2), (2,1), (3,0)}. Squared loss $|y-y'|^2$

General Case

- For m>1 points, and D_i distribution, and absolute loss function

General Case

- For m>1 points, and D_i distribution, and absolute loss function
- Example. N={1,2}. Constant f(x)=c
- S_1={ (1,1), (2,1), ~~(3,0)~~ } (3,1)
- S_2 = { (4,0), (5,0), (6,1) }
- f(x)=0; $R_1(f) = 2/3$ ⟶ f(x)=1; $R_1(f)=1/2$

General Case

- For m>1 points, and D_i distribution, and absolute loss function
- Example. N={1,2}. Constant f(x)=c
- S_1={ (1,1), (2,1), ~~(3,0)~~ } (3,1)
- S_2 = { (4,0), (5,0), (6,1) }
- f(x)=0; R_1(f) = 2/3 ⟶ f(x)=1; R_1(f)=1/2

- Solution: *project and fit.*

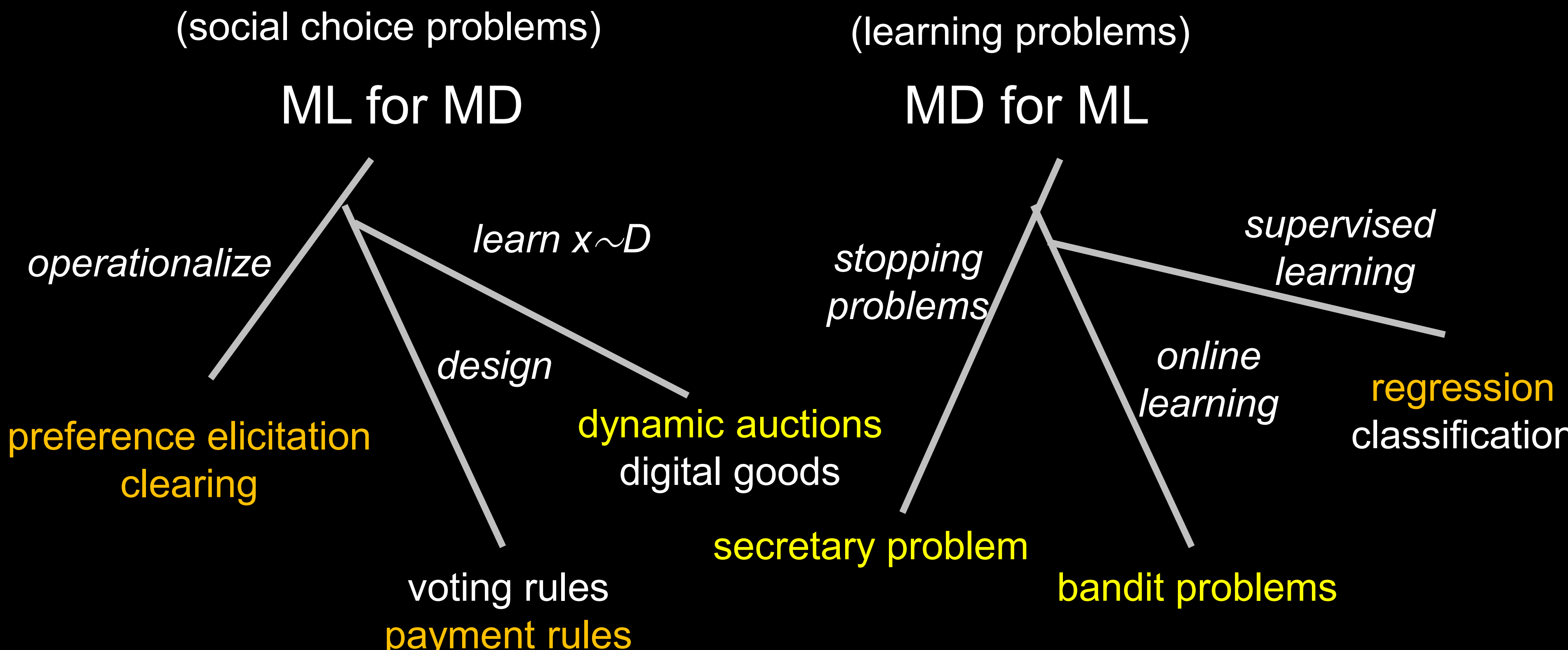
General Case

- For m>1 points, and D_i distribution, and absolute loss function
- Example. N={1,2}. Constant f(x)=c
- S_1={ (1,1), (2,1), ~~(3,0)~~ } (3,1)
- S_2 = { (4,0), (5,0), (6,1) }
- f(x)=0; R_1(f) = 2/3 ⟶ f(x)=1; R_1(f)=1/2

- Solution: *project and fit.* 3-competitive. ϵ-DSIC. Matching lower bound.

Other “MD for ML” problems

- Classification
- Reinforcement learning
- “Market of Minds”: Promote synergistic modular intelligence


(social choice problems)
ML for MD
operationalize
learn x∼D
design
preference elicitation
clearing
dynamic auctions
digital goods
voting rules
payment rules
(learning problems)
MD for ML
stopping problems
supervised learning
online learning
regression
classification
secretary problem
bandit problems