Payment Rules for Combinatorial Auctions via Structural Support Vector Machines

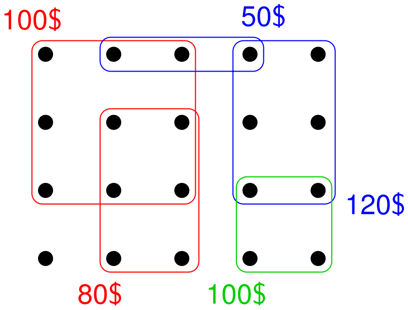
Felix Fischer
Harvard University

joint work with Paul Dütting (EPFL), Petch Jirapinyo (Harvard), John Lai (Harvard), Ben Lubin (BU), and David C. Parkes (Harvard)

September 7, 2011

Combinatorial Auctions

- n agents
- m items
- Bundles $Y = \{0,1\}^m$
- Valuation profiles $X = \mathbb{R}^{2^m \times n}$
- Allocation rule $g_i : X \to Y$
- Payment rule $t_i : X \times Y \to \mathbb{R}$

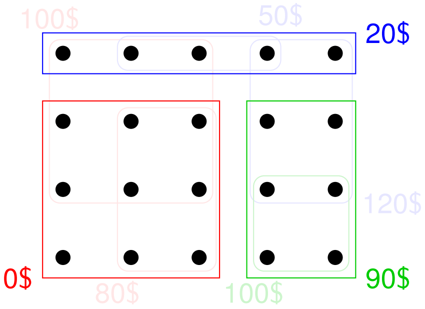


- Optimal allocation: maximize $\sum_i \mathbf{x}_i[\mathbf{y}_i]$ such that $\mathbf{y}_i \cap \mathbf{y}_j = \emptyset$
- Strategyproofness:

$$\mathbf{x}_i[g_i(\mathbf{x})] - t_i(\mathbf{x}, g_i(\mathbf{x})) \geq \mathbf{x}_i[g_i(\mathbf{x}_i', \mathbf{x}_{-i})] - t_i(\mathbf{x}_i', \mathbf{x}_{-i}, g_i(\mathbf{x}_i', \mathbf{x}_{-1}))$$

Combinatorial Auctions

- n agents
- m items
- Bundles $Y = \{0,1\}^m$
- Valuation profiles $X = \mathbb{R}^{2^m \times n}$
- Allocation rule $g_i : X \to Y$
- Payment rule $t_i : X \times Y \to \mathbb{R}$



- Optimal allocation: maximize $\sum_i \mathbf{x}_i[\mathbf{y}_i]$ such that $\mathbf{y}_i \cap \mathbf{y}_j = \emptyset$
- Strategyproofness:

$$\mathbf{x}_i[g_i(\mathbf{x})] - t_i(\mathbf{x}, g_i(\mathbf{x})) \geq \mathbf{x}_i[g_i(\mathbf{x}'_i, \mathbf{x}_{-i})] - t_i(\mathbf{x}'_i, \mathbf{x}_{-i}, g_i(\mathbf{x}'_i, \mathbf{x}_{-1}))$$

Problem Statement

- Elicitation of valuations and computation of optimal allocation are costly, often prohibitively so
- Canonical strategyproof mechanism: VCG
  - depends on ability to find efficient allocation
  - other problems: collusion, small or non-monotonic revenue
- Alternative solutions hard to come by
- Our approach: take allocation rule g as given, use to generate input for a learning algorithm
- Implicitly learns payment rule t that makes g "maximally incentive compatible" (we will see in what sense)

Outline

Combinatorial Auctions and Margin-Based Learning

Learning a Payment Rule

Summary and Open Problems

Learning What We Already Know

- By symmetry concentrate on agent 1, consider $g = g_1$ and $t = t_1$
- Assume g is given, as well as a distribution $P(X)$ on X
- Together they induce a distribution $P(X, Y)$ on $X \times Y$

- Sample set of training examples from $P(X, Y)$ and learn an allocation function $h : X \to Y$

Learning What We Already Know

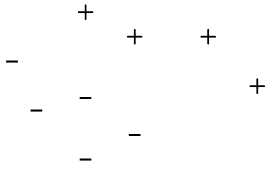
- By symmetry concentrate on agent 1, consider $g = g_1$ and $t = t_1$
- Assume g is given, as well as a distribution $P(X)$ on X
- Together they induce a distribution $P(X, Y)$ on $X \times Y$

- Sample set of training examples from $P(X, Y)$ and learn an allocation function $h : X \to Y$

- We know g, so we are not actually interested in h
- Rather: employ a margin-based learning method, infer t from the margin

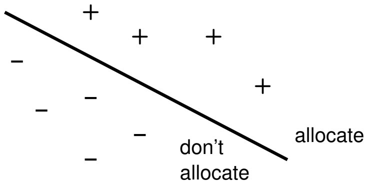
Learning How to Allocate

- Single-item case corresponds to an ordinary binary classifier: allocate the item or not



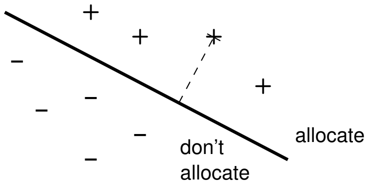
Learning How to Allocate

- Single-item case corresponds to an ordinary binary classifier: allocate the item or not



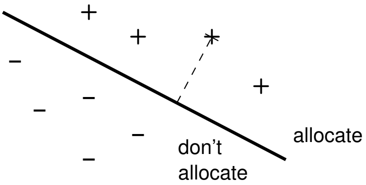
Learning How to Allocate

- Single-item case corresponds to an ordinary binary classifier: allocate the item or not



Learning How to Allocate

- Single-item case corresponds to an ordinary binary classifier: allocate the item or not



- In general: one class for each bundle that could be allocated
- Learn a discriminant function $f : X \times Y \to \mathbb{R}$ that rates bundles
- Define h to choose the most appropriate bundle:

$$h(\mathbf{x}) = \arg\max_{\mathbf{y} \in Y(x_{-1})} f(\mathbf{x}, \mathbf{y})$$

The Discriminant Function

- Impose additional structure on f:

$$f_{\mathbf{w}}(\mathbf{x}, \mathbf{y}) = w_1 \mathbf{x}_1[\mathbf{y}] + \mathbf{w}_{-1}^T \psi(\mathbf{x}_{-1}, \mathbf{y})$$

  - $\mathbf{w} = (w_1, \mathbf{w}_{-1}) \in \mathbb{R}^{M+1}$ is a parameter vector to be learned
  - $\psi(\mathbf{x}_{-1}, \mathbf{y}) \in \mathbb{R}^M$ is a feature vector derived from $\mathbf{x}_{-1}$ and $\mathbf{y}$

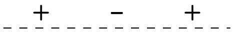
The Discriminant Function

- Impose additional structure on f:

$$f_{\mathbf{w}}(\mathbf{x}, \mathbf{y}) = w_1 \mathbf{x}_1[\mathbf{y}] + \mathbf{w}_{-1}^T \psi(\mathbf{x}_{-1}, \mathbf{y})$$

  - $\mathbf{w} = (w_1, \mathbf{w}_{-1}) \in \mathbb{R}^{M+1}$ is a parameter vector to be learned
  - $\psi(\mathbf{x}_{-1}, \mathbf{y}) \in \mathbb{R}^M$ is a feature vector derived from $\mathbf{x}_{-1}$ and $\mathbf{y}$

- Linear in $\mathbb{R}^M$, but can be very expressive in X



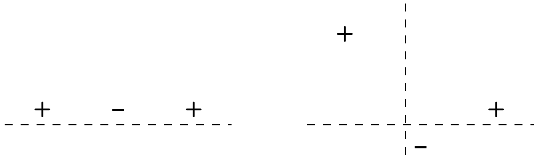
The Discriminant Function

- Impose additional structure on f:

$$f_{\mathbf{w}}(\mathbf{x}, \mathbf{y}) = w_1 \mathbf{x}_1[\mathbf{y}] + \mathbf{w}_{-1}^T \psi(\mathbf{x}_{-1}, \mathbf{y})$$

  - $\mathbf{w} = (w_1, \mathbf{w}_{-1}) \in \mathbb{R}^{M+1}$ is a parameter vector to be learned
  - $\psi(\mathbf{x}_{-1}, \mathbf{y}) \in \mathbb{R}^M$ is a feature vector derived from $\mathbf{x}_{-1}$ and $\mathbf{y}$

- Linear in $\mathbb{R}^M$, but can be very expressive in X



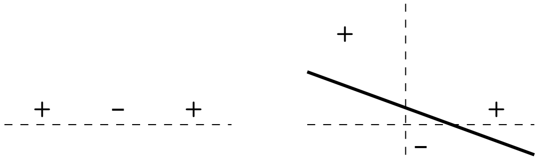
The Discriminant Function

- Impose additional structure on f:

$$f_{\mathbf{w}}(\mathbf{x}, \mathbf{y}) = w_1 \mathbf{x}_1[\mathbf{y}] + \mathbf{w}_{-1}^T \psi(\mathbf{x}_{-1}, \mathbf{y})$$

  - $\mathbf{w} = (w_1, \mathbf{w}_{-1}) \in \mathbb{R}^{M+1}$ is a parameter vector to be learned
  - $\psi(\mathbf{x}_{-1}, \mathbf{y}) \in \mathbb{R}^M$ is a feature vector derived from $\mathbf{x}_{-1}$ and $\mathbf{y}$

- Linear in $\mathbb{R}^M$, but can be very expressive in X



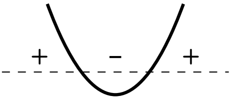
The Discriminant Function

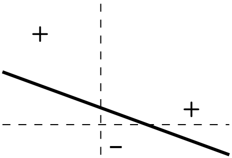
- Impose additional structure on f:

$$f_{\mathbf{w}}(\mathbf{x}, \mathbf{y}) = w_1 \mathbf{x}_1[\mathbf{y}] + \mathbf{w}_{-1}^T \psi(\mathbf{x}_{-1}, \mathbf{y})$$

  - $\mathbf{w} = (w_1, \mathbf{w}_{-1}) \in \mathbb{R}^{M+1}$ is a parameter vector to be learned
  - $\psi(\mathbf{x}_{-1}, \mathbf{y}) \in \mathbb{R}^M$ is a feature vector derived from $\mathbf{x}_{-1}$ and $\mathbf{y}$

- Linear in $\mathbb{R}^M$, but can be very expressive in X





The Payment Rule

- Ensure $w_1 > 0$ and let

$$t_{\mathbf{w}}(x, y) = -\left(\frac{\mathbf{w}_{-1}}{w_1}\right)^T \psi(x_{-1}, y)$$

  - agent-independent
  - $h_{\mathbf{w}}$ predicts the utility-maximizing bundle:

$$\begin{aligned} h_{\mathbf{w}}(x) &= \arg\max_{y \in Y(x_{-1})} f_{\mathbf{w}}(x, y) = \arg\max_{y \in Y(x_{-1})} w_1 \mathbf{x}_1[\mathbf{y}] + \mathbf{w}_{-1}^T \psi(\mathbf{x}_{-1}, y) \\ &= \arg\max_{y \in Y(x_{-1})} w_1 \mathbf{x}_1[\mathbf{y}] + \mathbf{w}_{-1}^T \left(-\frac{w_1}{\mathbf{w}_{-1}} t_{\mathbf{w}}(x, y)\right) \\ &= \arg\max_{y \in Y(x_{-1})} (x_1[y] - t_{\mathbf{w}}(x, y)) \end{aligned}$$

- Can ensure by translation that $\mathbf{w}_{-1}^T \psi(\mathbf{x}_{-1}, \mathbf{0}) = 0$, *i.e.*, that payment for empty bundle is zero

Truthfulness and Regret

- Looks like the characterization of a strategyproof mechanism, but $h_{\mathbf{w}}$ might not be feasible
- Also recall that we want to allocate according to g, not $h_{\mathbf{w}}$

Truthfulness and Regret

- Looks like the characterization of a strategyproof mechanism, but $h_{\mathbf{w}}$ might not be feasible
- Also recall that we want to allocate according to g, not $h_{\mathbf{w}}$
- Ex-post regret (for bidding truthfully): maximum gain in utility by bidding differently

Lemma: The ex-post regret for bidding truthfully in $(g, t_{\mathbf{w}})$ is

$$\frac{1}{w_1}\left(\max_{\mathbf{y}' \in Y(\mathbf{x}_{-1})} f_{\mathbf{w}}(\mathbf{x}, \mathbf{y}') - f_{\mathbf{w}}(\mathbf{x}, g(\mathbf{x}))\right).$$

Theorem: If $h_{\mathbf{w}}$ is exact, then $(g, t_{\mathbf{w}})$ is strategyproof.

Truthfulness and Regret

- Looks like the characterization of a strategyproof mechanism, but $h_{\mathbf{w}}$ might not be feasible
- Also recall that we want to allocate according to g, not $h_{\mathbf{w}}$
- Ex-post regret (for bidding truthfully): maximum gain in utility by bidding differently

Lemma: The ex-post regret for bidding truthfully in $(g, t_{\mathbf{w}})$ is

$$\frac{1}{w_1}\left(\max_{\mathbf{y}' \in Y(\mathbf{x}_{-1})} f_{\mathbf{w}}(\mathbf{x}, \mathbf{y}') - f_{\mathbf{w}}(\mathbf{x}, g(\mathbf{x}))\right).$$

Theorem: If $h_{\mathbf{w}}$ is exact, then $(g, t_{\mathbf{w}})$ is strategyproof.

- But: $h_{\mathbf{w}}$ will not always be exact, we know it cannot be if g is not monotonic

Regret and Generalization Error

- Generalization error of a classifier $h_{\mathbf{w}} \in \mathcal{H}_{\psi}$:

$$R_P(h_{\mathbf{w}}) = \int_{X \times Y} \Delta_{\mathbf{x}}(\mathbf{y}, h_{\mathbf{w}}(\mathbf{x}))\, dP(\mathbf{x}, \mathbf{y})$$

where $\Delta_{\mathbf{x}}(\mathbf{y}, \mathbf{y}') = \frac{1}{w_1}(f_{\mathbf{w}}(\mathbf{x}, \mathbf{y}') - f_{\mathbf{w}}(\mathbf{x}, \mathbf{y}))$

Theorem: If $h_{\mathbf{w}}$ minimizes generalization error then $t_{\mathbf{w}}$ minimizes expected ex-post regret for truthful bidding.

- Amount a random agent can gain by lying when all others tell the truth, for valuations drawn from $P(X)$
- Different from (approximate) ex-ante and ex-interim equilibrium, rather provides an upper bound on the expected ex-interim gain

Support Vector Machines?

- Learn a discriminant function that maximizes the margin
- Binary setting: minimize generalization error in the limit

- Version with structured/multi-class output due to Joachims et al.
- Training by solving a quadratic optimization problem with linear constraints, can be done efficiently under certain conditions

- Training requires computation of inner products in the (high- or infinite-dimensional) feature space $\mathbb{R}^M$
- Kernel trick: choose ψ carefully to ensure they can be computed efficiently from vectors in the original space
- Linear classification in $\mathbb{R}^M$ without any explicit calculations in $\mathbb{R}^M$

Summary

- Design of payment rules using margin-based classifier, given oracle access to valuation distribution and allocation rule
- Exact classifier yields strategyproof payment rule, minimization of error implies minimization of expected ex-post regret
- Experiments for 5 items, 2 to 6 agents, 200 training examples
  - 5 items, 2 to 6 agents, 200 training examples
  - $\psi(x_{-1}, y) = \phi([x_2 \setminus y, \ldots, x_n \setminus y])$
  - ϕ corresponding to RBF kernel $K(z, z') = \exp(-\|z - z'\|/2\sigma^2)$

	accuracy	average regret	IR violation
single item	96%	0.2%	2%
single-minded	90%	1%	6%
multi-minded, complements	94%	0.1%	3%
multi-minded, substitutes	75%	2%	15%

Open Problems

- Possibly $-\mathbf{w}_{-1}^T \psi(\mathbf{x}_{-1}, \mathbf{y}) \geq \mathbf{x}_1[\mathbf{y}]$, failure of individual rationality
  - tradeoff between individual rationality and strategyproofness
  - both at the same time (only?) by deviation from g, *e.g.*, by discarding $\mathbf{y}$ and allocating $\emptyset$
- Training problem has $\Omega(|Y(x_{-1})|)$ constraints, exponential in m in general
  - only polynomially many constraints matter, a separation oracle would suffice
  - when valuations can be represented succinctly, payment monotonicity would also suffice
  - more highly structured payment rules for restricted valuations
- More clever feature maps, *e.g.*, to allow for generalization across different numbers of agents

Thank you!