Advances in Deep Learning for Auction Design:

Fairness, Robustness, and Expressiveness

Michael J. Curry

Joint work with

Ping-yeh Chiang, Samuel Dooley, John Dickerson, Tom Goldstein, Elizabeth Horishny, Kevin Kuo, Uro Lyi, and Anthony Ostuni

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Market design is a kind of economic engineering, utilizing laboratory research, game theory, algorithms, simulations, and more.

Its challenges inspire us to rethink longstanding fundamentals of economic theory.

– Paul Milgrom, presentation when awarded 2008 Erwin Plein Nemmers Economics Prize

Market Design*

*for the sake of this talk, at least!

Market design is partially based on **mechanism design**, i.e., the design of incentives, toward a particular goal, in a strategic setting with one or more agents.

Design of incentives:

- Pricing and allocation rules in auctions
- Joining an agent to a contract in a matching market

Toward a particular goal:

- Maximize global welfare
- Ensure some notion of fairness
- Maximize revenue of the clearinghouse aka principal

Strategic settings:

- Agents' strategies are chosen knowing that their outcome will depend also on other agents' choices

"Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel" in 2020 (Paul Milgrom & Robert Wilson); 2012 (Al Roth & Lloyd Shapley); 2007 (Roger Myerson, Leonid Hurwicz, & Eric Maskin).

~~Market~~ Auction Design*

*for the sake of this talk, at least!

An auction mechanism is run by a central coordinator, who asks agents to report their preferences and then aggregates them in some way before deciding on an allocation and payments

Agents have private information about preferences (i.e., their private type θ), and can choose to lie about them to the coordinator

The distribution over which they draw their types is known to all.

Auction Model*

*for the sake of this talk, at least!

Items: $j = 1, 2, \ldots, k$
Agents: $i = 1, 2, \ldots, n$

Publicly known valuation distribution $P(\boldsymbol{v}_i)$

Private valuations $\boldsymbol{v}_i \in \mathbb{R}^k$

The mechanism outputs allocations of items, and a payment to charge each player

Players receive utility based on allocations, payments, and their private valuation.

$$\begin{tikzcd}
P(\boldsymbol{v}_i) \arrow[d] & & & \\
\boldsymbol{v}_i \in \mathbb{R}^k \arrow[dd] \arrow[rrrdd, bend left] & & & \\
 & & \boldsymbol{a}_i \in \mathbb{R}^k \arrow[dr] & \\
\boldsymbol{b}_i \arrow[r] & f_i(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \arrow[ur] \arrow[dr] & & u_i = \sum_{j=1}^{k} a_{ij} v_{ij} - p_i \\
 & & p_i \in \mathbb{R} \arrow[ur] &
\end{tikzcd}$$

Players strategically choose bids and send them to the allocation mechanism f

(Utilities may also take forms other than quasilinear.)

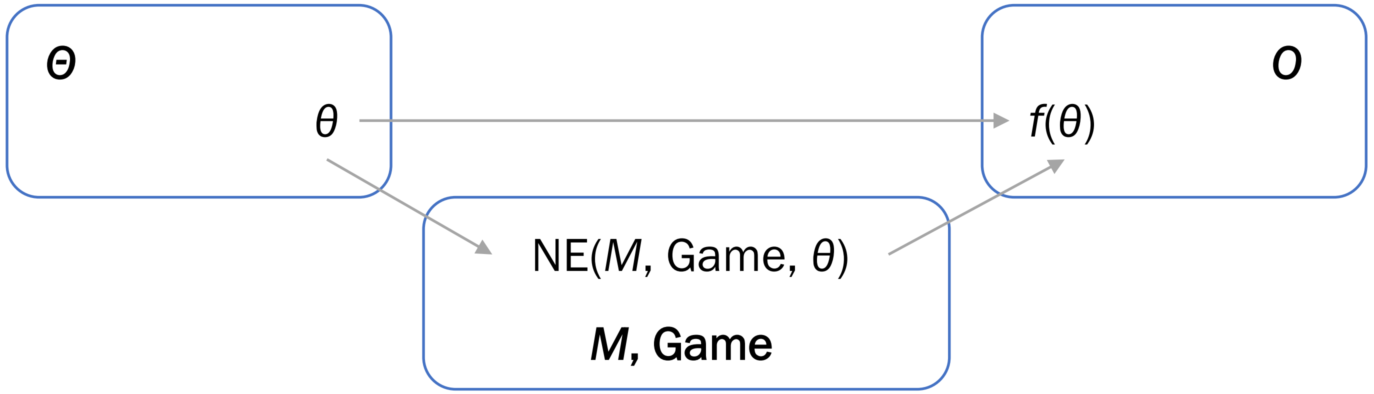
Mechanism Design & Equilibrium Behavior

Common assumptions of mechanism design:

- Agents are **rational utility maximizers** (rational in a very strong sense)
- Will play a (Bayes-)Nash equilibrium

Designing a mechanism with "good" equilibrium behavior is hard.

- Here, "good" means our mechanism **implements** a function *f* (e.g., social welfare maximization) under agents' strategic behavior.



Desirable Properties of Auctions

Individual rationality (IR): nobody who is truthful ever pays more than their expected value for the allocation

Dominant-strategy incentive compatible (DSIC): it is always optimal to bid your true valuation, no matter what others do:

$$\forall \boldsymbol{v}_{-i}: rgt_i = \max_{\boldsymbol{b}_i} u_i(\boldsymbol{b}_i, \boldsymbol{v}_{-i}) - u_i(\boldsymbol{v}_i, \boldsymbol{v}_{-i}) = 0$$

Revenue maximization: want $\sum_i p_i$ to be as large as possible

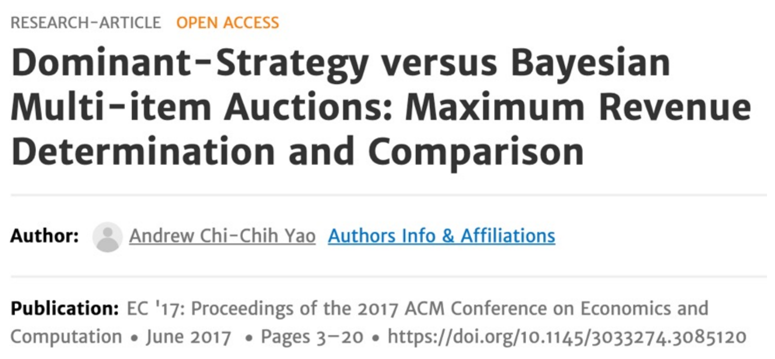
Optimal Auction Design I

An **optimal auction** is incentive compatible and revenue-maximizing.

Analytic results are **very tough** to create:

- Myerson [1981] resolved for 1 item!
- Some progress since then, but 2 items & 2 bidders is not fully resolved, still …

Many nice recent works (from Econ, CS, Math, OR/MS, …), typically focus on BNIC instead of DSIC, though …



RESEARCH-ARTICLE OPEN ACCESS

Dominant-Strategy versus Bayesian Multi-item Auctions: Maximum Revenue Determination and Comparison

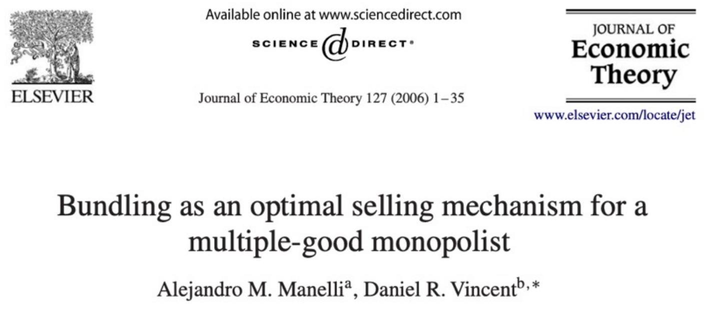
Author: Andrew Chi-Chih Yao Authors Info & Affiliations

Publication: EC '17: Proceedings of the 2017 ACM Conference on Economics and Computation • June 2017 • Pages 3–20 • https://doi.org/10.1145/3033274.3085120

Myerson [1981]: DSIC = BNIC revenue, 1 item case

Yao [2017]: DSIC ≠ BNIC revenue, 2 item case
&
Exact formulaic results for 2 items, distribution with support of size 2.

Optimal Auction Design II



Available online at www.sciencedirect.com

SCIENCE DIRECT

ELSEVIER

Journal of Economic Theory 127 (2006) 1–35

JOURNAL OF Economic Theory

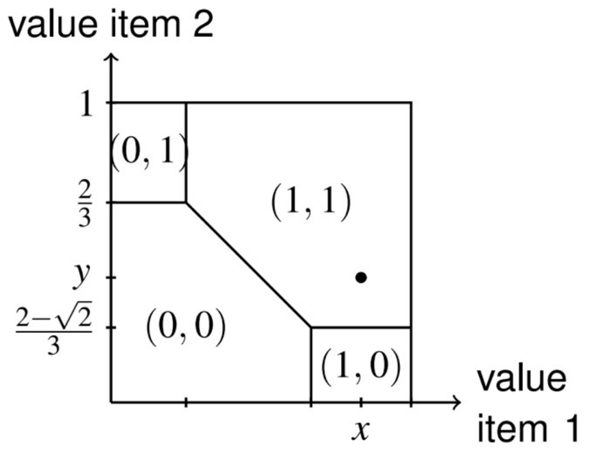
www.elsevier.com/locate/jet

Bundling as an optimal selling mechanism for a multiple-good monopolist

Alejandro M. Manelli[a], Daniel R. Vincent[b,*]

Analytic results are **very tough** to create.

- 2 item, 1 bidder with additive valuations [Manelli-Vincent 2006, Giannakopoulos&Koutsoupias 2015, Haghpanah&Hartline 2015, Daskalakis+ 2017, ..]
- 2 items, unit-demand bidders [Pavlov 2011]



Something to keep in mind:

- Not allocating an item → not okay in social welfare maximization
- Not allocating an item → sometimes okay in revenue maximization

Intuition: you get nothing for selling nothing, but you can "beat second price/VCG" if you do end up selling it

What If There Are Multiple Items and More Than 1 Bidder …?

Not totally out of luck!

1. **Informed** by theoretical analyses of smaller, general systems
2. Can design **approximately optimal** mechanisms
3. Can use **automated** mechanism design [Conitzer&Sandholm 2002]

We're most motivated by the general pitch of (3) – "automatically" designing a mechanism for a specific problem instance, but …

… older methods run into severe scalability issues, so not practical.

Recall: Auction Model*

*for the sake of this talk, at least!

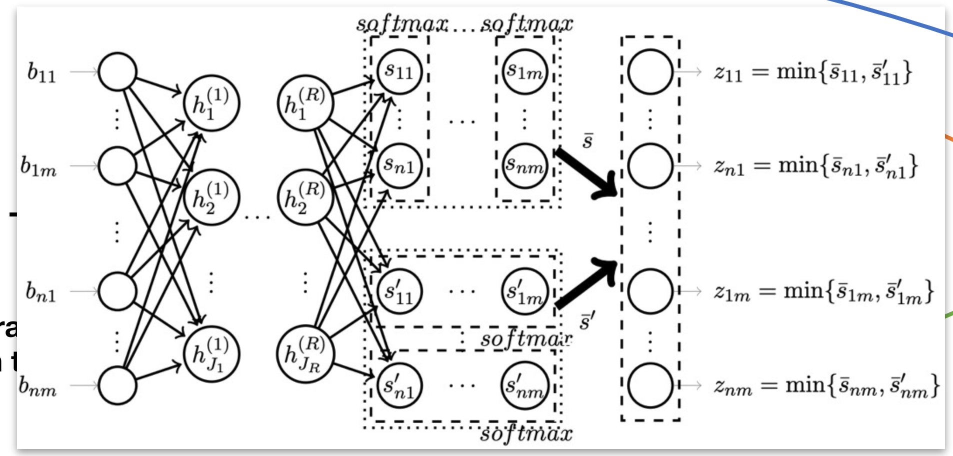
Publicly known valuation distribution $P(\boldsymbol{v}_i)$

Private valuations $\boldsymbol{v}_i \in \mathbb{R}^k$

$\boldsymbol{b}_i$

Players stra
send them t

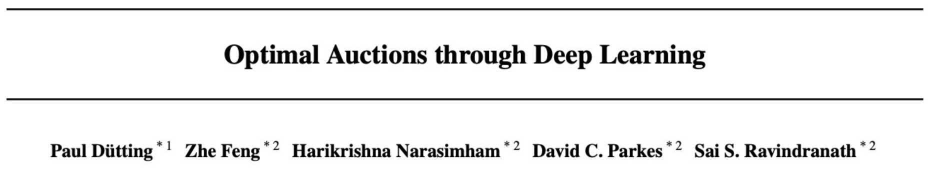
The mechanism outputs allocations of items, and a payment to charge each player



Players receive utility based on allocations, payments, and their private valuation.

$$u_i = \sum_{j=1}^{k} a_{ij} v_{ij} - p_i$$

Differentiable Economics



Optimal Auctions through Deep Learning

Paul Dütting[*1] Zhe Feng[*2] Harikrishna Narasimham[*2] David C. Parkes[*2] Sai S. Ravindranath[*2]

Differentiable programming — term coined by Yann LeCun

- “OK, Deep Learning has outlived its usefulness as a buzz-phrase. Deep Learning est mort. Vive Differentiable Programming! …Yeah, Differentiable Programming is little more than a rebranding of the modern collection Deep Learning techniques…”
- “But the important point is that people are now building a new kind of software by assembling networks of parameterized functional blocks and by training them from examples using some form of gradient-based optimization. “

Differentiable economics — term coined in same spirit by David Parkes

- Combines the same basic building blocks to create mechanisms that are differentiable, allowing them to be optimized using gradient descent

RegretNet (and other, similar networks)

Focus on optimal auctions: DSIC (i.e., strategyproof) & revenue maximizing

Dütting et al, “Optimal Auctions Through Deep Learning”: parameterize auction mechanism (function *f* from bids to winners/payments in previous slide) as deep neural network:

- Maximization of revenue → include a revenue term in the loss function during training
- Strategyproofness constraint → compute strategic inputs via gradient ascent, train on these to reduce how much strategyproofness is violated

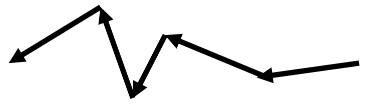
Estimating Regret

Given *f* and a draw *v_i*, how to compute $\max_{\boldsymbol{b}_i} u_i(\boldsymbol{b}_i, \boldsymbol{v}_{-i}) - u_i(\boldsymbol{v}_i, \boldsymbol{v}_{-i})$?

Just do gradient ascent on utility!

Networks and inputs relatively small, so can do many steps (25 train time, 1000 test time)

- (Easily implemented in PyTorch by just setting `requires_grad=True` on input tensors)

$\underset{\boldsymbol{b}_i}{\mathrm{arg}max}\, rgt_i(\boldsymbol{v}_i) \approx$  $\nabla_{\boldsymbol{b}_i} u_i(\boldsymbol{b}_i, \boldsymbol{v}_{-i})$

Learning Procedure

Dataset is a large number of randomly sampled valuation profiles $\boldsymbol{v}$

- Our end goal is a strategyproof mechanism; if we have that, agents bid truthfully; so, only need to ensure regret = 0 on truthful bids!

Loss on a single valuation profile:

Penalize low profit

Penalize high regret

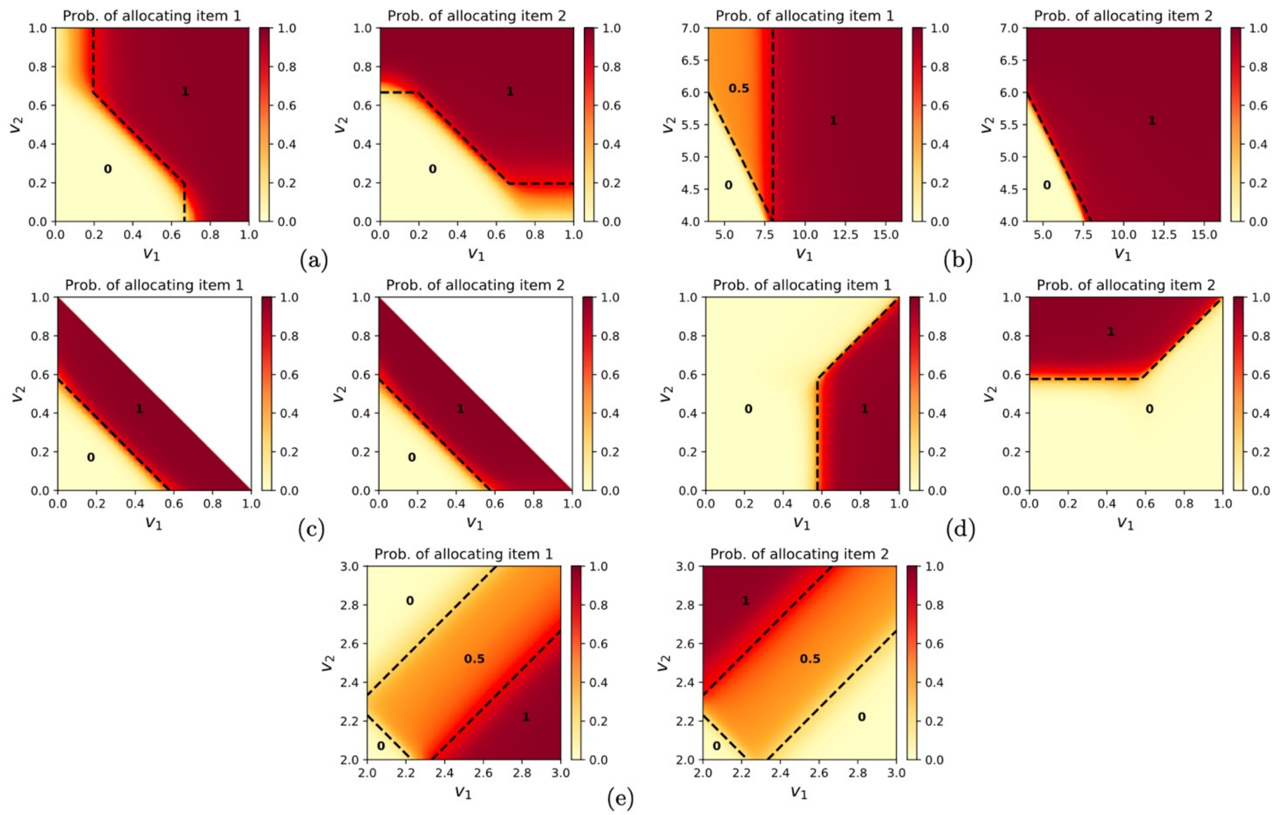
$$L(\boldsymbol{v}, f(\boldsymbol{v})) = -\sum_i p_i + \sum_i \lambda_i \mathrm{rgt}_i(\boldsymbol{v}) + \frac{\rho}{2}\left(\sum_i \mathrm{rgt}_i(\boldsymbol{v})\right)^2$$

Minimize *L* by minibatch SGD, just like any neural network

Successful Results

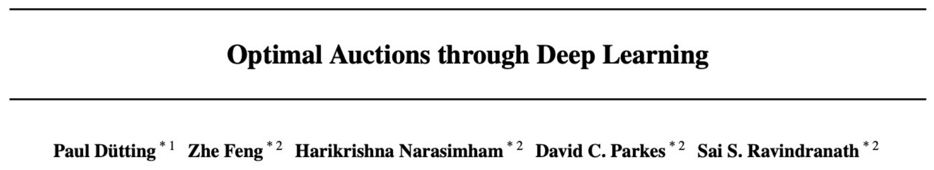
Optimal Auctions through Deep Learning

Paul Dütting [*1] **Zhe Feng** [*2] **Harikrishna Narasimham** [*2] **David C. Parkes** [*2] **Sai S. Ravindranath** [*2]



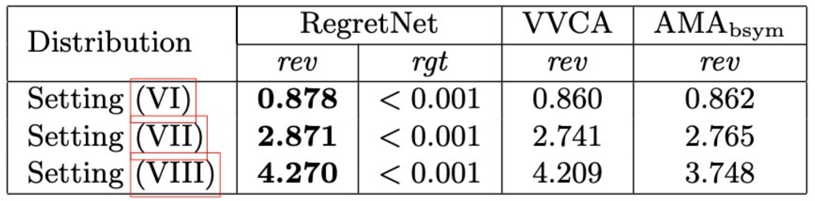
Dotted lines denote theoretically optimal mechanism; orange/red is what neural networks learned after training

Successful Results

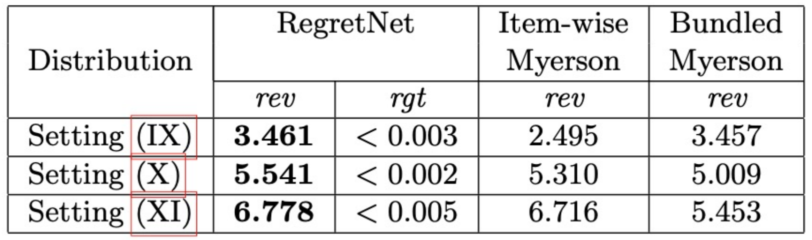


Optimal Auctions through Deep Learning

Paul Dütting [*1] Zhe Feng [*2] Harikrishna Narasimham [*2] David C. Parkes [*2] Sai S. Ravindranath [*2]

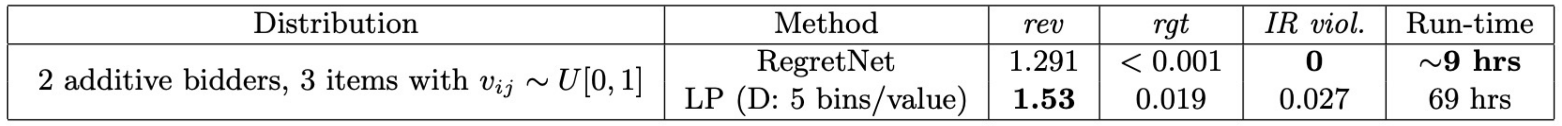


Distribution	RegretNet		VVCA	AMA_{bsym}
	rev	*rgt*	*rev*	*rev*
Setting (VI)	**0.878**	< 0.001	0.860	0.862
Setting (VII)	**2.871**	< 0.001	2.741	2.765
Setting (VIII)	**4.270**	< 0.001	4.209	3.748



Distribution	RegretNet		Item-wise Myerson	Bundled Myerson
	rev	*rgt*	*rev*	*rev*
Setting (IX)	**3.461**	< 0.003	2.495	3.457
Setting (X)	**5.541**	< 0.002	5.310	5.009
Setting (XI)	**6.778**	< 0.005	6.716	5.453

Their approach also beats a variety of strong baselines in more complicated situations where the optimal mechanism is not known.



Distribution	Method	*rev*	*rgt*	*IR viol.*	Run-time
2 additive bidders, 3 items with $v_{ij} \sim U[0,1]$	RegretNet	1.291	< 0.001	**0**	**~9 hrs**
	LP (D: 5 bins/value)	**1.53**	0.019	0.027	69 hrs

New Takes on Differentiable Economics

ProportionNet: Balancing Fairness and Revenue For Auction Design With Deep Learning (*u.r.* 2021)

- Impose **fairness constraints** on allocations

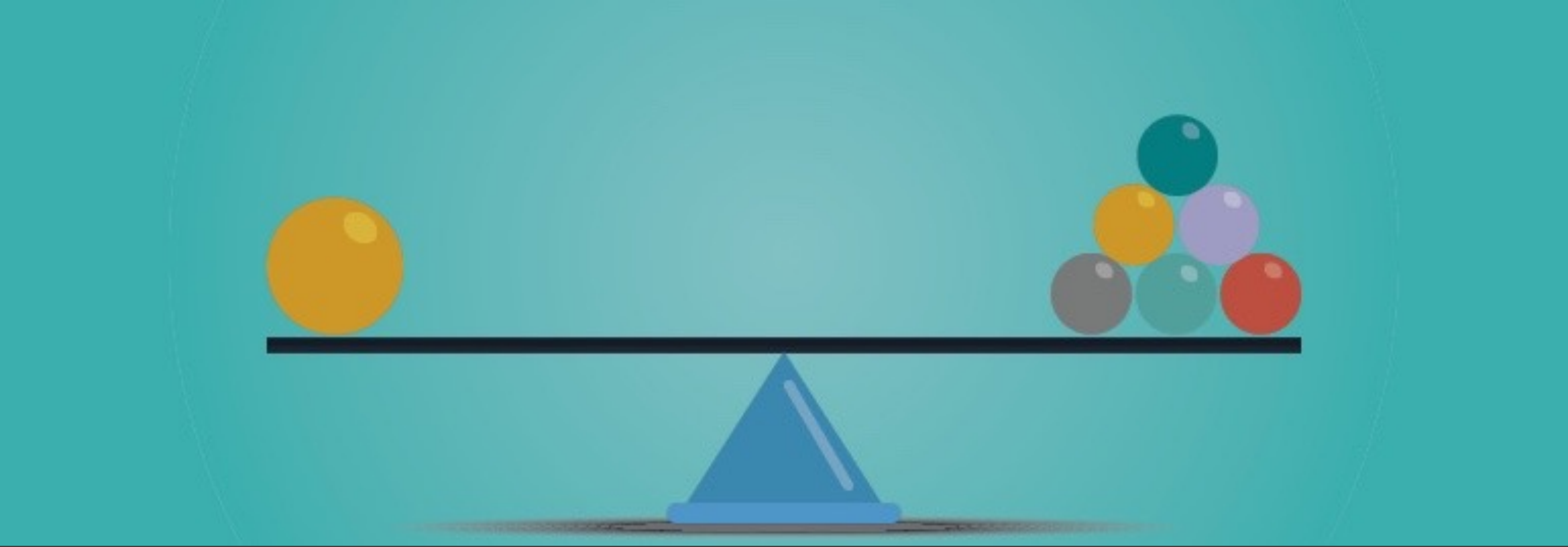
Certifying Strategyproof Auction Networks (NeurIPS 2020)

- Compute regret **exactly** — no gradient-based approximation

Learning Revenue-Maximizing Auctions With Differentiable Matching (*u.r.* 2021)

- Sinkhorn algorithm to perform differentiable matching; allows for **new demand types** (e.g., *k*-unit demand)

Curry*, Chiang*, Goldstein, Dickerson. **Certifying Strategyproof Auction Networks.** NeurIPS-20.
Kuo*, Ostuni*, Horishny*, Curry*, Dooley*, Chiang*, Goldstein, Dickerson. **ProportionNet: Balancing Fairness and Revenue for Auction Design with Deep Learning.** Under review, 2021.
Curry*, Lyi*, Goldstein, Dickerson. **Learning Revenue-Maximizing Auctions With Differentiable Matching.** Under review, 2021.



ProportionNet

Unfairness in Advertising

Recent work showcases **discrimination** in online advertising

- Google displayed job listings of a higher income to men, and lower income job listings to women [Datta et al. 2015].
- Facebook sent ads discriminating on gender regardless of advertiser preferences [Muhammad et al. 2019].
- Online ads have been shown to discriminate on race, history of drug use, parent status, etc [Chawla et al. 2020].

Unintentional, but not necessarily inevitable.

New Goal: prevent unfair* allocations between users.

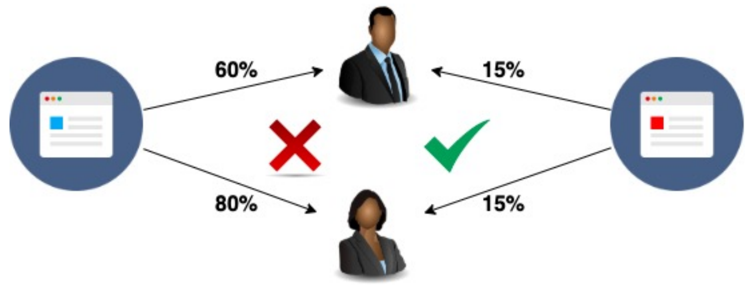
*for some definition of "unfair" – this is, of course, a morally-laden step to take and deciding on which definition of fairness, if any, should be incorporated into an automated decisioning system is one that should, necessarily, involve stakeholder discussion.

A Specific Application of Auction Design

Auctions are used by many platforms to decide online ad allocations.

- Advertiser *i* bids for a user *j*, and the auctioneer allocates users (i.e., “items”) depending on bids.
- Other qualities, such as relevancy, impact allocation decisions as well.

We would like to prevent these allocations from inducing unfairness.



Past real-world experimentation surfaces that online advertising systems show ads (e.g., career advancement opportunities) at different rates to equally-qualified individuals.

Our Definition of Fairness

Total Variation Fairness: users who are similar should be allocated advertisements similarly [Dwork+ 2012, Chawla+ 2020].

- The more similar users are, the smaller their distance *D*. Similar users must be treated similarly.

Between two users *j* and *j'*, the allocation vector *z* is fair if:

$$\sum_{i \in N} |z_{ij} - z_{ij'}| \leq D(j, j')$$

For our setting: *D* should take into account relevant differences but ignore differences in protected attributes.

We use total variation fairness as an example. However, again, what constitutes "fair" is up to the auctioneer and other stakeholders impacted by the system.

ProportionNet’s Loss Function

RegretNet-style loss:

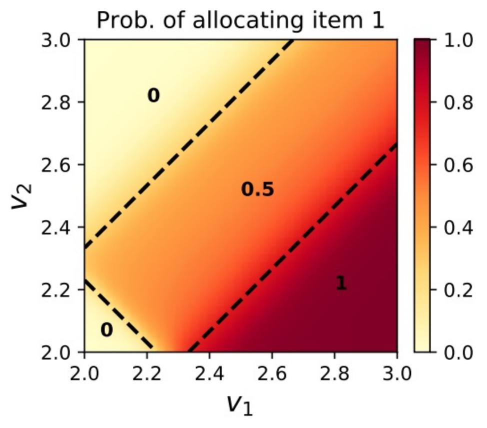
ProportionNet loss:

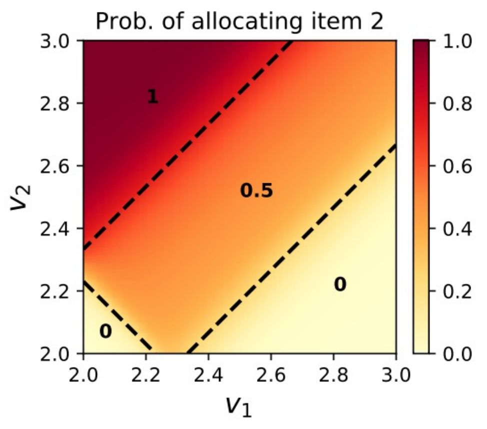
Total Variation Fairness:

(Network architecture and training can stay the same — we are still enforcing strategyproofness and maximizing revenue.)

Sample Results

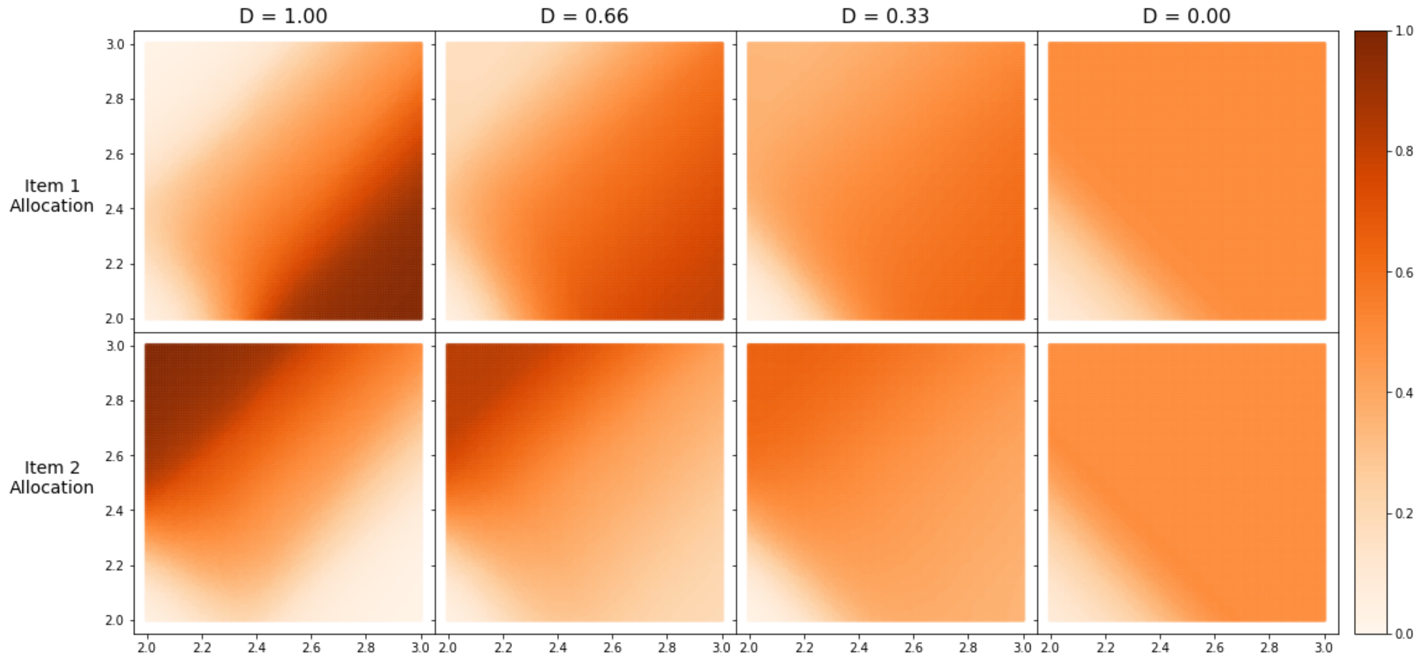
Results from original RegretNet
(Unit demand, U[2,3], 2 items):





ProportionNet in same setting
with increasingly strong
fairness constraints:

← Efficiency Fairness →



ProportionNet: Discussion

We present one way to learn auctions with fairness constraints on allocations.

- Amenable to user “knob turning” – with regard to the intensity of the fairness regularizer & the definition of fairness itself.

Measuring expected unfairness can’t be done directly (we must estimate from samples of valuation profiles).

- Similar to Dütting [2019], we can bound generalization error & show that our sample estimate is a good upper bound of true expected unfairness

Our model of fairness is motivated by real advertising auctions, but it is still extremely stylized. Discussion with real policymakers and stakeholders is important.

Certifying Strategyproof Auction Networks

Limitation of RegretNet (& ProportonNet)

The RegretNet family is concerned with minimizing regret.

- Gradient-based approximation: good for training, but may underestimate regret.

We would rather compute regret **exactly** — but how?

- Integer programming (ILP) techniques from robustness literature

ILP for Neural Networks

Modern neural networks are mostly piecewise linear (ReLU)

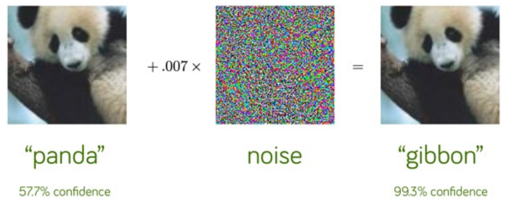
- It is possible to embed them into integer programs using big-M formulation for nonlinear activations

One can then maximize over inputs

- Often: worst-case adversarial image
- Our case: highest-regret bid

$$\hat{\boldsymbol{x}}_{i+1} = W_i \boldsymbol{x}_i + \boldsymbol{b}_i$$
$$\boldsymbol{x}_{i+1} = \max(0, \hat{\boldsymbol{x}}_{i+1})$$

$$\begin{aligned} \boldsymbol{\delta}_i \in \{0,1\}^d, \quad & \boldsymbol{x}_i \geq 0, \quad && \boldsymbol{x}_i \leq \boldsymbol{u}_i \boldsymbol{\delta}_i \\ & \boldsymbol{x}_i \geq \hat{\boldsymbol{x}}_i, && \boldsymbol{x}_i \leq \hat{\boldsymbol{x}}_i - \boldsymbol{l}_i(1 - \boldsymbol{\delta}_i) \end{aligned}$$



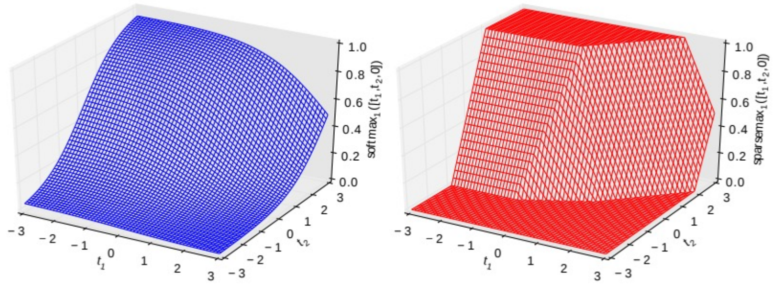
Challenge: Allocation Constraints

RegretNet uses a softmax to enforce allocation constraints (must sum to 1).

- No way to get rid of it, but can’t fit directly into integer **linear** program.

Solution: replace with **sparsemax**

- Formulate as optimization problem (projection), embed KKT conditions as additional constraints



$$\arg\min_z \frac{1}{2}\|z - x\|_2^2 \text{ s.t.}$$
$$1^T z - 1 = 0$$
$$0 < z < 1$$

$$(z - x) + \mu_1 - \mu_2 + \lambda 1 = 0$$
$$z - 1 \leq 0, -z \leq 0, 1^T z - 1 = 0$$
$$\mu_1 \geq 0, \mu_2 \geq 0$$
$$\mu_1(z - 1) = 0, \mu_2(-z) = 0$$

(technically SOS1 not linear but still easy to deal with...)

Challenge: Individual Rationality

RegretNet's architecture enforces individual rationality exactly

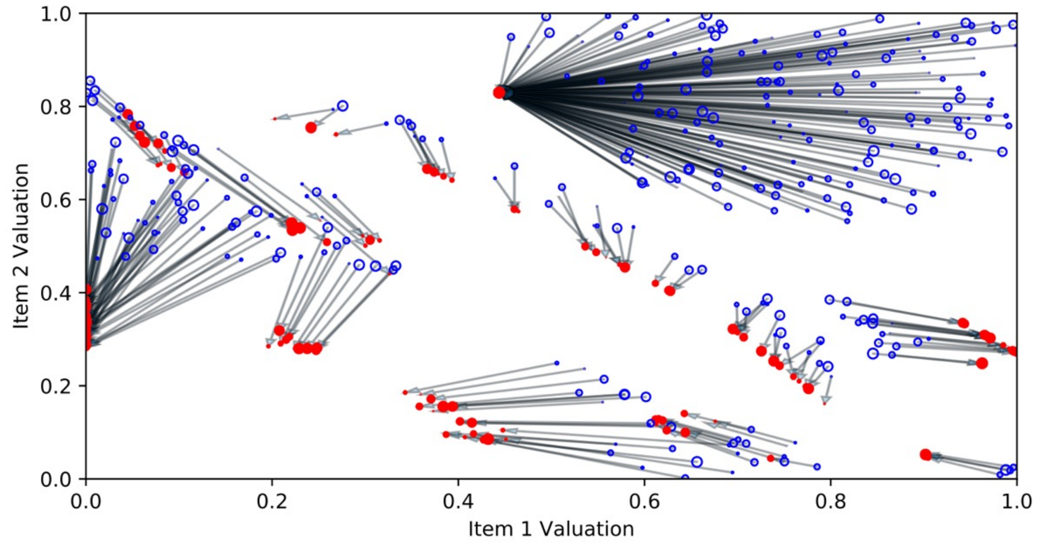
- In an ILP, this is a (nonconvex) bilinear equality constraint (product of two disjoint variables, i.e., "charge payment iff allocated item").

Two solutions we've tried:

- Eliminate it and add a penalty term to encourage IR instead.
- Surprisingly, the newest version of Gurobi* can deal with these …

* Gurobi 9.1 and later: "Bilinear constraints are a special case of non-convex quadratic constraints, and the algorithms Gurobi uses to handle the latter are also well suited to solving bilinear programming problems."

Experimental Results: Certifying RegretNet



2 items, 1 bidder, normal RegretNet: blue circles are truthful bids, red circles are (non-truthful) bids that result in gain >0.005 computed by our certifier.

Auction Setting	IR	Relu Reg.	Solve time (s)	Revenue	Empirical Regret	Certified Regret	Emp./Cert. Regret
1x2	Yes	No	25.6 (72.0)	0.593 (0.404)	0.014 (0.012)	0.019 (0.016)	0.731
1x2	Yes	Yes	7.2 (17.5)	0.569 (0.390)	0.003 (0.002)	0.004 (0.003)	0.700
1x2	No	Yes	0.034 (0.007)	0.568 (0.398)	0.009 (0.005)	0.011 (0.004)	0.839
2x2	Yes	No	13.9 (37.0)	0.876 (0.286)	0.009 (0.013)	0.014 (0.016)	0.637
2x2 (2nd)	Yes	No	17.4 (51.9)	—	0.007 (0.011)	0.011 (0.013)	0.676
2x2	Yes	Yes	5.8 (16.3)	0.874 (0.285)	0.008 (0.012)	0.013 (0.015)	0.626
2x2 (2nd)	Yes	Yes	7.520 (24.2)	—	0.008 (0.012)	0.012 (0.014)	0.680
2x2	No	Yes	5.480 (5.577)	0.882 (0.334)	0.006 (0.007)	0.011 (0.011)	0.533
2x2 (2nd)	No	Yes	2.495 (2.271)	—	0.011 (0.010)	0.017 (0.017)	0.666

Comparison of empirical regret (computed via ”normal” gradient-based approximation a la RegretNet) to our certified regret, under hard IR and regularized IR training.

Certifying Auction Networks: Discussion

RegretNet's gradient-based approximate computation **underestimates** regret → will not reach global optimality

We can produce bounds on the **maximum** regret a player could suffer under a particular valuation profile

- Nice to maintain provable properties with deep learned mechanisms!

Scalability remains an issue (but our ILP solution method is relatively naïve)

Local bounds only – but enough local bounds do guarantee global properties "with high probability" – future work is **global** bounds on regret

Learning Revenue-Maximizing Auctions With Differentiable Matching

RegretNet: Feasible Allocation via Softmax

Feasibility of final allocation matrix is enforced via a mixture of softmaxes and mins (e.g., to prevent over-allocation) at output layer

- Structure is dependent on the demand type of bidders

Example (unit-demand), where bidders want at most one item:

- Softmax over rows (allocate at most 1 mass of any single item)
- Softmax over cols (allocate at most 1 "item mass" to a bidder)
- Min over both enforces the unit-demand constraint

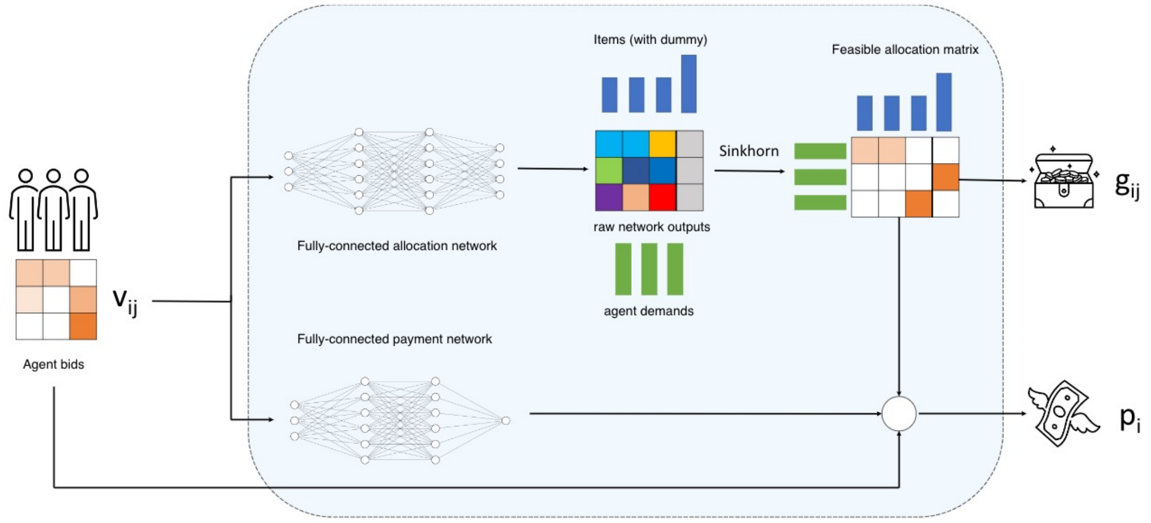
Combinatorial setting is similar, but needs separate output head for every possible bundle of items (= very large!)

Feasible Allocation via Matching LP?

Feasible allocation of items to bidders is just a **bipartite matching**

Formulate as a (discrete) optimal transport problem:

- "Move a set of masses to another place while minimizing cost"
- Discrete version is a minimum cost bipartite matching!



Items (with dummy)
Feasible allocation matrix
Sinkhorn
g_{ij}
raw network outputs
Fully-connected allocation network
V_{ij}
agent demands
Fully-connected payment network
Agent bids
p_i

1. Agents bid on items
2. Inputs to feedforward network(s)
3. Outputs become cost matrix for OT
4. OT gives (approximate) permutation matrix, yielding allocation and charging agents payment based on value of items

Training

Lots of ways to solve discrete OT; we use the Sinkhorn algorithm:

- Iterative, GPU-parallelizable solution method (for entropy regularized version OT problems)

$$\min_{\boldsymbol{P}} \sum_{i=1}^{N} \sum_{j=1}^{M} \boldsymbol{P}_{ij} \boldsymbol{C}_{ij} + \epsilon \sum_{i=1}^{N} \sum_{j=1}^{M} \boldsymbol{P}_{ij} \log \boldsymbol{P}_{ij}$$

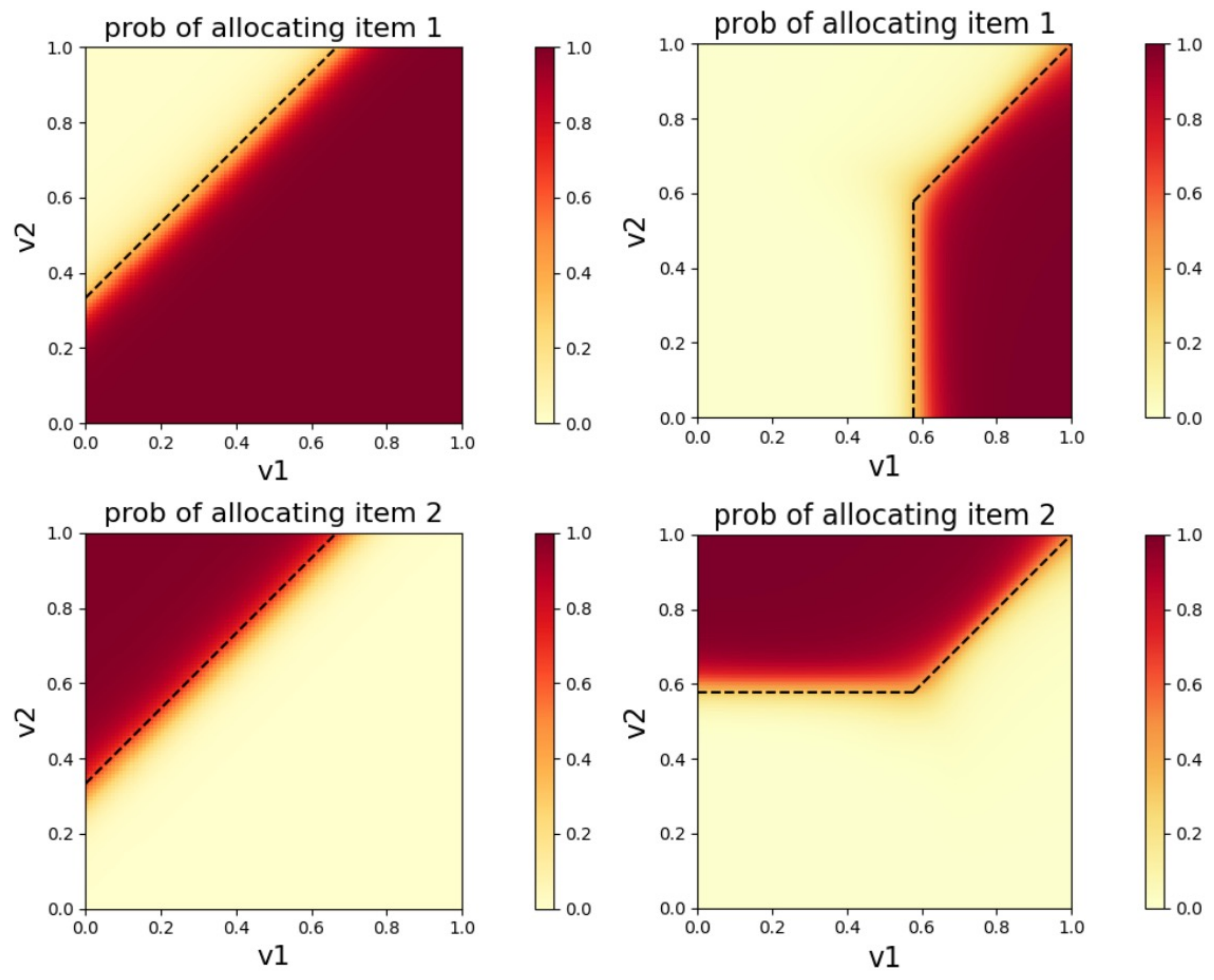
Forward pass: Sinkhorn algorithm

Backward pass: use autodiff to backprop through several iterations of numerical operations of that iterative algorithm!

Sample Results

2 items, 1 bidder, valuations drawn independently from U[0,1] for two cases:

- Unit demand (right), which can be represented by RegretNet
- Exactly-one demand (left), which cannot! Optimal mechanism due to Kash & Frongillo [2016].



(a) Exactly-One (b) Unit-Demand

Conclusions & Final Thoughts

Differentiable economics is a new view on a classic set of problems

- Scalability, training, generalization bounds, etc

Can be viewed as a tool to push theory forward!

- Can use this to "guess" analytic best mechanisms

Everything is differentiable $\rightarrow$ could wrap this into an RL setting …

- (Would want to get a better grip on single-shot scalability first.)

Thanks!

M.C. &

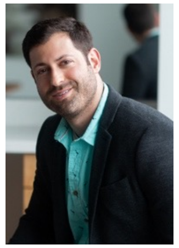
Uro Lyi

Kevin Kuo

John Dickerson

Ping Chiang

Samuel Dooley


Tom Goldstein

Liz Horishny

Anthony Ostuni

More information:

jpdickerson.com

Code:

/imkevinkuo/proportionnet-pytorch

/urolyi1/MechanismDesign

Joint work with:

NeurIPS-20: Certifying Strategyproof Auction Networks
arXiv:2010.06398: ProportionNet: Balancing Fairness and Revenue for Auction Design with Deep Learning
arxiv:TBD: Learning Revenue-Maximizing Auctions With Differentiable Matching

Funding & support: