Deep Learning in Mechanism Design

Shaily Mishra , Sujit Gujar

Abstract

This report contains summary of papers that use Machine Learning or Deep Learning in Mechanism Design. Broadly we look at mechanism design with money (Iterative Combinatorial Auctions) and without money (Multi-Facility Location) , and fairness.

1 Introduction

1.1 Mechanism Design with money

1.1.1 Combinatorial Auction

Combinatorial Auction is an auction in which bidders can bid over bundle (combination) of items rather than just biding individually. Based on the bidders's preferences, the goal of the auction is to maximize economic efficiency.
For *e.g.*, there are 3 bidders - {1,2,3} and 2 items - $\{a, b\}$. bidder 1 bids 10 on $\{a\}$, bidder 2 bids 19 on $\{a, b\}$ and bidder 3 bids 8 on $\{b\}$ then the allocation a^* that maximize social welfare is $(\phi, \{a, b\}, \phi)$ and social welfare value is 19. If payments are allocated using VCG Mechanism, bidder 2's payment is 18, and hence revenue is 18.
These items may be complements or substitutes. If two items {a,b} are complements, they have superadditive utility $v(a, b) \geq v(a) + v(b)$. Similarly, If they are substitutes, they have subadditive utility $v(a, b) \leq v(a) + v(b)$. Combinatorial Auctions lead to economic efficiency, rather than having individual auctions for items.
In this auction, the problem of finding an efficient allocation involves deciding a Bidding Language and solving Winner Determination Problem (WDP). For now, we will assume a simple bidding language in which each possible allocation is attach to a monetary value.

There are m items and n bidders. We use the notation $[m] = \{1, 2, .., m\}$ to denote the set of items.Similarly $[n] = \{1, 2, ..., n\}$ denotes set of bidders. A Bundle is a combination of items. Set of bundles $\chi = \{0, 1\}^m$ Each bidder i has valuation function i.e. $v_i : \chi \mapsto R_{\geq 0}$. Social Welfare is $v(a) = \sum_{i=1}^{n} v_i(a_i)$ where$a = (a_1, a_2, \dots, a_n)$, a_i is the allocation given to bidder i . By WDP we mean, that given a set of bids, find an optimal allocation a^* that maximizes social welfare where a_i is the bundle of items i^{th} bidder received.

$$\begin{aligned}
\max_{a} \quad & \sum_{i=1}^{n} v_i(a_i) \\
\text{subject to} \quad & \sum_{i=1}^{n} a_{ij} \leq 1, \forall j \in [m] \\
& a_{ij} \in 0, 1, \forall j \in [m], \forall i \in [n]
\end{aligned}$$

WDP is equivalent to Set Packing Problem(SPP) and can be formulated as Integer Programming Problem, which is NP-Complete meaning to find optimal solution, the algorithm can run exponential long in worst case.

Most famous Combinatorial Auction is Generalized Vickrey Auction (GVA) [Cramton *et al.*, 2006] We assume (i) Private Value Model - Bidders know their valuations and not others, and their valuation is not dependent on others (ii) Bidders have Quasilinear Utilities, i.e. bidder i's utility for bundle B = valuation of bundle B minus payment, i.e. $u_i(B) = v_i(B) - p$. Each bidder i is supposed to report its full valuation function $v_i(\cdot)$ The algorithm will find allocation a^* that maximizes social welfare including all the bidders . Further the algorithm will find allocations that maximizes social welfare without bidder i i.e. $\forall i$, finding a^*_{-i} Agent i will recieve bundle a^*_i and his payment will be $\sum_{j \neq i} v_i(a^*) - \sum_{j \neq i} v_i(a^*_{-i})$. The GVA runs $n + 1$ times algorithm to find optimal allocation. In GVA, truthfully bidding is dominant strategy and also guarantees efficiency. But because of several issues, it is not practical to use in real world scenarios.
Issues with Vickery Auctions-

- Asking full valuations from bidders - It's costly and highly complex for bidders to calculate their valuations in all the bids (2^m)
- Overall less Revenue
- Not Collusion Proof
- All the bidders true valuations is revealed to everyone
- ...and many more

1.1.2 Iterative Combinatorial Auctions

One of the improvisation of Combinatorial Auctions is Iterative Combinatorial Auctions which addresses the problem of costly preference elicitation (hard valuation problem

- the problem of bidders evaluating their full valuation function). The idea is to iteratively communicate with bidders and querying them accordingly to reach optimal allocation. So the bidders need to evaluate only what the auctioneer will ask them and we can find efficient allocation without knowing bidders full valuation. There are different types of queries - value queries, marginal value queries, indirect-utility queries, and demand queries. [Cramton *et al.*, 2006]
Designing ICA involves designing features like timing issues (continuous/discrete interaction), information feedback(giving price feedback, current provisional allocation,etc. Tradeoff is between how much information to reveal that helps bidder to bid converging to optimal allocation and bidders don't mislead the auction), bidding rules, termination conditions(fixed deadlines/rolling closure), bidding languages, proxy agents (a bidder gives his valuation to a proxy agents and the agent will bid in the auction), etc.

There can be two approaches - Price-Based Iterative Combinatorial Auctions and Non Price-Based Iterative Combinatorial Auctions. In price-based, auctioneer gives option to bidders to ask prices and provides the current provisional allocation and then bidders submit their new bids, the algorithm then calculate the new allocation, prices are updated accordingly and checks for termination conditions. An example is Ascending Price Auctions. Non Price-Based Auctions can be further categorized into decentralized approaches, proxy auctions and direct-elicitation approaches. In decentralized approach, the responsibility to bid and solving WDP is on bidders. An example is Adaptive User Selective Mechanism(AUSM). In proxy auctions, interaction between bidders and auctioneer happens via proxy agents. Agent will decide what to bid, what to query. Bidders will respond to query of agents. In direct elicitation approach, bidders are queried on their valuations. Example of such query can be "Is bundle B_1 preferred over bundle B_2?", "What is valuation of bundle B?". There can be two approaches to elicitation - price based and allocation based. In price based elicitation, we query bidders until the value information is sufficient to verify a set of UCE prices (Universal Competitive Equilibrium Prices)and a supporting allocation for the main economy. In allocation based elicitation, we query bidders until the value information provides a certificate for the efficient allocation and Vickrey payments. [Cramton *et al.*, 2006]

1.2 Mechanism Design without money

1.2.1 Multi Facility Location

In a single facility location problem, given a set of agents $\mathcal{A}$ and space of locations in R^d, distance between an agent i and a location j is given as d_{ij}, the goal is locating the facility such that it reduces social cost. The problem falls under class of mechanism design without money. In a special scenario,where agents have single peaked preferences, and 1-D location, the current literature has interesting characterisation i.e. if the facility is located at the median of all the peaks, it gives minimum social cost, and it's strategy-proof. If we consider space of location to be $L = [0, 1]$, then for agent i, there is a single point $x \in L$, such that as we move away from x, preference (utility) decreases, then agent i has single peaked prefernce with x being the peak.

Moulin provided a characterization of unanimous strategy-proof mechanisms for a single facility, [Golowich *et al.*, 2018]

Theorem 1.1 *(Moulin Generalized Theorem) A unanimous mechanism $f : U \longrightarrow \Omega$ is strategy-proof if and only if it is a generalized median rule, i.e. for each $S \subseteq \{1, \ldots, n\}$, there exists some $a_S \in \Omega$ s.t. for all $(u_1, \ldots, u_n) \in U$,*

$$f(u) = \min_{S \subseteq \{1,\ldots,n\}} \max\left\{\max_{i \in S}\{\tau(u_i)\}, a_S\right\}$$

Extending on the following theorem, A rule f is a generalized median voter scheme (g.m.v.s) if there exist 2^n points in $[0, 1]$, $\{\alpha_S\}_{S \subseteq N}$, such that [Nisan *et al.*, 2007]

1. $S \subseteq T \subseteq N$ implies $\alpha_S \geq \alpha_T$
2. $\alpha_\phi = 1$, $\alpha_N = 0$
3. For all τ_i , $f(tau) = \min_{S \subset N} \max \alpha_S, \tau_i : i \in S$

For example, let there be 3 agents $\{1, 2, 3\}$, with peaks $\{0.2, 0.7, 0.9\}$ respectively. As shown in Table 1, any set of a_S which don't violate constraints, will lead to strategy proof mechanism.

So Moulin Generalized Median rules gives us the class of strategy proof mechanism for single facility problem, but not all of them will lead to optimal solution. For the problem involving locating multiple facilities, the current literature doesn't have complete characterization of strategy-proof mechanisms.

1.3 Fairness in Classification

Algorithmic Decision making might lack fairness in its results. For e.g. Pretrial Risk Assessment , Mortgage Apporvals, NYPD Stop-question-and-frisk program, content recommendations, etc. We need systems not to discriminate based on sensitive attributes (race, gender, etc.)

There are various notion of fairness.

- Disparate Treatment : A decision making process suffers from Disparate Treatment if the decisions are based on subjects' sensitive attributes. If we keep the sensitive attributes, while training our model, we introduce disparate treatment in our model. [Zafar *et al.*, 2017]
- Disparate Impact : A decision making process suffers from Disparate Impact if the outcomes disproportionately hurt (or benefit) people with certain sensitive attributes. [Zafar *et al.*, 2017] Even if we remove sensitive attributes from our decision making process, still there might be correlation between sensitive attributes and class labels as our data might contain past discrimination, which will cause disparate impact. E.g. If in a dataset, gender is correlated getting a post, then % males getting post and % females getting post will be different.
- Disparate Mistreatment : A decision making process to be suffering from disparate mistreatment with respect to a given sensitive attribute (e.g. race) if the misclassification rates differ for groups of people having different values of that sensitive attribute. [Zafar *et al.*, 2017]

$S \subseteq N$ {3,2,1}	max τ_i	a_S	max	a_S	max	a_S	max	a_S	max	a_S	max	a_S	max
000	0	1	1	1	1	1	1	1	1	1	1	1	1
001	0.2	0.9	0.9	0.99	0.99	0.15	0.2	0.6	0.6	0.89	0.89	0.3	0.3
010	0.7	0.8	0.8	0.98	0.98	0.14	0.7	0.55	0.7	0.88	0.88	0.29	0.7
011	0.7	0.7	0.7	0.97	0.97	0.13	0.7	0.5	0.7	0.87	0.87	0.27	0.7
100	0.9	0.6	0.9	0.96	0.96	0.12	0.9	0.45	0.9	0.86	0.9	0.26	0.9
101	0.9	0.5	0.9	0.95	0.95	0.11	0.9	0.4	0.9	0.85	0.9	0.25	0.9
110	0.9	0.4	0.9	0.94	0.94	0.10	0.9	0.35	0.9	0.84	0.9	0.24	0.9
111	0.9	0	0.9	0	0.9	0	0.9	0	0.9	0	0.9	0	0.9
f value			min = 0.7		0.9		0.2		0.6		0.87		0.3

Table 1: Example of a_S

2 Combinatorial Auctions via Machine Learning-based Preference Elicitation

2.1 Problem Addressed

In a setting of Non Price-Based Allocation-Based Elicitation Approach ICA, using value queries, the main challenge is to decide what preference to elicit so that we reach an optimal allocation?

2.2 Solution

This paper presents an ML-based elicitation algorithm which identifies which value to query and then design a mechanism called PVM where payments are determined so that the bidders incentives are aligned with allocative efficiency [Brero *et al.*, 2018]

2.3 Working of ML-based Elicitation Algorithm

- Defining terms, bundle value pair of any bidder i , is b_i : $(x, \hat{v_i}(x))$, where x is the bundle i.e. $x \in \chi$ and $\hat{v_i}(x)$ is the reported valuation by bidder i (maybe truthful or not) of that bundle. Set of bundle value pairs of a bidder i is written as B^i
- Initally we have B_0 where $(B_0^1, B_0^2, ..., B_0^n)$ is initial bundle value pairs reported by all bidders, at t=0
- Next the ML Algorithm $\mathcal{A}$, using this B_0, gives a inferred social welfare function $\tilde{v}^0$.
- We then solve the IP problem to find allocation a_0 that maximizes this inferred social welfare
- Now we check for each bidder i, if the allocation (a_0^i) that is allocated to them is already queried or not.
- If it is not queried then we query them, and create a new set of bundle value pairs (B_i^1) which is $B_i^0 \cup (a_i^0, \hat{v_i}(a_i^0))$
- this cycle goes on until at some round t, we have that all elements of B^t is already present in B^{t-1}, meaning in that round, we didn't need to query bidders about any new valuations, the valuations we had lead to the optimal allocation.

Rough example of this algorithm : There are 2 items : {a,b} and 2 bidders {1,2}. Four combination of items will be (0,0), (0,1), (1,0), (1,1) Following is the valuations for both the bidders:

	(0,0)	(0,1)	(1,0)	(1,1)
Bidder 1	0	2	0	2
Bidder 2	0	0.5	0.5	2

Now, say the inital bundle value reported , B_1^0 is {{(0,1):2}} and B_2^0 is {{(1,0):0.5}}. Assume we have linear valuation functions, (linear regression algorithm)

$$v_1 = w_1 x_{11} + w_2 x_{12}$$
$$v_2 = w_3 x_{21} + w_4 x_{22}$$
$$v = v_1 + v_2$$
$$\text{where} \quad x_{ij} = 1 \text{ , if item j is allocated to bidder } i$$
$$w_i \text{ are the weights}$$

Based on the B^0 the algorithm might inferr that $\tilde{v_1}^0 = 2x_{11} + 2x_{12}$ and $\tilde{v_2}^0 = 0.5x_{21} + 0.5x_{22}$ and thus the infered social welfare function is $\tilde{v}^0 = 2x_{11}+2x_{12}+0.5x_{21}+0.5x_{22}$. On the next step, we find such an allocation that maximizes this infered function and thus we obtain $a^0 = ((1,1), \phi)$. Now, we check each a_i^0, if it is not queried then we query them and make a new bundle-value pair containing the existing bundle-value pairs. Since a_1^0 is not in B_1^0, we query it, and B_1^1 is {{(0,1):2},{(1,1):2}} and B_2^1 is {{(1,0):0.5}} Again the algorithm will learn from the B^1 and the new infered social welfare function is $\tilde{v}^1 = 2x_{11} + 0.5x_{21} + 0.5x_{22}$. and the allocation that maximizes it is $a^1 = ((1,0),(0,1))$. Since all the a_i^1 are already queried, the algorithm will terminate here.

SVRs (Support Vector Regression) is used to infer each $\tilde{v}_i$. The paper uses linear and quadratic kernels.

Designing Mechanism $\mathcal{M}$: 1) procedure to determine an allocation ($a^{\mathcal{M}}$) 2) payment rules ($p^{\mathcal{M}}$) Explanation of PVM : In Step1, we run the Preference Elicitation algorithm n+1 times, i.e. 1st time we include all the bidders, and the next n times, we exclude a different bidder each time. Output gives us bundle value pairs obtained by the preference elicitation algorithm n+1 times - $\{B^{-\phi}, B^{-1}, B^{-2}, ..., B^{-n}\}$. Step2 determines the allocation a^{-i} based on B^{-i} that maximizes social welfare using Integer Programming (WDP). (We are not finding allocation that maximizes inferred social welfare function) In Step3, final allocation is chosen from $\{a^{-\phi}, a^{-1}, ..., a^{-n}\}$ that gives the maximum social welfare. In Step4, payments are calculate in a similar way it is done in

Algorithm 1 Algorithm1 : Ml-based Elicitation Algorithm - MLEA

Paramter: Machine Learning Algorithm ($\mathcal{A}$)
B^0 = initial bundle-value pairs of all bidders, $t = 0$
do
    $t \leftarrow t + 1$
    Get Infered Social Welfare function $\tilde{V}^t = \mathcal{A}(B^{t-1})$
    Determine allocation $a^t \in \text{argmax}_{a \in \mathcal{F}} \tilde{V}^t$
    for each bidder i **do**
        if $a_i^t \notin B_i^{t-1}$ **then**
            Query value $\hat{v}_i(a_i^t)$
            $B_i^t = B_i^{t-1} \cup (a_i^t, \hat{v}_i(a_i^t))$
        else
            $B_i^t = B_i^{t-1}$
while $\exists i \in [n] : a_i^t \notin B_i^{t-1}$
Output final set of bundle-value pairs B^T, where $T = t$

VCG Mechanism, (sum of all bidder's valuation on the optimal allocation that was calculated when bidder i is not present) minus (sum of all bidder's valuation except i on the optimal allocation that was calculate including all bidders)

Algorithm 2 Algorithm2 : Pseudo-VCG Mechanism (PVM)

Run Algorithm1 n+1 times to get $(B^{-\phi}, B^{-1}, B^{-2}, ..., B^{-n})$
Determine Allocations : $(a^{-\phi}, a^{-1}, ..., a^{-n})$
Pick $a^{pvm} \in (a^{-\phi}, a^{-1}, ..., a^{-n})$ that maximal $\tilde{v}$
Charge each bidder i

$$p_{pvm}^i = \sum_{j \neq i} (\hat{v}_j(a_j^{(-i)})) - \sum_{j \neq i} (\hat{v}_j(a_j^{(pvm)}))$$

2.4 Properties of this mechanism

- Under PVM, bidders incentives are aligned with allocative efficiency
- PVM satifies Individually Rationality
- It is not strategy proof as bidders can indirectly affect what values the preference elicitation algorithm might ask other bidders. The paper gives an example of a situtation when its not strategyproof. and informally states that better the ML algorithm inferres valuation function of bidders, smaller the incentive to bidders to manipulate.
- The paper doesn't give any efficiency guarantees, but experimental evaluation achieves high average efficiency
- PVM doesn't guarantee no deficit property. Experimentally they have never observed any deficit. (Though one can keep lower bounds on payments, and ensure no deficit)
- The paper doesn't give any bounds of time.

2.5 Variation of this Mechanism

- PVM with paritions(PVMp): As n grows, running n+1 times preference elicitation algorithm is highly expensive. So insteading of just excluding one bidder, we can make k groups of bidders and exclude different group each time, hence running k+1 times the preference elicitation algorithm. PVMp is also incentive aligned. Bad grouping might lead to low revenue and/or deficit violations.
- Elicitation with upper and lower bounds : (Modification of algorithm1) Instead of giving exact valuation of any bundle, giving upper bound and lower bound. then the algorithm will consider the reported value as average of upper and lower bound, and measure the interpolation error : (upper bound - lower bound)/2 to inferred the social welfare function. We cannot use the current PVM, as now our bundle value pairs have bounds, hence don't have enough information to find optimal allocation.

2.6 Experiment Results

- Dataset : Spectrum Auction Test Suite (SATS) - has 2^m valuations, hence we can find optimal allocation using IP. Three domain of this dataset are taken : The Global Synergy Value Model (GSVM) - (18 items,7 bidders), The Local Synergy Value Model (LSVM) - (18 items,6 bidders), and The Multi-Region Value Model (MRVM) - (98 items, 10 bidders). In GSVM, the value of bundle depends on total number of items in it, hence it has most simplest structure. Valuations structure is more complex in LSVM than in GSVM. There is a time limit set on solving IP - 1 hour, if it goes beyond that, best solution found so far is considered.
- Performance of ML-based Algorithm : Four scenario is consider - No elicitation, random query, ML-based, and full elicitation. No query - without any query, assign each item uniformly randomly to bidders.(Lower bound) Random query - randomly a set of bundle is decided to be queried. and Allocation is found using WDP from those bundle. and Full Elicitation is having full valuation of bidders, and then finding optimal allocation (Upper bound) Elicitation efficieny is compared across the four category.
- Performance of PVM : PVM achieves more than 94% efficiency in all three domains.

3 Deep Learning–powered Iterative Combinatorial Auctions

3.1 Problem Addressed

In the ML-based ICA, we were only using linear and quadratic kernels. Valuations function might have complex structure. As ML-based ICA always timed out with Gaussian or complex kernels, it won't be able to scale in larger domains.

3.2 Solution

Using DNNs instead of SVRs for Preference Elicitation Algorithm, rest the whole algorithm remains same. Also the

Domain	Elicitation Method	# of Queries/Bidder	Elicitation Efficiency
44emGSVM	No Elicitation	0	22:0% (0:9%)
	Random Query	50	68:8% (0:7%)
	ML-based	≤ 50	98:5% (0:1%)
	Full Elicitation	2^18	100:0% (0:0%)
44emLSVM	No Elicitation	0	20:3% (0:6%)
	Random Query	50	62:5% (0:8%)
	ML-based	≤ 50	93:5% (0:4%)
	Full Elicitation	2^18	100:0% (0:0%)
44emMRVM	No Elicitation	0	32:7% (0:6%)
	Random Query	100	51:5% (0:4%)
	ML-based	≤ 100	93:3% (0:1%)
	Full Elicitation	2^18	100:0% (0:0%)

Table 2: ML-ICA Results of Elicitation Efficiency (in brackets are standard error)

PVM mechanism remains same. [Weissteiner and Seuken, 2019] The paper first shows how DNN based WDP can be formulated into MIP and then compares prediction performance and economic efficiency between ML-based ICA and DL-based ICA.

3.3 Working

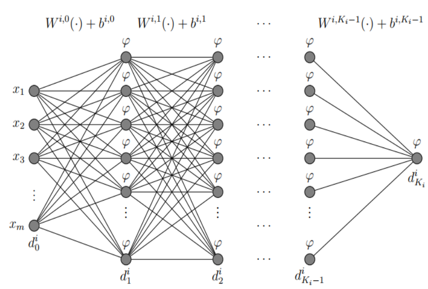

Figure 1: Representation of DNN $\mathcal{N}_i$

$\tilde{V} = \sum_{i \in N} \mathcal{N}_i$, where $\mathcal{N}_i$ is the DNN that learns valuation function of bidder i. Each layer uses relu as activation function $\varphi : R : R+, \varphi(x) = max(0, x)$. $\mathcal{N}_i$ maps $x \in \{0,1\}^m$ i.e. combination of items to its valuations for bidder i.

Advantage of this is that we always obtain linear MIP. while in SVR, if we have quadratic kernels then we are solving QIP, and so on.

WDP can be formulated as

$$\begin{aligned} \max_a \quad & \sum_{i=1}^{n} \mathcal{N}_i(W_i, b_i)(a_i) \\ \text{subject to} \quad & \sum_{i=1}^{n} a_{ij} \leq 1, \forall j \in [m] \\ & a_{ij} \in 0, 1, \forall j \in [m], \forall i \in [n] \end{aligned}$$

Now this is a non linear and non convex optimization problem. The paper further formulates this DNN's WDP into MIP formulation.

3.4 Experiment Results

- All the results are average over 100 auction instances for GSVM and LSVM, and 50 auction instances for MRVM
- Testing DNNs prediction vs SVR prediction for dataset - GSVM and LSVM. As quadratic kernel fits GSVM structure efficiently, SVR's does slightly better, and this can be looked as worst case for comparing DNNs and SVRs. For LSVM, DNNs outperforms SVRs.
- For efficiency, for each dataset, the paper tried out DNNs with different hidden layers (2/3) and neurons in each layers (either multiple of 10 or 16). and picked the winner model which gave highest efficiency. Then they compared it with SVRs, for both dataset - GSVM and LSVM, and DNNs were similar in terms of efficiency for GSVM model (as SVRs gives the optimal solution for this one), and for For LSVM, DNNs clearly outperforms.
- Over the 100 auction instances, on the selected winner model, 29 instances gave 100% efficiency , but 2 gave less than 90 % efficiency.
- DNNs gave low revenue than SVRs, but that can be controlled by having lower bounds on payments
- On MRVM, DNNs performs better than SVRs, however they ask more queries than SVRs. So its unclear if its because of DNNs or high no. of queries
- The paper also gives experimental runtime of algorithm, and results show that it is practically feasible. Runtime of MIP also depends on big-M variable L. Since ML-based ICA Paper didn't give any data on time it cannot be compared. But in conversation with authors, SVRs regularly timeout.

Domain	MIP Runtime	Iteration Runtime	Auction Runtime
GSVM	15.90 sec	30.51 sec	44 min
LSVM	39.75 sec	51.69 sec	65 min
MRVM	3.67 sec	26.75 sec	457 min

4 Deep Learning for Multi-Facility Location Mechanism Design

4.1 Problem Addressed

We already have class of strategy-proof single facility mechanisms. But for multi facility, we don't have much literature. So the goal of the paper is to design strategy-proof, multi-facility mechanisms that minimize expected social cost.

4.2 Solution

Design a MoulinNet (NN) to already existing characterization results (Generalized Moulin Median Rule) for single facility as well as multi facility. Design RegretNet (NN) for multi facility which is almost strategy proof and is as good as available theoretically results

4.3 Working of the Algorithm

Setting of paper : N agents $\{1, 2, \ldots, N\}$, set of location Ω = [0,1], K facilities. Each agent has single peaked preference over Ω. For each agent i, utility will be the based on the facility that was placed closet to its preferred location. i.e. $u_i(o) = \max_{k \in K} u_i(o_k)$. We assume $u(x) = -|x - a|$ where $a = \tau(u)$ is the peak

4.3.1 MoulinNet

Earlier, in introduction we have mentioned about Single Facility generalized Median Rules. This paper extends that idea to multifacility scenario.

Let MGM denote the class of multi-facility generalized median rules given by $f = (f_1, \ldots, f_K)$, where each f_k is a 1-facility generalized median rule for parameters $a_S^k \in \Omega, S \subseteq N$ As a_S are montone, We use standard neural network to learn montone functions, it maps an n-dimensional binary representation of a set S i.e. $v(S)$ to a real value $a_S = h(v(S))$

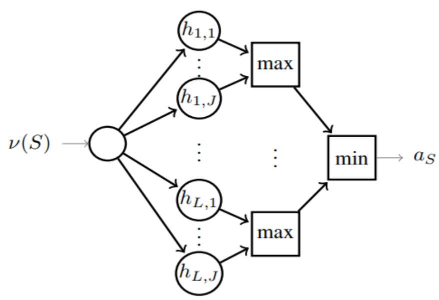

Figure 2: NN to learn a_S

The following single-facility mechanism is strategy-proof

$$f^{w,b}(u) = \underset{S \subseteq N}{min} \left\{ \max_{i \in S} \{\tau(u_i), h^{w,b}(v(S))\} \right\}$$

This method still requires to find 2^n a_S and this formulation can be further reduced to,

$$f^{w,b}(u) = \underset{S \subseteq N}{min} \left\{ \max_{i \in S} \{\tau(u_i), h^{w,b}(v(S))\} \right\}$$

$$\begin{aligned} f^{w,b}(u) &= \min_{1 \le i \le n} \left[\max\{\tau(u_{\pi(i)}), h^{w,b}(v(S_{\pi,i}))\} \right] \\ \text{where} &\quad \pi \text{ is sorted ordered of peaks} \\ &\quad S_{\pi,i} = \{\pi(1), \ldots, \pi(i)\} \end{aligned}$$

For example, if $\tau_1 < \tau_2 < \tau_3$, then $\pi = \{1, 2, 3\}$, $S_{\pi,1} = \{1\}$, $S_{\pi,2} = \{1, 2\}$, and $S_{\pi,3} = \{1, 2, 3\}$.

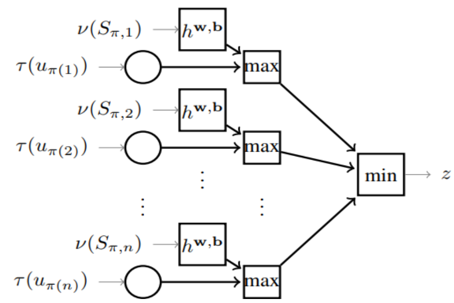

Figure 3: MoulinNet for single facility

To train the network, we optimize the parameters, by using social cost as loss function.

4.3.2 RegretNet

We use a NN, fully connected, L hidden layers, takes inputs as agents peaks and gives locations of K facilities as output Goal here is to minimize $\mathcal{L}(w)$ expected social cost subject to expected ex post regret being zero for all agents such that it is almost strategy-proof. Expected Ex-post regret from mechanism f to agent i,

$$rgt_i(f) = \underset{U \sim D}{E} \left[\max_{u_i' \in U_i} u_i(f(u_i', u_{-i})) - u_i(f(u_i', u_{-i})) \right]$$

So the objective function becomes,

$$\min_{w \in R^d} \mathcal{L}(w) \quad \text{s.t. } rgt_i(f^w) = 0 \ \ \forall i \in N$$

The paper uses stochastic gradient descent (SGD) to optimize the network parameters w to minimize the Lagrangian formulation of following objective function.

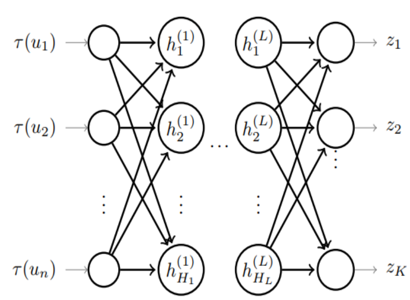

Figure 4: RegretNet Architecture

S N {3,2,1}	max ti	as	max	as	max	as	max	as	max	as	max	as	max
000	0	1		1		1		1		1		1	
001	0.2	0.9	0.9	0.99	0.99	0.15	0.2	0.6	0.6	0.89	0.89	0.3	0.3
010	0.7	0.8		0.98		0.14		0.55		0.88		0.29	
011	0.7	0.7	0.7	0.97	0.97	0.13	0.7	0.5	0.7	0.87	0.87	0.27	0.7
100	0.9	0.6		0.96		0.12		0.45		0.86		0.26	
101	0.9	0.5		0.95		0.11		0.4		0.85		0.25	
110	0.9	0.4		0.94		0.10		0.35		0.84		0.24	
111	0.9	0	0.9	0	0.9	0	0.9	0	0.9	0	0.9	0	0.9
f value			min = 0.7		min = 0.9		0.2		0.6		0.87		0.3

Table 3: Optimized way of finding generalized median rules f

K	Perc.	Dict.	Cons.	Moulin Net	RegretNet Sc	max Regret	Non SP
1	0.200	0.267	0.253	0.201	0.201	0.0003	0.200
2	0.0833	0.126	0.126	0.0837	0.0833	0.0003	0.0708
3	0.0335	0.0609	0.0834	0.0353	0.0376	0.0009	0.0278
4	0.0171	0.0236	0.0635	0.0188	0.0177	0.0024	0.0083

Table 4: Experimental Results of MoulinNet and RegretNet

4.4 Experimental Results

- Compared the results with the existing mechanism - best percentile rule (A percentile rule locates each facility at a fixed percentile of the reported peaks), best dictatorial rule (A dictatorial rule locates each facility at the peak of a fixed agent) and best constant rule (A constant rule locates each facility at a fixed point)
- Evaluated over Unweighted social cost, Weighted social cost and Non Independent Valuations For Unweighted social cost, both kinds of networks yield similar performance as the best percentile rule. For weighted social cost, RegretNet-nm and MoulinNet yield significantly smaller social cost than the baseline mechanisms. For Non Independent Valuations, RegretNet-nm outperform all the benchmarks

5 Fairness Constraints: Mechanisms for Fair Classification

5.1 Problem Addressed

Among many fairness notion, this paper focuses on making a classifier free of Disparate Treatment and Disparate Impact.

5.2 Solution

To design a fair classifier covering two scenarios: 1) Maximizing accuracy with given fairness constraints. 2) Maximizing fairness given accuracy constraints (business necessity). Also, to generalize over any convex classifiers (ML classifiers), dataset having multiple sensitive attributes, and each sensitive attributes might have multiple values.

5.3 Working

We will not consider sensitive attributes while training our classifiers, hence we won't have Disparate Treatment. We consider the scenario where we know that the training data already has bias against certain attributes, in that case balancing the results over those attributes (apply p% rule) will migitate Disparate Impact. But directly incorporating p % rule in convex-margin based classsifier will result into a non convex optimization problem.

p% rule : If the ratio between the percentage of users with a particular sensitive attribute value having $d_\theta(x) \geq 0$ and the percentage of users without that value having $d_\theta(x) \geq 0$ is no less than (p:100) i.e. 80 % rule means the ratio is at least 80:100

Formulating measure of decision boundary (un)fairness as decision boundary co-variance. i.e. measuring the unfairness by finding co-variance between user's sensitive attributes z and signed distance from user's feature vectors (x) to decision boundary $d_\theta(x)$

$$Cov(z, d_\theta(x)) \quad = \frac{1}{N}\sum_{i=1}^{N}(z_i - \bar{z})d_\theta(x_i)$$

If a decision boundary satisfies the 100 % rule then the co-variance will be approximately zero for large data set

5.3.1 Maximizing Accuracy Under Fairness Constraints

Design a classifier that maximizes accuracy to fairness constraints (i.e. a specific p % rule)

$$
\begin{aligned}
&minimize \;\; L(\theta) \\
&subject\ to \; \frac{1}{N}\sum_{i=1}^{N}(z-\bar{z})d_\theta(x_i) \leq c \\
&\qquad\qquad \frac{1}{N}\sum_{i=1}^{N}(z-\bar{z})d_\theta(x_i) \geq -c \\
&\qquad where \;\; \text{x is the feature vector without sensitive attributes} \\
&\qquad\qquad\quad \text{z is the sensitive attributes of feature vector} \\
&\qquad\qquad\quad \text{c is co-variance threshold}
\end{aligned}
$$

5.3.2 Maximizing Fairness Under Accuracy Constraints

Design a classifiers that maximizes fairness to accuracy constraints i.e. without any fairness constraints, we find the loss of our classifier, and that will be our optimal loss. so we minimize the co-variance i.e. unfairness, subject to accuracy

$$
\begin{aligned}
&minimize \;\; \left|\frac{1}{N}\sum_{i=1}^{N}(z_i-\bar{z})\right| \\
&subject\ to\ L_i(\theta) \leq (1+\gamma_i)L_i(\theta^*) \;\; \forall i \in 1,2,...,N \\
&\qquad where \;\; L_i(\theta^*) \text{ is individual optimal loss of ith data} \\
&\qquad\qquad\quad \gamma_i \geq 0 \text{ is allowed additional loss} \\
&\qquad\qquad\quad \gamma_i = 0 \text{ means that loss should } \leq \text{ optimal loss}
\end{aligned}
$$

5.4 Experimental Results

Data Sets - Synthetic data by adding attribute that is correlated to class labels, and Real data - Adult income data set and Bank Marketing data set. In Synthetic data, there is single sensitive attribute which has binary values, and it is binary Classification In Adult income data, classification of data is based on whether an individual has income above 50K USD or not. It contains two sensitive attributes - gender (having binary values) and race (having multiple values). In Bank Marketing data set, classification is based on whether an individual has subscribed or not. It contains one sensitive attribute - age, which here takes binary values to indicate whether age is between 25 to 60 years.

All the above data set are trained over Logistic regression and SVM classifiers The paper compares, classification with binary attributes with similar competing methods and achieves similar results, while for non-binary and multiple sensitive attributes, existing competing methods cannot handle. As our co-variance decrease, p % increases. Empirically we see the trade-off between fairness and accuracy, and chose the parameters accordingly.

References

[Brero *et al.*, 2018] Gianluca Brero, Benjamin Lubin, and Sven Seuken. Combinatorial auctions via machine learning-based preference elicitation. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 128–136. ijcai.org, 2018.

[Cramton *et al.*, 2006] Peter Cramton, Yoav Shoham, and Richard Steinberg. *Combinatorial Auctions*. The MIT Press, 2006.

[Golowich *et al.*, 2018] Noah Golowich, Harikrishna Narasimhan, and David C. Parkes. Deep learning for multi-facility location mechanism design. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 261–267. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[Nisan *et al.*, 2007] Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.

[Weissteiner and Seuken, 2019] Jakob Weissteiner and Sven Seuken. Deep learning-powered iterative combinatorial auctions. *CoRR*, abs/1907.05771, 2019.

[Zafar *et al.*, 2017] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness constraints: Mechanisms for fair classification. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 962–970. PMLR, 2017.

Paper	Appeared in	Problem Statement	Solution Proposed
Combinatorial auctions via machine learning-based preference elicitation	IJCAI 18	In a setting of Non Price-Based Allocation Based Elicitation Approach ICA, using value queries, the main challenge is to decide what preference to elicit so that we reach an optimal allocation?	ML-based elicitation algorithm which identifies which value to query and then design a mechanism called PVM, where payments are determined so that the bidders incentives are aligned with allocative efficiency
Deep Learning–powered Iterative Combinatorial Auctions	AAAI 20	In the ML-based ICA, we were only using linear and quadratic kernels. Valuations function might have complex structures. As ML-based ICA always timed out with Gaussian or complex kernels, it won't be able to scale in larger domains.	Using DNNs instead of SVRs for Preference Elicitation Algorithm, rest the algorithm remains same. Also the PVM mechanism remains same.
Deep Learning for Multi-Facility Location Mechanism Design	IJCAI 18	We already have a class of strategy-proof single facility mechanisms. For multi-facility, we don't have much literature. So the goal of the paper is to design strategy-proof, multi-facility mechanisms that minimize expected social costs.	Design a MoulinNet to already existing characterization results (Generalized Moulin Median Rule). Design RegretNet (NN) for a generalized mechanism is as good as available theoretically results
Fairness Constraints: Mechanisms for Fair Classification	AISTATS 17	Among various fairness notion, this paper focuses on making a classifier free of Disparate Treatment and Disparate Impact	To design a fair classifier covering two scenarios: 1) Maximizing accuracy with given fairness constraints. 2) Maximizing fairness given accuracy constraints To generalize over any convex classifiers. dataset having multiple sensitive attributes, and each sensitive attributes might have multiple values

Table 5: List of Papers summarized