ML and DL in Iterative Combinatorial Auctions

February 2020

1 Introduction to Combinatorial Auctions

- What is 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 preference given by bidders, the goal of the auction maximize economic efficiency (Social welfare and Revenue)
    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, and if payments are allocated using Vickrey Mechanism(i.e. Second Price Auction), then payment made by bidder 2 will be 18. Revenue will be 18.
  - These items may be complements or substitutes. If two items {a,b} are complements,then they have superadditive utility $v(a, b) \geq v(a) + v(b)$. Similarly, If they are substitutes,then 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 a combinatorial 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.
  - Defining a formal model : 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. $v_i : \chi \mapsto \mathbb{R}_{\geq 0}$. Social Welfare is $v(a) = \sum_{i=1}^{n} v_i(a_i)$ where$a = (a_1, a_2, ..., a_n)$, 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 expotenial long in worst case. (TODO:Write IP formaultion here)
- Most famous Combinatorial Auction is Generalized Vickrey Auction (GVA) (TODO:Put a reference here)
  - 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 used in practice.
- Issues with Vickery Auctions-
    * Asking full valuations from bidders - Its 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

2 Introduction to Iterative Combinatorial Auctions

- One of the improvisation of Combinatorial Auctions is Iterative Combinatorial Auctions which addresses the problem 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. (TODO: reference here)
- 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 decentrailzed apporach, 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. (TODO:Add reference here)

3 ML powered ICA

- Title : Combinatorial Auctions via Machine Learning-based Preference Elicitation
- Authors : Gianluca Brero and Benjamin Lubin and Sven Seuken
- Problem : 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?
- 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
- Explaination 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)

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

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 $(p^{\mathcal{M}})$
- Explaination 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 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)
- Properties of this mechanism
  - Under PVM, bidders incentives are aligned with allocative efficiency

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 $T = t$

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

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

- 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.

• Variation of this Mechanisms

- 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.

• Experimental 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

Domain	Elicitation Method	# of Queries/Bidder	Elicitation Efficiency
GSVM	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%)
LSVM	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%)
MRVM	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 1: Results of Elicitation Efficiency (in brackets are standard error)

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.

4 DL powered ICA

- Title : Deep Learning–powered Iterative Combinatorial Auctions
- Authors : Jakob Weissteiner and Sven Seuken
- Problem : In the ML-based ICA, we were only using linear and quadratic kernels. Valuations function might have complex structure. As the ML-based ICA always timed out with guassian or complex kernels, it won't be able to scale in larger domains.
- Solution : Using DNNs instead of SVRs for Preference Elicitation Algorithm, rest the whole algorithm remains same. Also the PVM mechanism remains same. The paper first shows how DNN based WDP can be formulated into MIP (WDP will be then solved efficiently?) and then compares prediction performance and economic efficiency between ML-based ICA and DL-based ICA.
- Working of the algorithm
  - Formulations $\tilde{v} = \sum_{i \in \mathbb{N}} \mathcal{N}_i$, where $\mathcal{N}_i$ is the DNN that learns valuation function of bidder i. Each layer uses relu as activation function $\varphi : \mathbb{N} : \mathbb{N}_=$,
  - 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 $\varphi(x) = max(0, x)$ 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. We formulating this into MIP (MIP FORMULATION HERE)

- Experimental Evaluations
  - 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

References