Optimal Ordering of Sequential Auctions

Sicco Verwer

Yingqian Zhang

Motivating example: Dutch flower auctions

- approximately
- 125,000 transactions
- per day

- Dutch auction:
- open descending auction

- Auctioneers try to sell the flowers for the highest price
  - Ordering of auctions matter!

Examples

- Two agents: A1, A2
- two types of flowers: Tulip, Rose
  - A1: v(T) = v(R) = 5, with budgets 5
  - A2: v(R) = 4, with budgets 4
- Two possible orderings:
  - <T, R>
    - T -> A1, R->A2; with revenue 9
  - <R, T>
    - R -> A1, T unsold; with revenue 5

EXAMPLES

- Two agents: A1, A2
- two types of flowers: Tulip, Rose
  - A1: v(R) = 1, v(T)=1, v(T, R) = 10
  - A2: v(R) = 5
- Two possible orderings:
  - <R, T>
    - R -> A2, T->A1; with revenue 6
  - <T, R>
    - T -> A1, R -> A1; with revenue 10

Causes of revenue change

- Budget constraints
  - In some orders items are unsold because the agents that want them have no money left
- Combinatorial preferences
  - In some orders parts of combinatorial preferences are already sold to other agents
- Strategies
  - Neglected for now….

Design ordering of auctions

- Learning the individual preferences and strategies in combinatorial domains is hard
- Even if utiltiy functions are known, computing an optimal ordering is NP-complete

- Instead, we learn a single model for the entire agent population
- We use established machine learning methods to learn this model

Design ordering of auctions

- Learning to order auctions

- Given historic data of auctions:
  - The ordering of items
  - Prices of the items
  - Who bought the items
- Find a model for the expected revenue (wrt random) of a given order

Setting and assumptions

- Sequential (first-price) auctions
  - One auctioneer; n buyers; multi-items
- The revenue of an ordering is independent of the participants
- The revenue of an item depends on:
  - The items sold before
  - The items to be sold after
  - But not the order in which they were sold

Features

- For detecting combinatorial preferences:
  - Feature 1: The items already auctioned
  - Feature 2: The items still to be auctioned
- For detecting budget constraints:
  - Feature 3: The sum of revenue so far
  - Feature 4: The revenues obtained per item type

- Features satisfy the model setting and assumptions

Finding optimal ordering

- Suppose we have the following ordering:
  - R,R,T,R,T,T,T,R
- With corresponding revenues:
  - 10,8,4,8,6,3,3,14

Finding optimal ordering

Type	Revenue	R before	R after	T before	T after	Sum	Sum R	Sum T
R	10	0	3	0	4	0	0	0
R	8	1	2	0	4	10	10	0
T	4	2	2	0	3	18	18	0
R	8	2	1	1	3	22	18	4
T	6	3	1	1	2	30	26	4
T	3	3	1	2	1	36	26	10
T	3	3	1	3	0	39	26	13
R	14	3	0	4	0	42	26	16

Finding optimal ordering

Type	Revenue	R before	R after	T before	T after	Sum	Sum R	Sum T	Mean R	Mean T
R	10	0	3	0	4	0	0	0	10	4
R	8	1	2	0	4	10	10	0	11	4
T	4	2	2	0	3	18	18	0	11	4
R	8	2	1	1	3	22	18	4	14	4
T	6	3	1	1	2	30	26	4	14	3
T	3	3	1	2	1	36	26	10	14	3
T	3	3	1	3	0	39	26	13	14	NA
R	14	3	0	4	0	42	26	16	NA	NA

Finding optimal ordering

- The current reward is the item revenue
- The future reward is the sum of mean revenues of all remaining items

- Greedy Algorithm:
  - Use the models to select an item type that maximizes the sum of expected current and future rewards
  - Order this item first, and iterate for the next position

- Output is an ordering with high expected reward

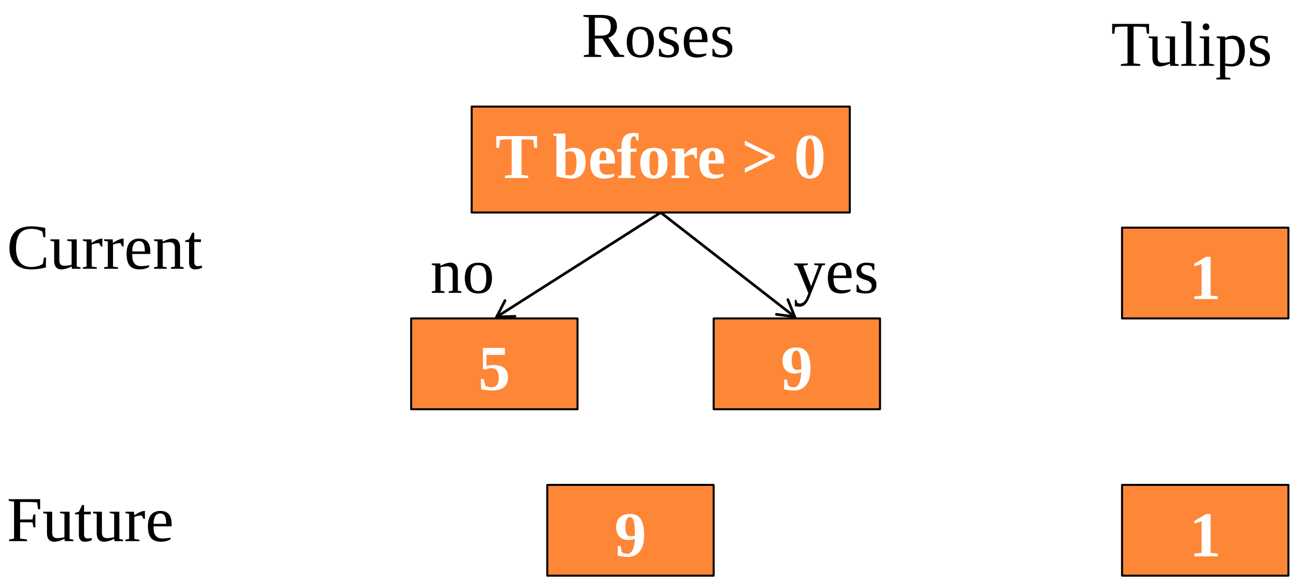
Example

- Two orderings as input:
  - <(R,5), (T,1)>; total revenue 6
  - <(T,1), (R,9)>; total revenue 10
- Transform into a dataset:

Type	Revenue	R before	R after	T before	T after	Sum	Sum R	Sum T	Mean R	Mean T
R	5	0	0	0	1	0	0	0	NA	1
T	1	1	0	0	0	5	5	0	NA	NA
T	1	0	1	0	0	0	0	0	9	NA
R	9	0	0	1	0	1	0	1	NA	NA

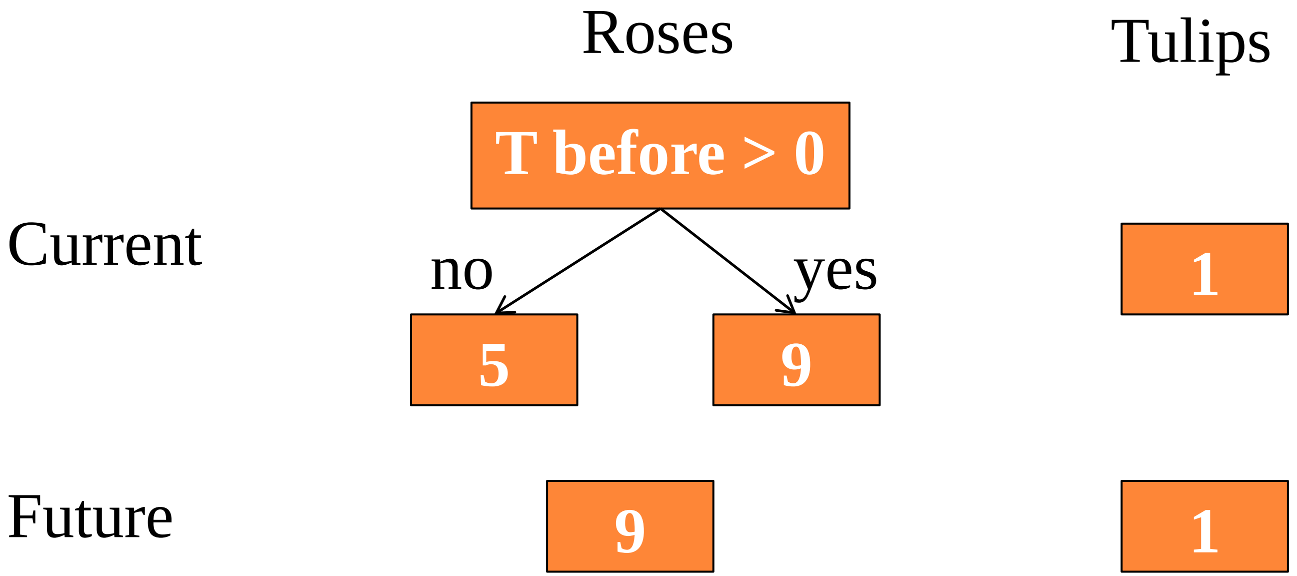
Regression trees

- Two decision tree models for expected revenue per item type:



Regression trees

- The models compute for a given item with given previous and next items and sales, the expected revenue and influence on later revenues



Regression trees

- Suppose we want to order RTT, compute data for the first position:

Type	Revenue	R before	R after	T before	T after	Sum	Sum R	Sum T	Mean R	Mean T
R	?	0	0	0	2	0	0	0	?	?
T	?	0	1	0	1	0	0	0	?	?

Regression trees

- Use the models to estimate the ?s

Type	Revenue	R before	R after	T before	T after	Sum	Sum R	Sum T	Mean R	Mean T
R	5	0	0	0	2	0	0	0	9	1
T	1	0	1	0	1	0	0	0	9	1

Regression trees

- Compute the expected revenue of putting R first:
  - 5 + 2 * 1 = 7
- And T first:
  - 1 + 1 * 9 + 1 * 1 = 11

Type	Revenue	R before	R after	T before	T after	Sum	Sum R	Sum T	Mean R	Mean T
R	5	0	0	0	2	0	0	0	9	1
T	1	0	1	0	1	0	0	0	9	1

- So put T first, and iterate

Regression trees

- Data for the second position:

Type	Revenue	R before	R after	T before	T after	Sum	Sum R	Sum T	Mean R	Mean T
R	?	0	0	1	1	1	0	1	?	?
T	?	0	1	1	0	1	0	1	?	?

Regression trees

- The expected revenue of putting R second:
  - 1 + 9 + 1 = 11
- And T first:
  - 1 + 1 + 9 = 11

Type	Revenue	R before	R after	T before	T after	Sum	Sum R	Sum T	Mean R	Mean T
R	9	0	0	1	1	1	0	1	9	1
T	1	0	1	1	0	1	0	1	9	1

- So pick one randomly, and iterate
- Solution: TRT

Why machine learning?

- Many auctions are held every day, leading to a large number of data sets
- Machine learning generalizes over items and their revenues
  - possible to evaluate orders not seen before
- Regression learns quickly and is robust to noise

- In experiments on artificial data our method is able to come up with orders that obtain a higher revenue than any historic order…

Some experiments

- 4 myopic agents:

	Rose	Tulip	R+T	Budget
Agent 1	10	-	-	10
Agent 2	6	-	-	100
Agent 3	8	5	-	20
Agent 4	4	3	20	100

- Example:

	R	T	T	T	R	R	T	R
Sold to	1	3	3	3	2	2	3	2
Revenue	10	5	5	5	6	6	5	6

Some experiments

- 4 myopic agents:

	Rose	Tulip	R+T	Budget
Agent 1	10	-	-	10
Agent 2	6	-	-	100
Agent 3	8	5	-	20
Agent 4	4	3	20	100

- Optimal:

	R	T	T	R	T	T	R	R
Sold to	1	3	3	3	4	4	4	4
Revenue	10	5	5	8	3	3	17	17

Some experiments

- Ordering 4 roses and 4 tulips:
  - Run 40 simulations using random orderings
  - Remove ordering with a revenue of 68 (optimal)
  - Translate the orderings + revenues to a data set
  - Learn models, and run the greedy algorithm

- In 8 out of 10 times, the result is an optimal sequence! (RRTTTTRR or RTTRRTTR)

Some more experiments

- 5 types of myopic agents:

	Rose	Tulip	Lily	Orchid	R+T	R+L	Budget
Agent 1	8-12	-	-	-	-	-	10-30
Agent 2	-	3-7	6-10	-	-	-	10-30
Agent 3	-	-	-	13-17	-	-	10-30
Agent 4	2-6	2-6	-	-	20-25	-	10-30
Agent 5	2-5	-	2-5	-	-	26-30	10-30

- Every auction
  - the participants are randomly selected
  - utility functions and budgets are drawn uniformly
  - flowers are randomly selected

Some more experiments

- Run 250 simulations using random orderings
- Create data set, learn models, run greedy
- Test on 250 new simulations
- Result:

ordering	mean revenue
random	131
our method	135
best found	136.5
most valuable first	128

CONCLUSIONS

- Regression methods from machine learning are used to estimate the revenue of orderings of itmes in sequential auctions
- Our method learn quickly, and are able to capture revenue increasing rules in complex, noisy settings

- We cannot wait to test it on real data…