So Who Won?
Dynamic Max Discovery with the Crowd

Stephen Guo, Aditya Parameswaran, Hector Garcia-Molina

Vipul Venkataraman
Sep 9, 2015

Outline

- Why Crowdsourcing?
- Finding Maximum
  - Judgement Problem
  - Next Votes Problem
- Conclusion
- References

Why Crowdsourcing?

To solve problems that are difficult for computers

- Sort / Max [1]
- Graph search [5]
- Categorize [4]
- Filter [3]

Tradeoffs [3]

- Latency
- Cost
- Uncertainty

Why Crowdsourcing?

To solve problems that are difficult for computers

- Sort / Max [1]

focus of this talk

- Graph search [5]
- Categorize [4]
- Filter [3]

Tradeoffs [3]

- Latency
- Cost
- Uncertainty

Closest point?

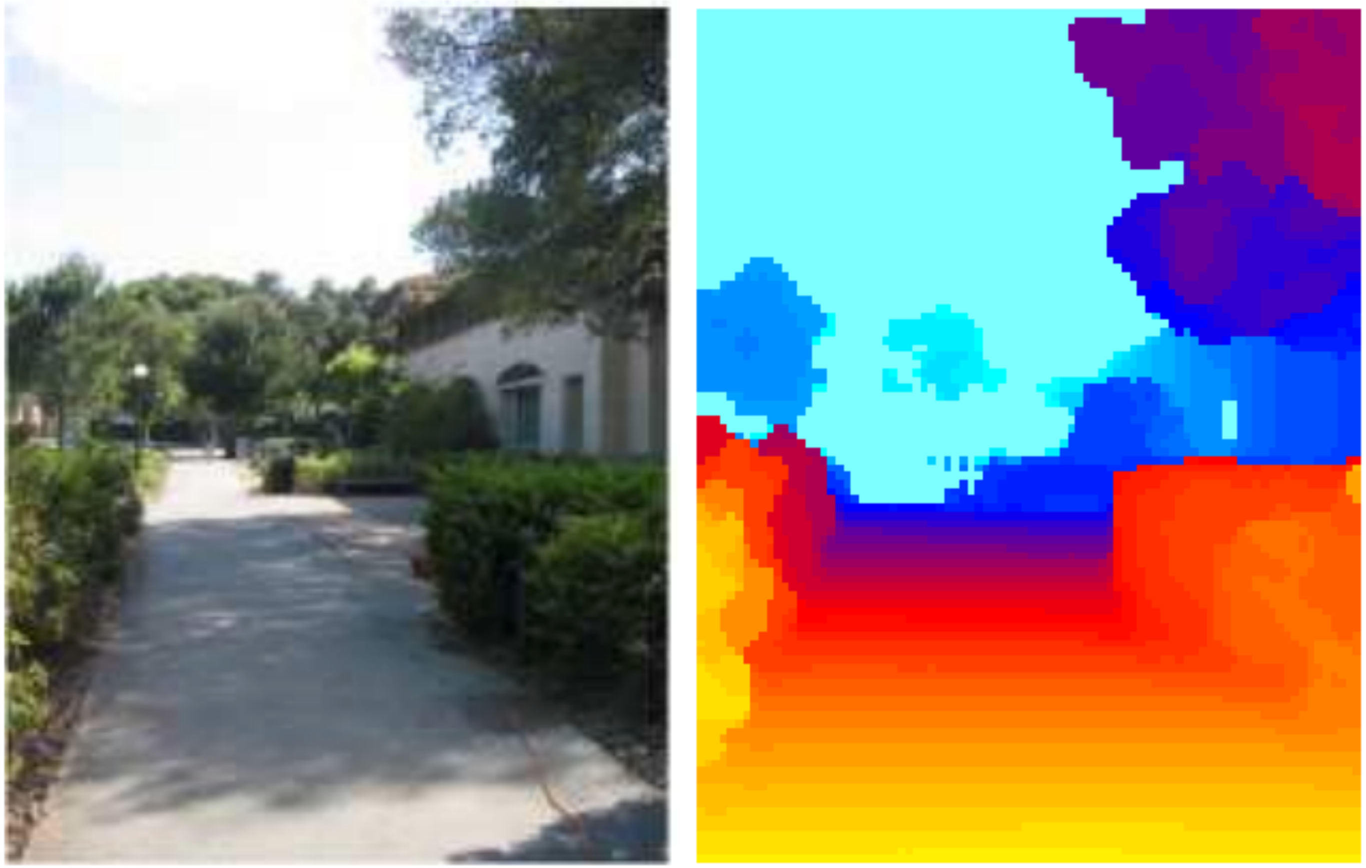
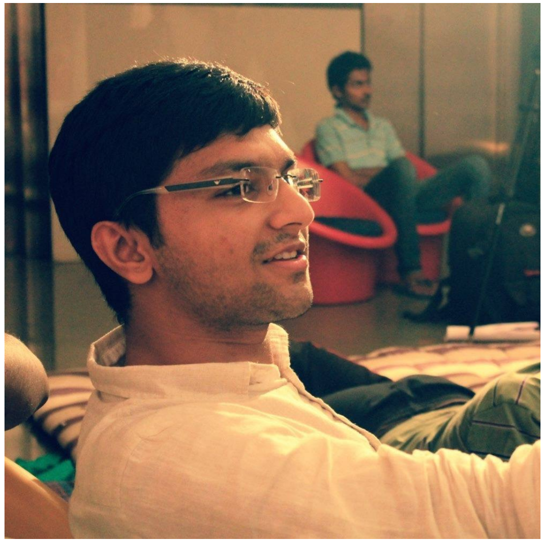
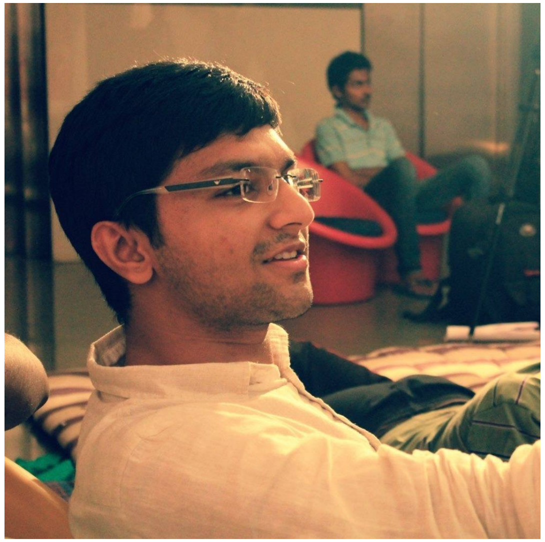


Figure: Make 3D [2]

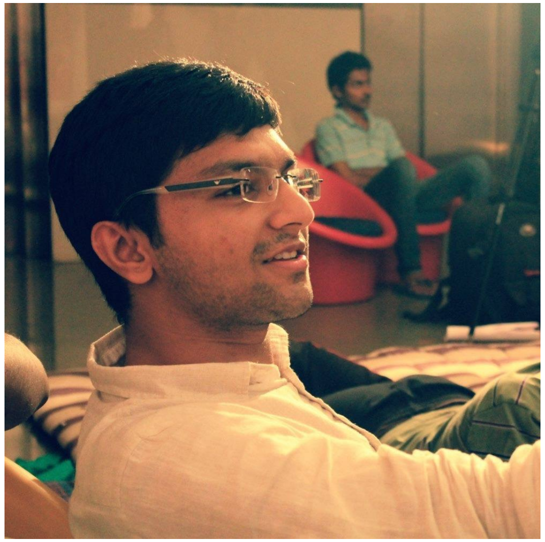
Best profile picture?



Best profile picture?



Best profile picture?



Max Problem

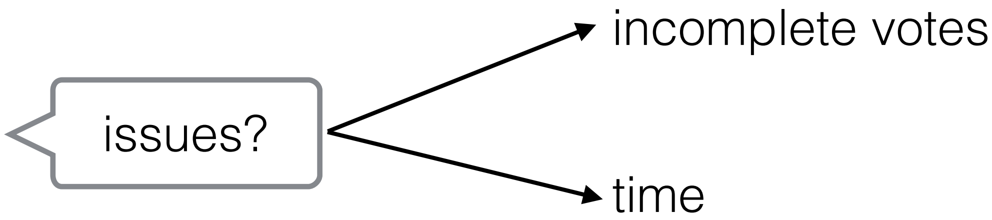
- Goal: Find object with maximum *quality*
- How: ask pairwise comparisons - votes
- Workers vote correctly with probability p
- Variants
  - Structured
  - Unstructured

Max Problem

- Goal: Find object with maximum *quality*
- How: ask pairwise comparisons - votes
- Workers vote correctly with probability p
- Variants
  - Structured
    issues?
  - Unstructured

Max Problem

- Goal: Find object with maximum *quality*
- How: ask pairwise comparisons - votes
- Workers vote correctly with probability p
- Variants
  - Structured
  - Unstructured



Unstructured setting

- *Judgement Problem*: what is our current best estimate for the overall max winner?
- *Next Votes Problem*: how to choose most effective votes to invoke, given current standing?

Unstructured setting

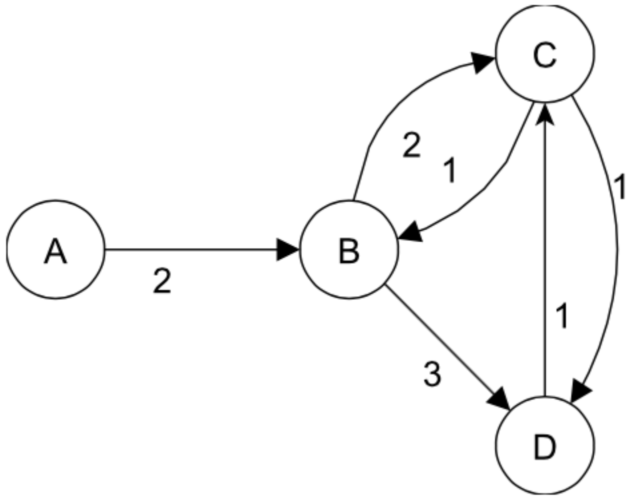
Both problems are:

- NP-Hard (why? see [1] for details)
- Good heuristics exists! #phew
  - More on this soon

Judgement Problem

Current best estimate for the overall max winner?

Representation: weighted directed graphs



$$W = \begin{pmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 3 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure: [1]

ML Formulation

Let π denote a permutation function

For object i, π(i) denotes its rank

$$P(\boldsymbol{\pi}^{-1}(k) = j|W) = \frac{\sum_{d:\pi_d^{-1}(k)=j} P(W|\pi_d)}{\sum_l P(W|\pi_l)}$$

ML Formulation: Given W and p, determine:

$$\arg\max_j P(\boldsymbol{\pi}^{-1}(1) = j|W)$$

Heuristic Strategies

- Indegree Strategy
- Local Strategy
- PageRank Strategy
- Iterative Strategy

Indegree Strategy

- Need to know worker accuracy p
- Scoring function - represents number of in-degrees
- Transform graph such that $l(i,j) + l(j,i) = 1$
- $l(j,i) = P(\pi(i) < \pi(j) \mid w(i,j), w(j,i))$
- Find node with highest sum of in-degree weights

Local Strategy

Use local evidence

$$wins(i) = \sum_j w_{ji} \qquad losses(i) = \sum_i w_{ij}$$

$$s(i) = wins(i) - losses(i) + \sum_j [\mathbf{1}(w_{ji} > w_{ij}) wins(j)] - \sum_j [\mathbf{1}(w_{ij} > w_{ji}) losses(j)]$$

We now consider evidence 2 steps away

PageRank Strategy

Use global evidence

$$pr_{t+1}(i) = \sum_j \frac{w_{ji}}{d^+(j)} pr_t(j) \qquad d^+(i) = \sum_j w_{ij}$$

The probability masses concentrate on the Strongly Connected Components

Subtle differences from original PageRank

Does not converge. How to handle this?

Iterative Strategy

Rank objects using a scoring metric (which one?)

Remove lower ranked objects

Repeat until final object is obtained

Any metric can be used

Iterative Strategy

Rank objects using a scoring metric (which one?)

Remove lower ranked objects

Repeat until final object is obtained

Any metric can be used, eg: $wins(i) - losses(i)$

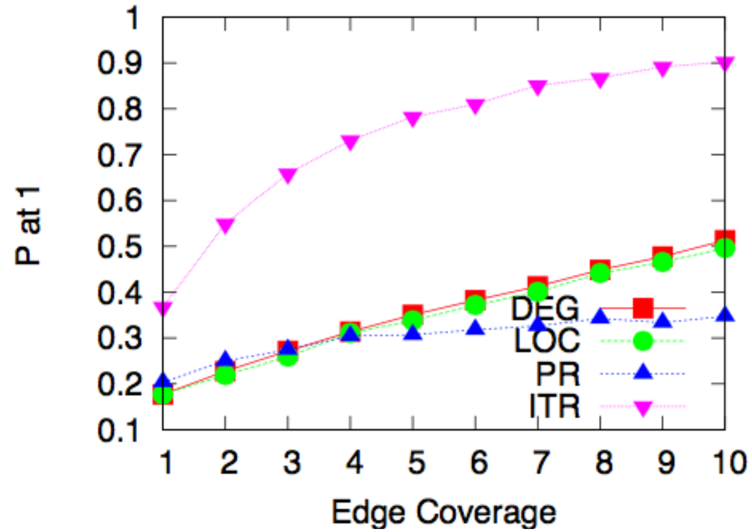
Comparison

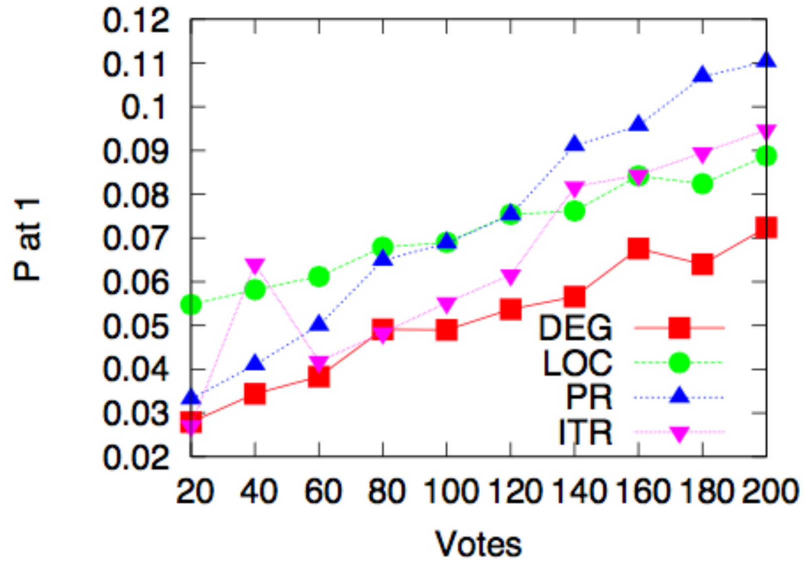
Heuristic	Prediction
ML	(D, C, B, A) and (C, D, B, A)
Indegree	(D, C, B, A)
Local	(D, C, B, A)
PageRank	Maximum object = C
Iterative	$(C, D, B, A), (C, D, A, B), (D, C, B, A)$, or (D, C, A, B)

Table: [1]

Experiments

- ML has best performance
- Iterative is best when votes sampled are high
- PageRank bad with low worker accuracy
- PageRank best when votes sampled are low, with high worker accuracy



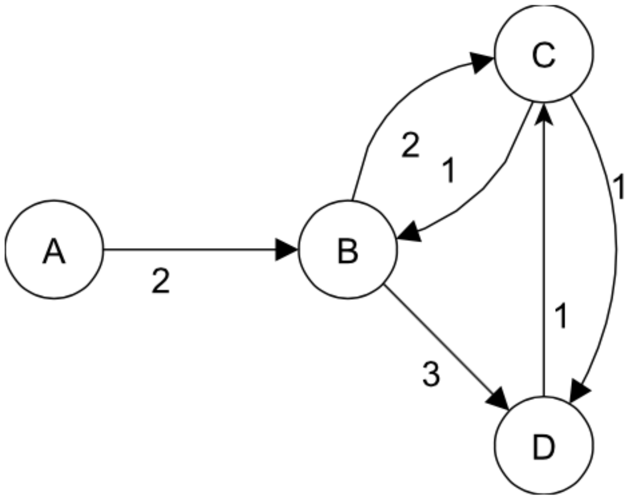


Figures: [1]

Next Votes Problem

Given current standing and an additional vote budget b, what votes to invoke?

Adaptive v One-shot



$$W = \begin{pmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 3 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure: [1]

ML Formulation

Q: a vote multiset; |Q| = b

A(Q): corresponding answer multiset for Q

$$P_{max}(a \wedge W) = \max_i P(\boldsymbol{\pi}^{-1}(1) = i | a \wedge W)$$

ML Formulation: Given b, W and p, determine Q, that maximizes:

$$\sum_{a \in A(Q)} \max_i P(\boldsymbol{\pi}^{-1}(1) = i, a \wedge W)$$

Evaluation

We use the following framework to evaluate additional votes:

- Use W to score all objects with a scoring function
- Select a batch of *b* votes to request
- Compute new scores and find maximum object

Evaluation

We use the following framework to evaluate additional votes:

- Use W to score all objects with a scoring function
- Select a batch of *b* votes to request
- Compute new scores and find maximum object

which one?

Evaluation

We use the following framework to evaluate additional votes:

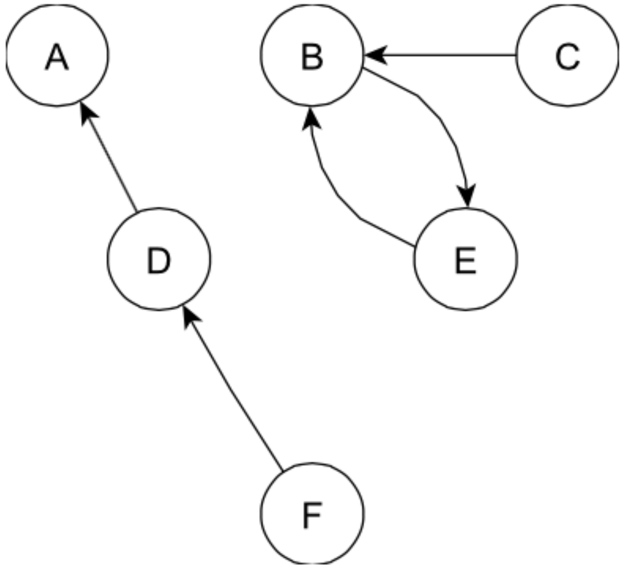
- Use W to score all objects with a scoring function
- Select a batch of *b* votes to request
- Compute new scores and find maximum object

which one?

which one?

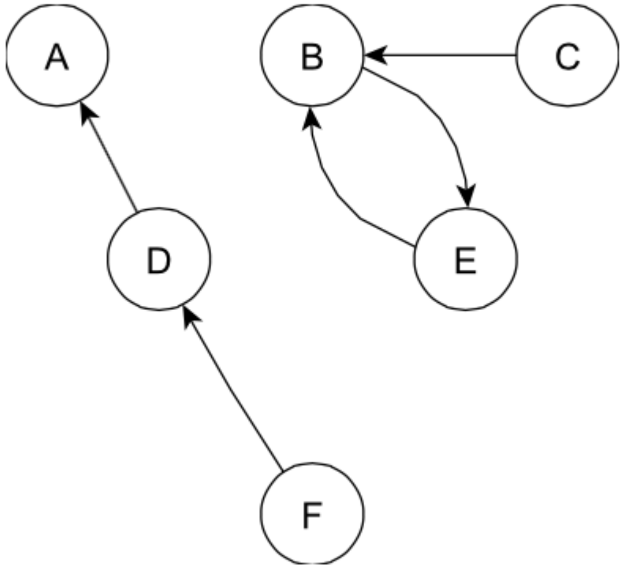
Heuristic Strategies

- Paired Vote Selection
- Max Vote Selection
- Greedy Vote Selection
- Complete Tournament Vote Selection



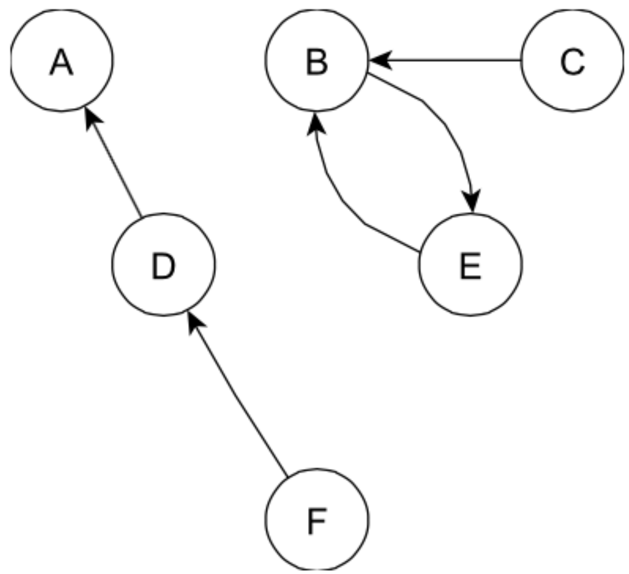
Paired Vote Selection

- Greedy
- No object chosen twice
- Performs well when objects have similar scores
- Good/bad - why?



Max Vote Selection

- More focus on find top ranked object
- Should be better than Paired Vote
- Good/bad - why?

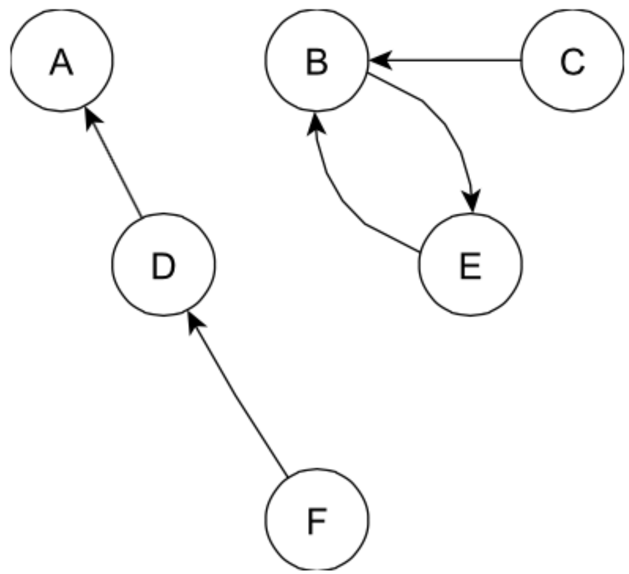


Greedy Vote Selection

- Find product of scores of pairs

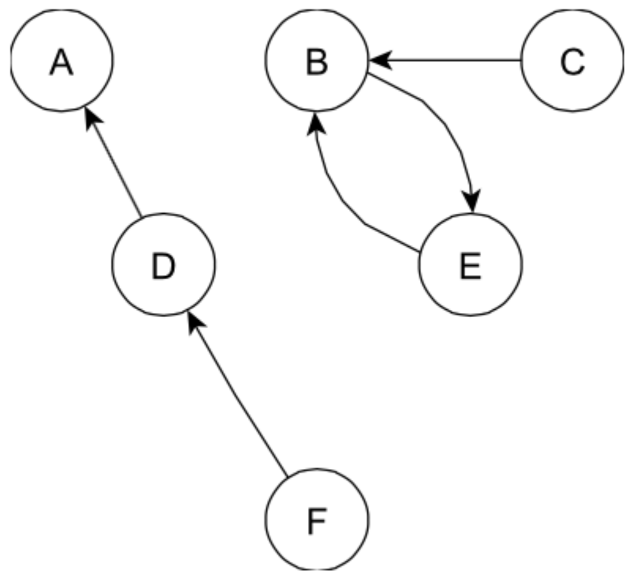
- Choose *b* highest weighted pairs

- Good/bad - why?



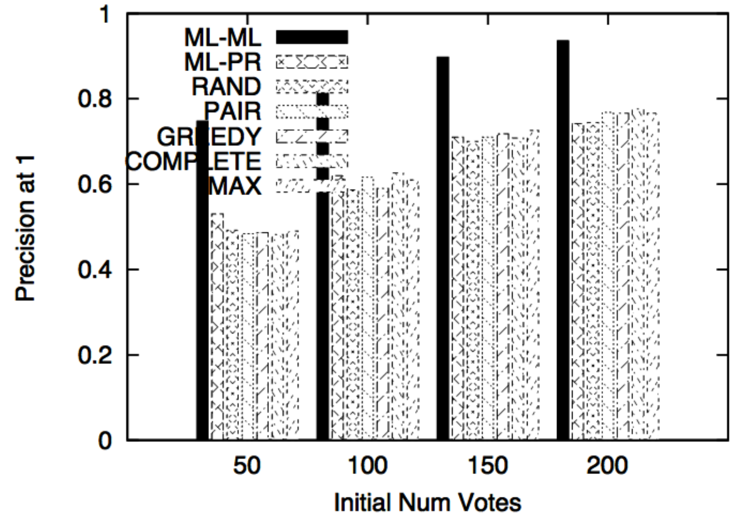
Complete Tournament

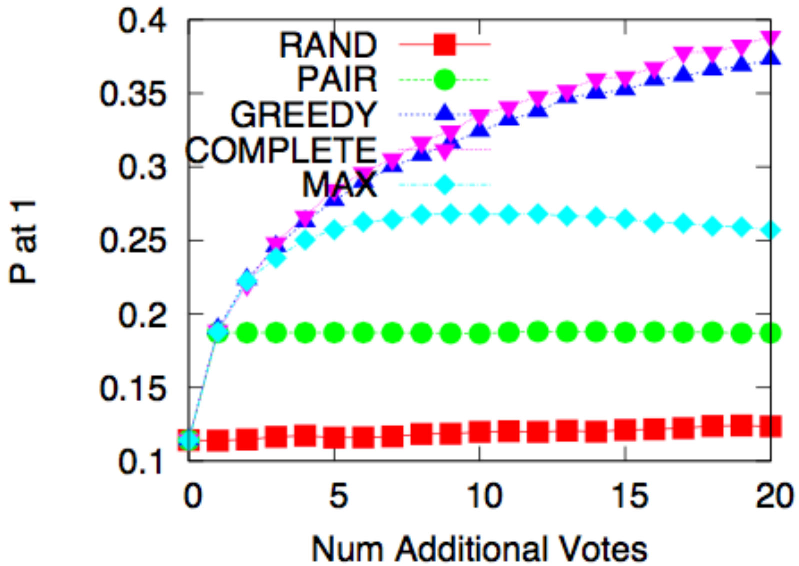
- Take top K objects
- Do round-robin among them (choose K accordingly)
- Given K, should we choose an even lower value? Why?
- Good/bad - why?



Experiments

- ML-ML has best performance
- Prediction performance increases with *b* (concave)
- Prediction performance increases with *p* (convex)
- Complete Tournament and Greedy are clear winners



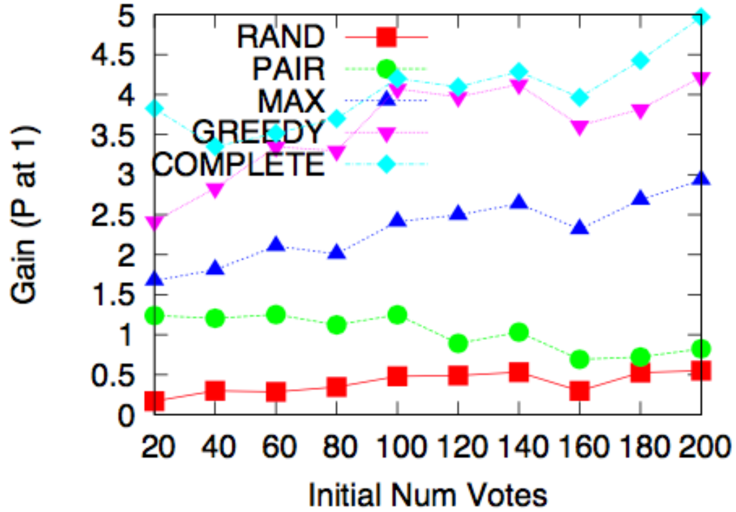


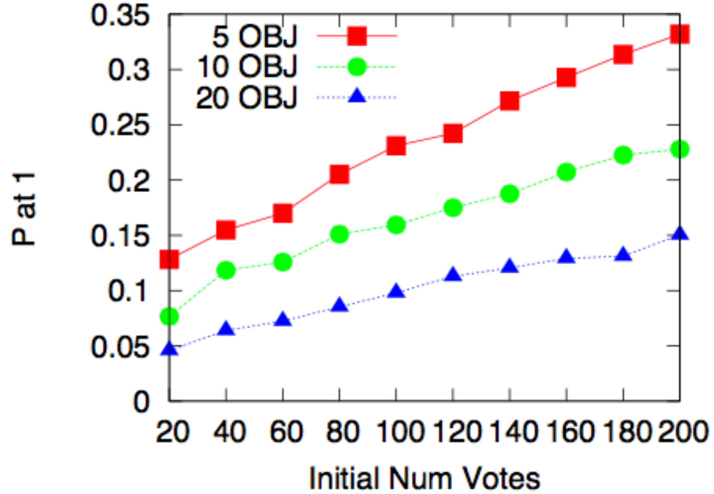
Figures: [1]

Experiments

Greedy v Complete Tournament

- Objects of different types
- Initial votes across same types more likely
- We assume no initial votes across different types
- Complete Tournament is the winner
- Complete Tournament works better with fewer objects





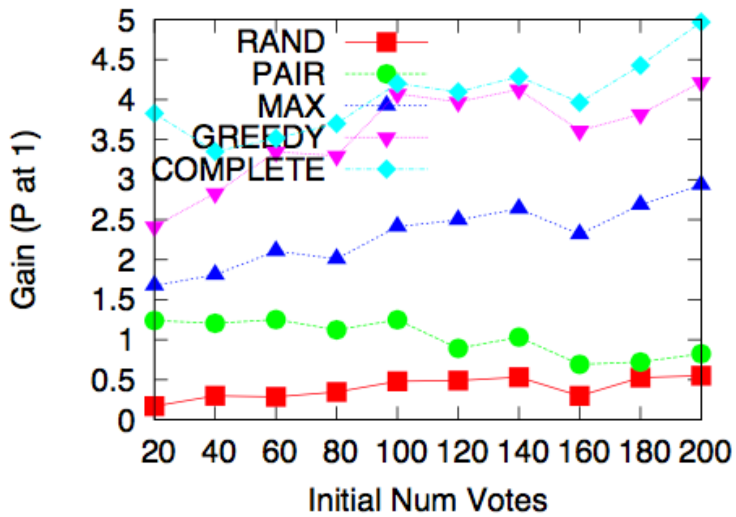
Figures: [1]

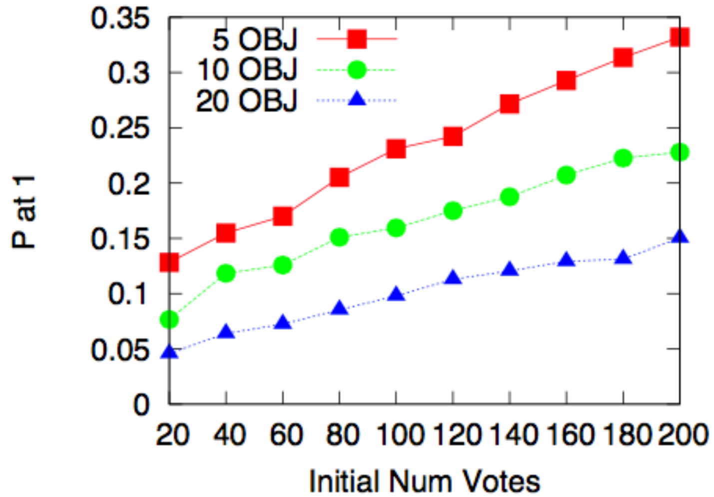
Experiments

Greedy v Complete Tournament

- Objects of different types
- Initial votes across same types more likely
- We assume no initial votes across different types
- Complete Tournament is the winner
- Complete Tournament works better with fewer objects

why?





Figures: [1]

Discussion

- Can the first round of votes be invoked better?
- Theoretical basis for the behavior of heuristics
- Why does PageRank work well with fewer initial votes?
- Why is Complete Tournament better than Greedy?

Conclusion

- Judgement Problem
- Next Votes Problem
- Effective Heuristics
  - PageRank
  - Complete Tournament

References

[1] So who won?: dynamic max discovery with the crowd, S Guo, A Parameswaran, H Garcia-Molina

[2] Learning Depth from Single Monocular Images, Ashutosh Saxena, Sung H. Chung, Andrew Y. Ng

[3] CrowdScreen: Algorithms for Filtering Data with Humans. Aditya Parameswaran, Hector Garcia-Molina, Hyunjung Park, Neoklis Polyzotis, Aditya Ramesh and Jennifer Widom

[4] Human-assisted Graph Search: It's Okay to Ask Questions. Aditya Parameswaran, Anish Das Sarma, Hector Garcia-Molina, Neoklis Polyzotis and Jennifer Widom

[5] Active Sampling for Entity Matching. Kedar Bellare, Suresh Iyengar, Aditya Parameswaran and Vibhor Rastogi

Questions?

THANKS FOR LISTENING

AND

KEEP CLAPPING