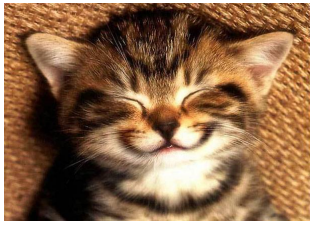
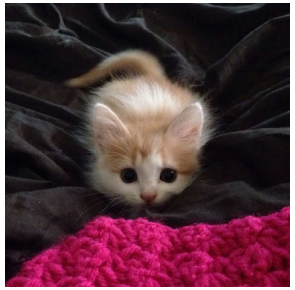
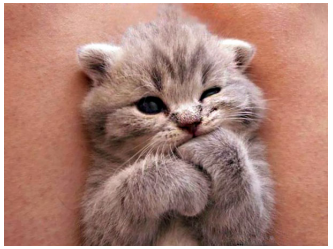
So Who Won: Dynamic Max Discovery with the Crowd

Assma Boughoula

Motivation

Which kitten is **cutest**?




Which dog is **largest**?

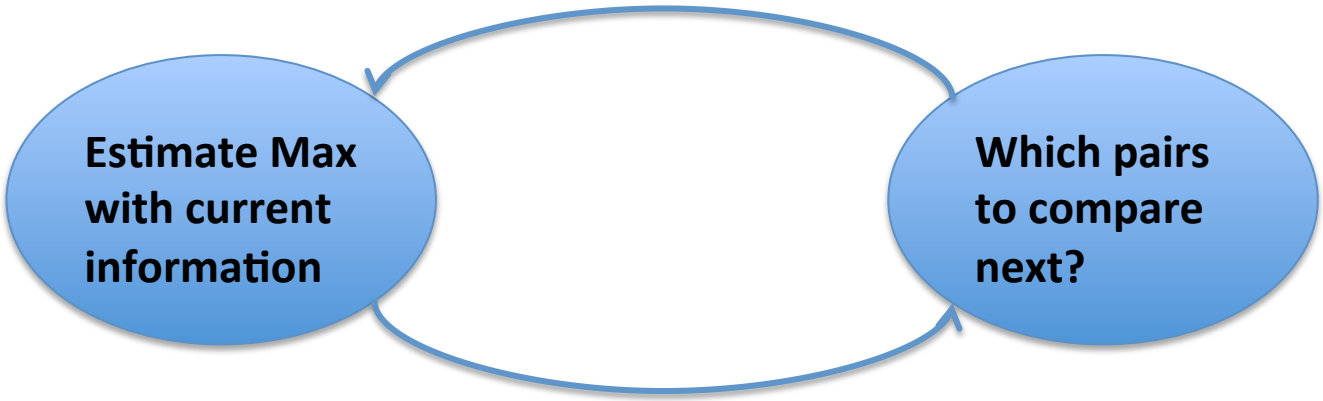


Problem Statement

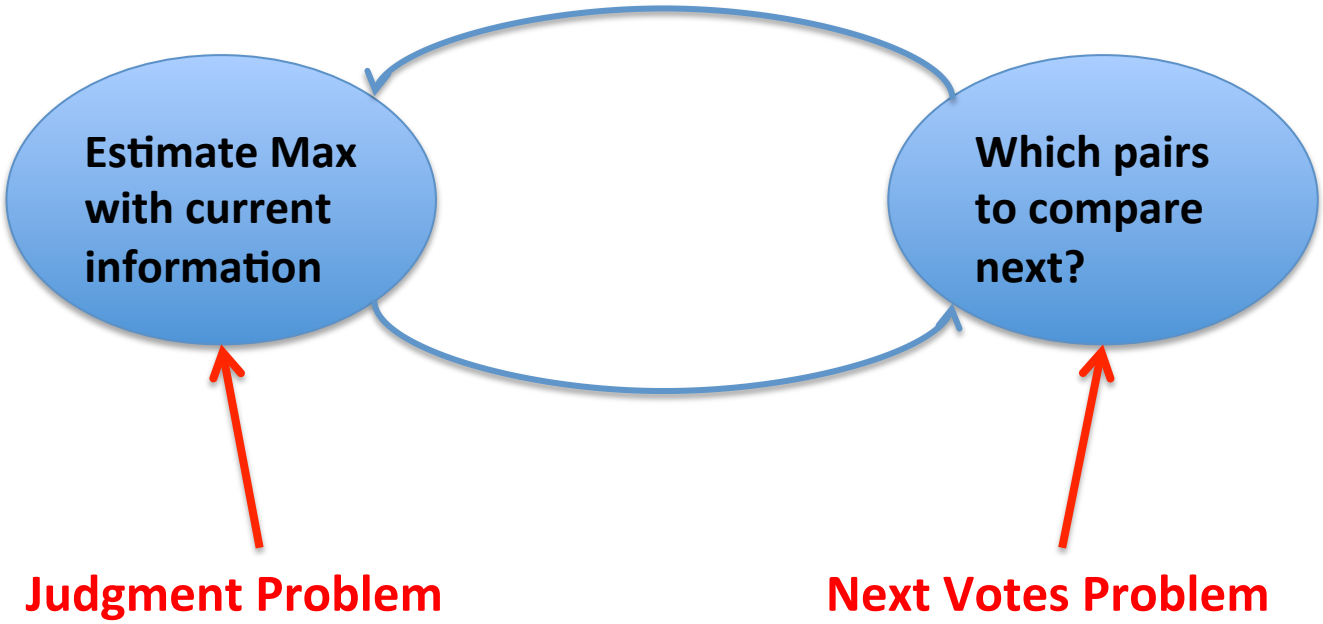
- Set O of n objects $\{o_1, \ldots, o_n\}$
- Each o_i has quality c_i
- π : permutation or ranking of O
- If $\pi(i) < \pi(j)$ then o_i is ranked higher than o_j and $c_i > c_j$
- W : $n \times n$ matrix of votes from workers
- Assumptions:
  - $c_i \neq c_j$ for all $i \neq j$
  - Workers can only compare a pair of objects
- **Goal**: arrive at the correct permutation π quickly and efficiently

Framework

- Two sub-problems



Framework

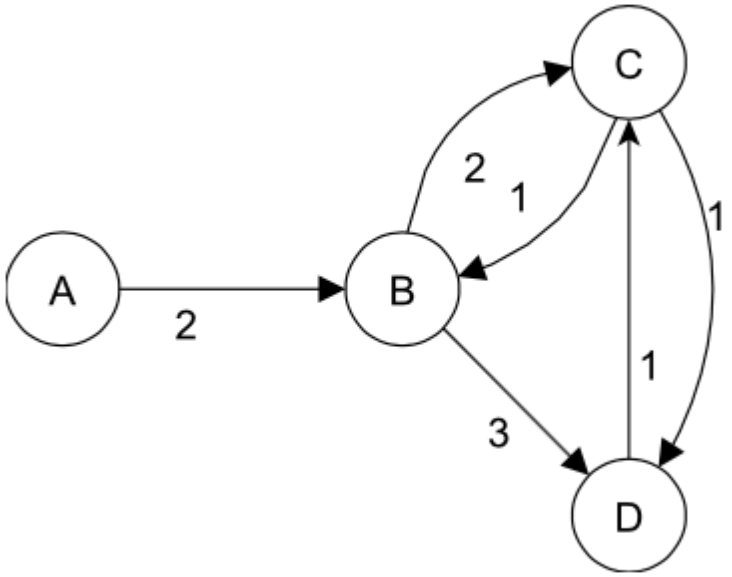

Estimate Max with current information
Which pairs to compare next?
Judgment Problem
Next Votes Problem

Outline

- Judgment Problem
  - Maximum Likelihood (ML) baseline
  - Heuristic Strategies
    - Indegree
    - Local
    - PageRank
    - Iterative
- Next Votes Problem
  - Maximum Likelihood (ML) baseline
  - Heuristic Vote Selection Strategies
    - Paired
    - Max
    - Greedy
    - Round Robin

Judgment Problem

$$W = \begin{array}{c} \begin{array}{cccc} \textbf{A} & \textbf{B} & \textbf{C} & \textbf{D} \end{array} \\ \begin{pmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 3 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \begin{array}{l} \textbf{A} \\ \textbf{B} \\ \textbf{C} \\ \textbf{D} \end{array}$$



Judgment Problem

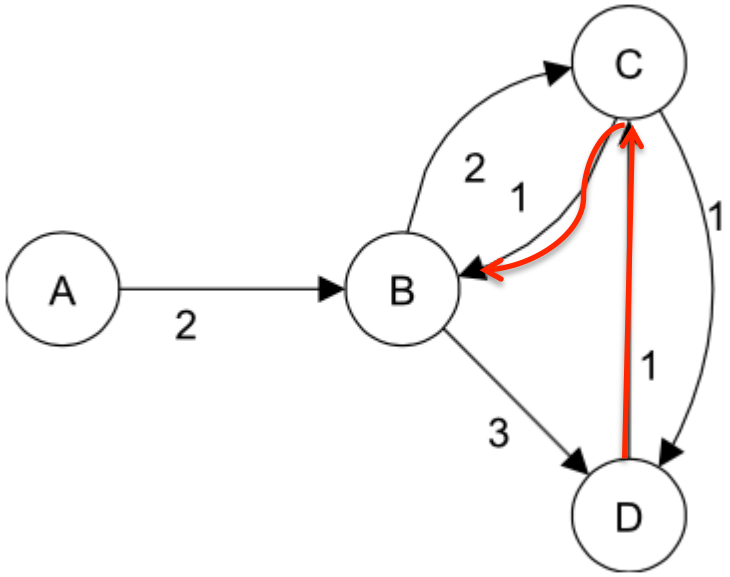
Maximum Likelihood

- Average worker accuracy *p* is known
- Calculate $P(\pi_d | W)$ for each permutation π_d
- Choose most probable permutation
  In example: *(D, C, B, A) (C, D, B, A)*
- There are *n!* possible permutations!!!
  Exhaustive, inefficient

Judgment Problem

Kemeny permutation:

- Minimizes number of edges *i-j* that don't respect ranking.
- *(D, C, B, A)*

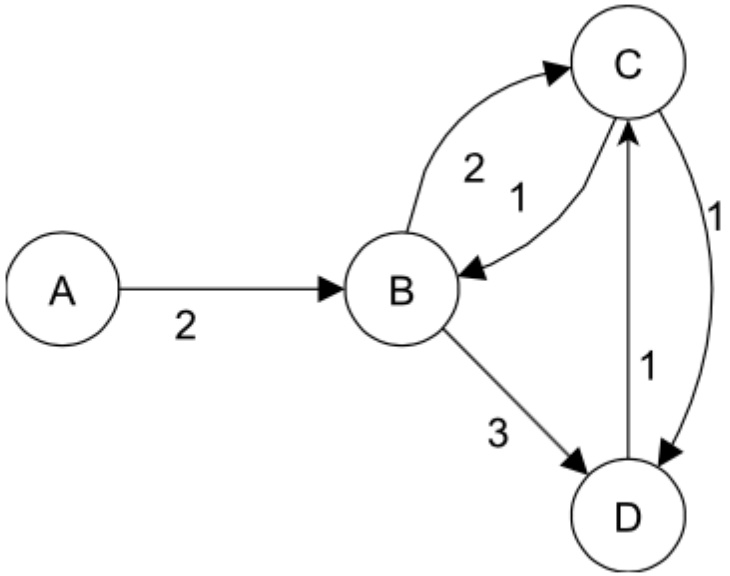


Judgment Problem

Indegree Strategy

- For each node i, look at all neighbors j:
- Calculate score:
  $s(i)= P(i<j \mid w_{ij}, w_{ji})$
- Choose max s(i)

Only considers neighbors, not whole W

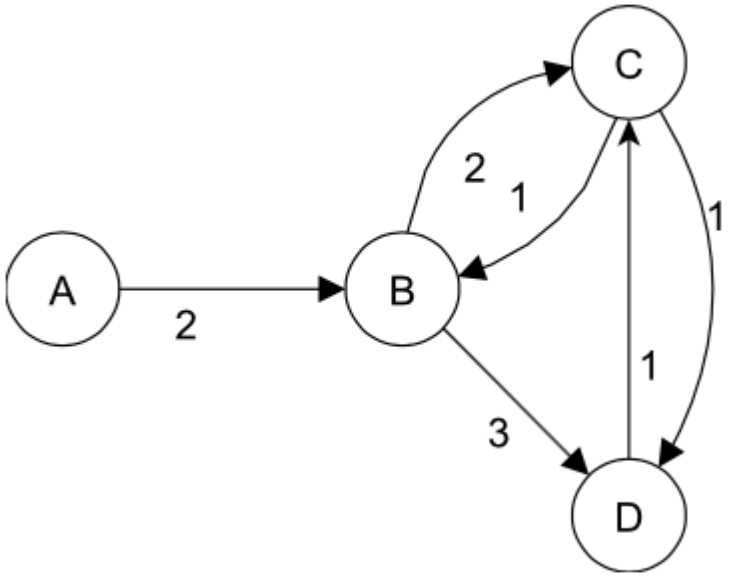


Judgment Problem

Local Strategy

- Considers "neighbors of neighbors"
- $wins(i) = \Sigma_j w_{ji}$ $losses(i) = \Sigma_j w_{ij}$
- $s(i) = wins(i) - losses(i) + \Sigma_{j:ci>cj} wins(j) - \Sigma_{j:ci<cj} losses(j)$

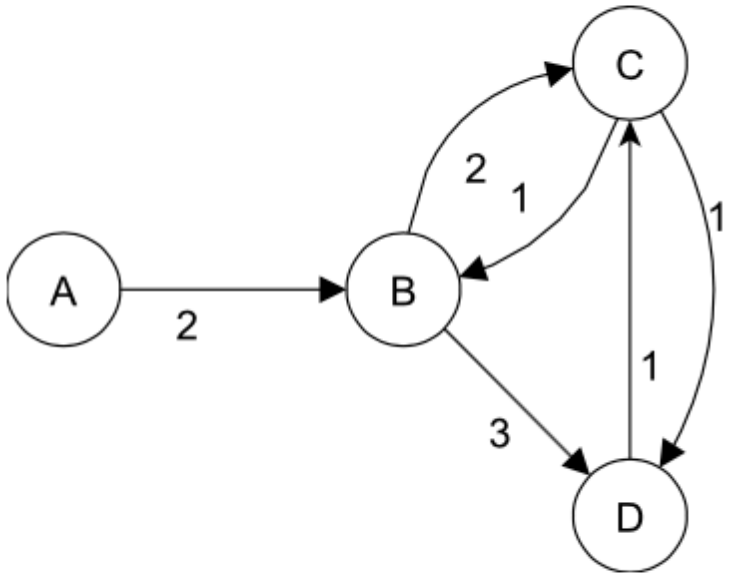
Reward **Penalty**



Judgment Problem

PageRank Strategy

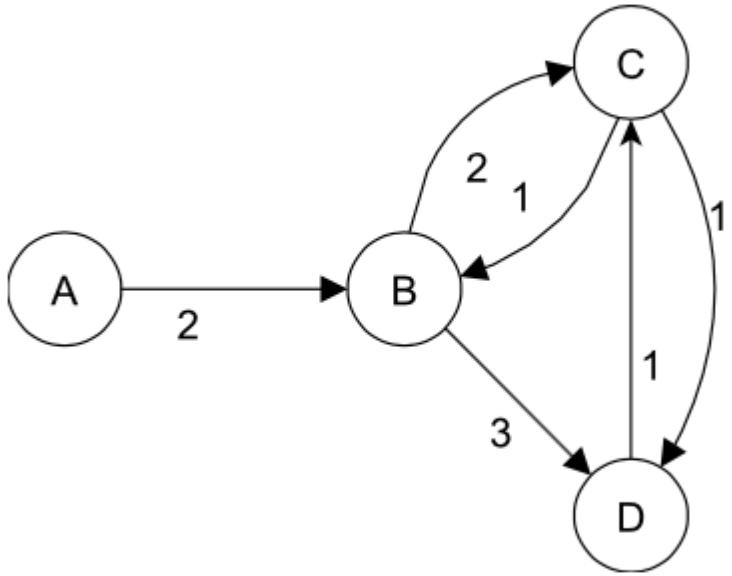
- Considers whole graph
- *S(i) = PageRank(i)*
- $Pr_{t+1}(i)$ $=\Sigma_j w_{ji}\, pr_t(j)/outdeg(j)$
- $Pr_0(i) = 1/n$



Judgment Problem

Iterative Strategy

- *dif(i) = wins(i) − losses(i)*
- Eliminate lower half of objects according to *dif(i)* scores
- Repeat with remaining objects



Judgment Problem

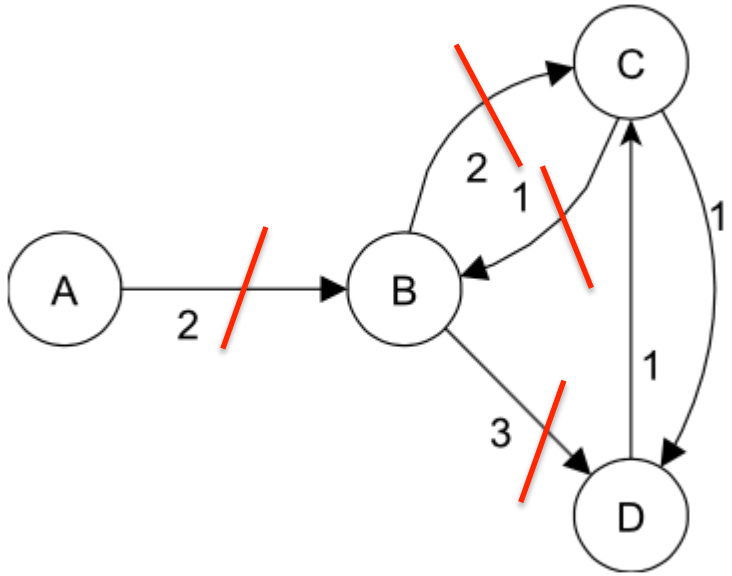
Iterative Strategy

Iter1:

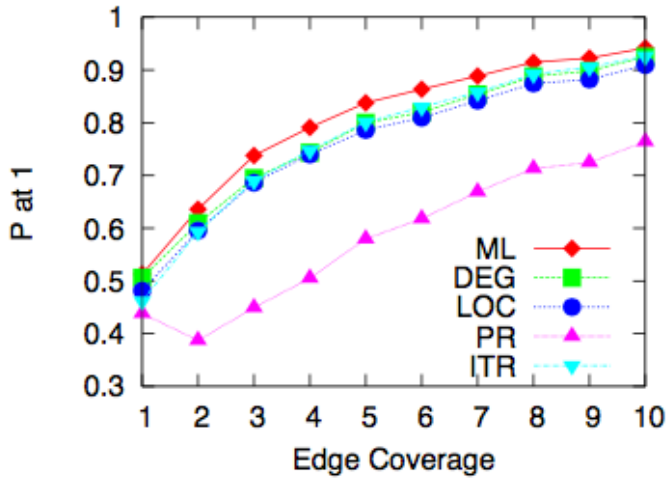
dif(A)=-2, dif(B)=-4,
dif(C)=1, dif(D)=3

Iter2:

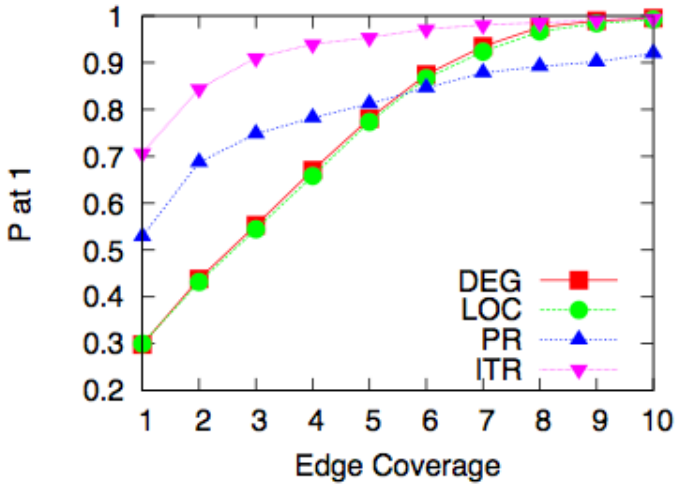
dif(C)=0, dif(C)=0



Judgment Problem

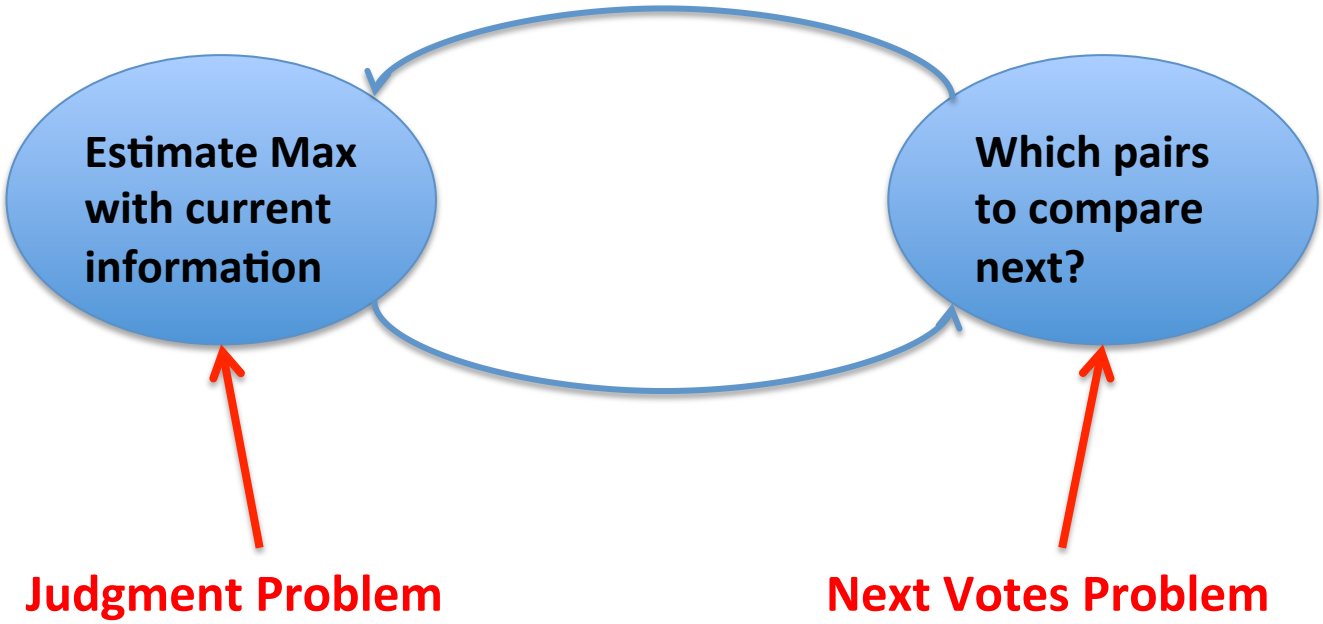


p=0.75



p=0.95

Framework


Estimate Max with current information
Which pairs to compare next?
Judgment Problem
Next Votes Problem

Next Votes Problem

Some Notation…

- Budget b = # votes to be requested
- Potential vote $\{o_i, o_j\}$
- $Q = \{\{o_i, o_j\}, \ldots\}$ possible multiset of b votes
- Answer tuple $(\{o_i, o_j\}, o_x)$
- Answer multiset $a=\{(\{o_i, o_j\}, o_x), \ldots\}$
- 2^b possible answer multisets for each Q *!!!*
- $a \curlywedge W$: updated W matrix

Next Votes Problem

Maximum Likelihood

- Look for Q such that $a \wedge W$ increases our confidence when estimating the max the most.
- Choose Q that maximizes:
  $\Sigma_a max_i P(i \text{ is max}, a \wedge W)$
- Assumes average worker accuracy p known

Next Votes Problem

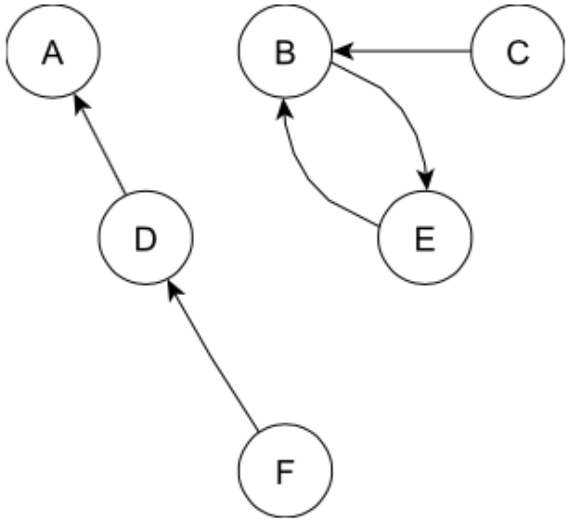
Paired Vote Selection

- Order objects by score *s(i)* from judgment problem

b = 2

Q = (A,B), (E,D)

(A, B, E, D, C, F)



Next Votes Problem

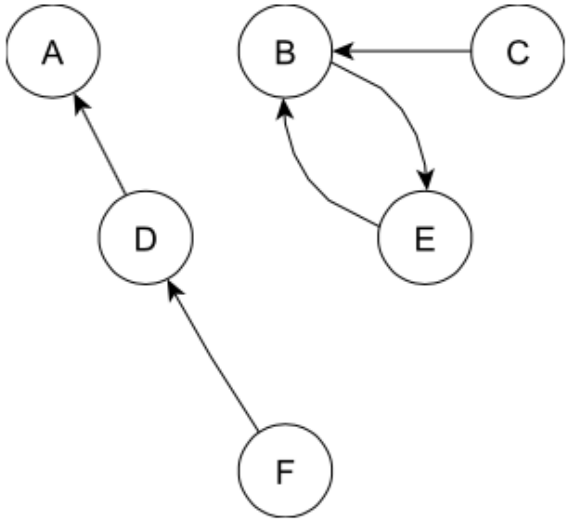
Max Vote Selection

- Compare with current max

b = 2

Q = (A,B), (A,E)

(A, B, E, D, C, F)



Next Votes Problem

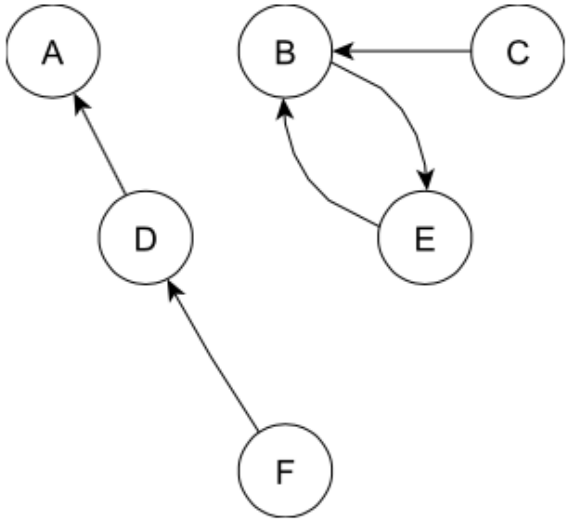
Greedy Vote Selection

- Sort objects by s(i)
- *s(i) x s(j)* for each pair *(i,j)*
- Select b pairs with highest *s(i) x s(j)*

(A, B, E, D, C, F)

S=(1,1,0,0,-1,-1)

((A,B),(A,E),(A,D),(B,E)…)



Next Votes Problem

Complete (Round Robin) Vote Selection

- K = size of tournament

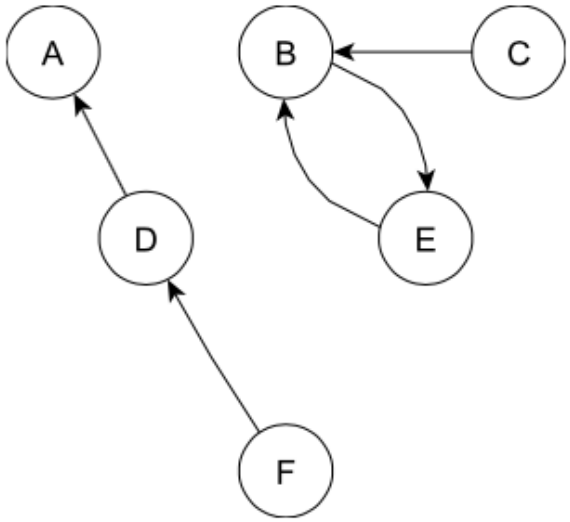
b=6

K=3

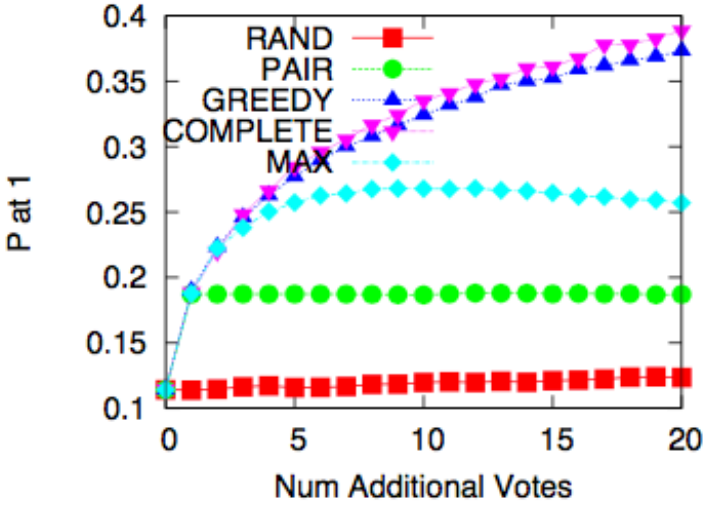
(A, B, E, D, C, F)

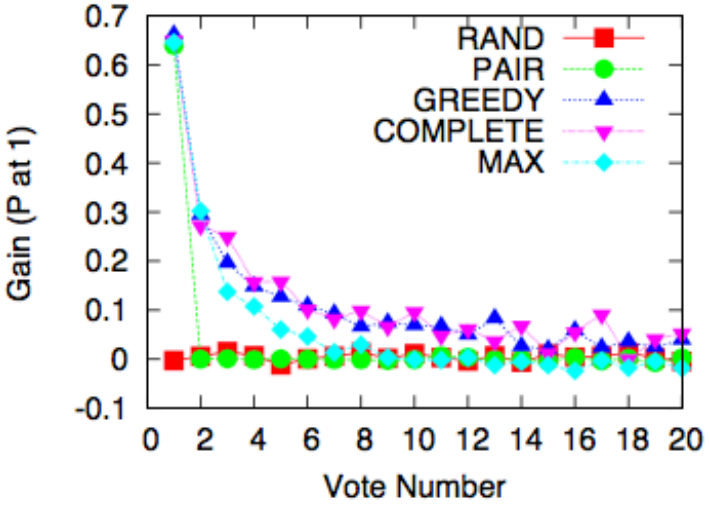
S=(1,1,0,0,-1,-1)

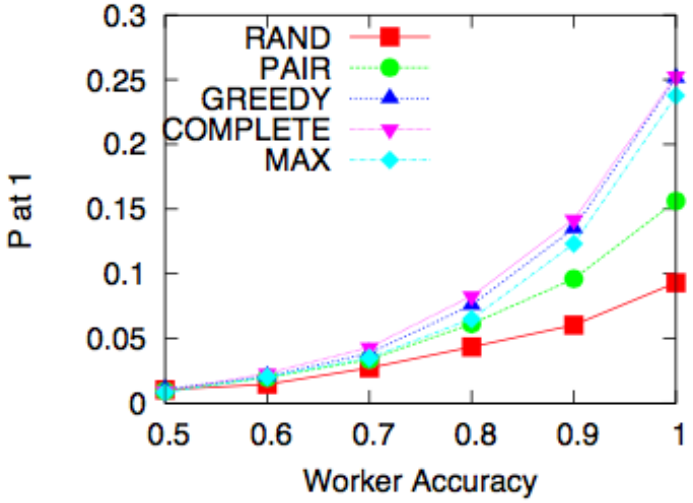
((A,B),(A,E),(B,E),(A,D),(B,D),(E,D))



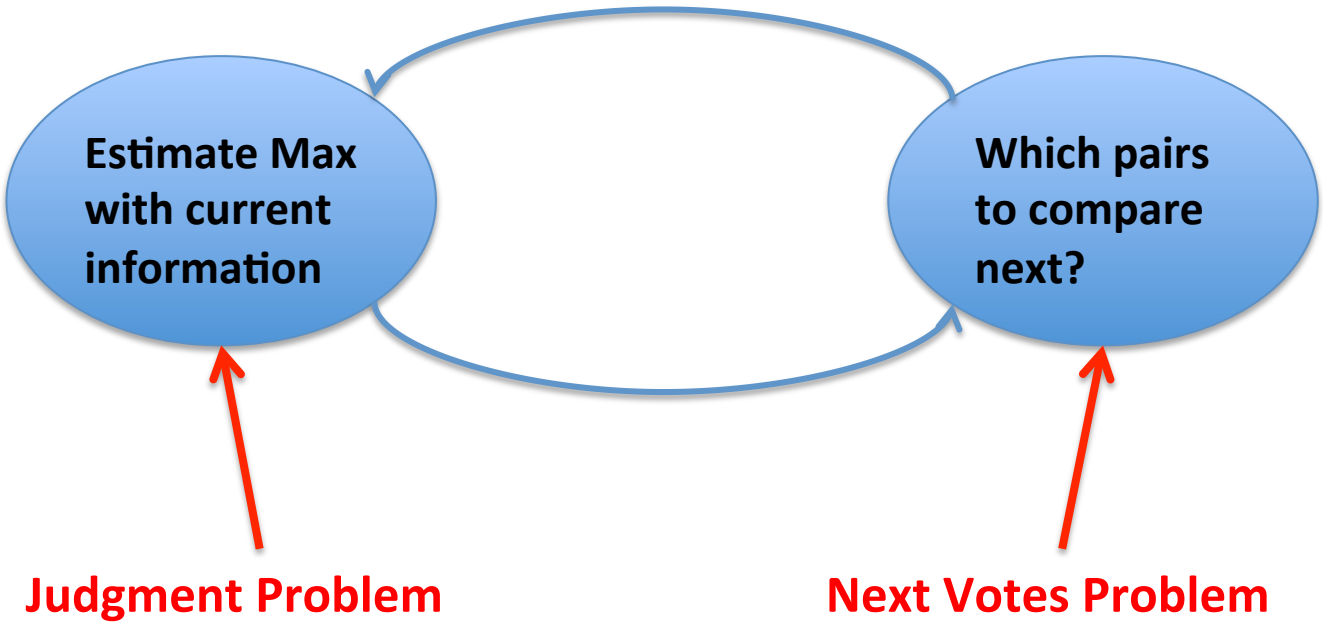
Next Votes Problem







Recap


Estimate Max with current information
Which pairs to compare next?
Judgment Problem
Next Votes Problem

Recap

- Judgment Problem
  - Maximum Likelihood (ML) baseline
  - Heuristic Strategies
    - Indegree
    - Local
    - **PageRank**
    - **Iterative**
- Next Votes Problem
  - Maximum Likelihood (ML) baseline
  - Heuristic Vote Selection Strategies
    - Paired
    - Max
    - **Greedy**
    - **Round Robin**