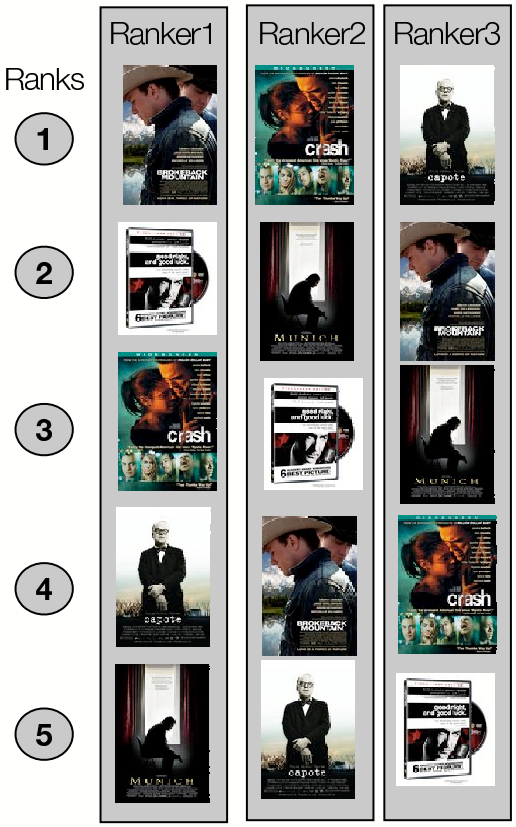
The Impact of Ranker Quality on Rank Aggregation Algorithms: Information vs. Robustness

Sibel Adalı, Brandeis Hill and Malik-Magdon Ismail
Rensselaer Polytechnic Institute

Motivation



- Given a set of ranked list of objects, what is the best way to aggregate them into a final ranked list?
  - The correct answer depends on the what the objective is.
    - The consensus among the input rankers
    - The most correct final ordering
- In this paper:

➡ We implement existing rank aggregation methods and introduce new ones.

➡ We implement a statistical framework for evaluating the methods and report on the rank aggregation methods.

Related Work

- Rank aggregation methods
  - Use of cheap methods such as average and median is common
  - Methods based on consensus introduced first by Dwork, Kumar, Naor and Sivakumar [WWW 2001] and median rank as an approximation by Fagin, Kumar, Sivakumar [SIGMOD 2003]
  - Methods that integrate rank and textual information are common in meta-searching, for example Lu, Meng, Shu, Yu, Liu [WISE 2005]
  - Machine learning methods learn the best factors for a user by incorporating user feedback, for example Joachims [SIGKDD 2002]
- Evaluation of rank aggregation methods are mainly with real data using fairly small data sets, for example Renda, Straccia [SAC 2003]

Error Measures

- Given two rankers A and B
  - Precision (p) finds the number of objects A and B in common (maximization problem)
  - Kendall-tau (τ) finds the total number of pairwise disagreements between A and B (minimization problem)

Input Rankers

A	B	C
o1	o2	o4
o2	o3	o2
o3	o1	o3
o4	o4	o1
o5	o6	o7

Aggregate

D
o2
o1
o3
o4
o5

- Precision of D with respect to A,B, and C

  $p(A,D) + p(B,D) + p(C,D) = 5 + 4 + 4 = 13$

- Kendall-tau of D with respect to A, B, and C

  $\tau(A,D) + \tau(B,D) + \tau(C,D) = 1 + 1 + 4 = 6$

- Missing values for τ are handled separately.

Aggregation Methods

- Cheap Methods:
  - Average (Av)
  - Median (Me)
  - Precision optimal (PrOPT)
- Methods that aim to optimize the Kendall-tau error of the aggregate with respect to the input rankers
  - Markov chain methods (Pagerank, Pg)
  - Iterative methods that improve a given aggregate methods
    - adjacent pairs (ADJ)
    - iterative best flip (IBF)

Precision Optimal

- Rank objects with respect to the number of times they appear in all the lists
  - Break ties with respect to their ranking in average rankers
  - Break remaining ties randomly

Input Rankers

A	B	C
o1	o2	o4
o2	o3	o2
o3	o1	o5
o4	o4	o1
o5	o6	o7

Number of times each object appears

D
o1, o2, o4
o3, o5
o6, o7

Break ties

D
o2
o1
o4
o3
o5
o6
o7

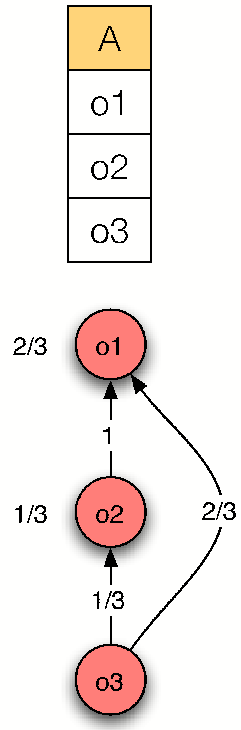
Choose top K

D
o2
o1
o4
o3
o5

Pagerank

- Construct a graph from rankings (similar to Dwork et. al. WWW2001)
  - Each object returned in a ranked list is a vertex
  - Insert an edge (i,j) for each ranked list where i is ranked higher than j
- Compute the pagerank [Brin & Page, WWW 1998] on this graph
  - The edges are weighted ($w_{j,i}$) proportional to the difference in rank it represents
  - The navigation probability is proportional to the edge weights
  - The random jump probability (p_i) is proportional to the indegree of each node
  - Alpha (α) is set to 0.85.
  - The pagerank Pg_i is the solution to the equations below:

$$Pg_i = \alpha p_i + (1-\alpha) \sum_{(j,i)\in E} Pg_j w_{j,i}$$



Iterative Improvement Methods

- Adjacent Pairs (ADJ)
  - Given an aggregate ranking, flip adjacent pairs until the total error with respect to the input rankers is reduced -> normally the Kendall-tau error metric is used [Dwork]
- Iterative Best Flip (IBF)
  - Given an aggregate ranking

~ While not done
  ~ For each object
    ~ record the current configuration
    ~ find the best flip among all other objects and do this flip even if it increases the error temporarily and make this the current configuration
  ~ Choose the lowest error configuration from the history
  ~ If the overall error is lower or if this is a configuration not seen before, then make this the current configuration
  ~ Else break ;

Iterative Best Flip

Input Rankers

A	B	C
o1	o5	o1
o2	o2	o4
o3	o3	o2
o4	o4	o3
o5	o1	o5

Aggregate

D
o5
o1
o2
o4
o3

Error
$\tau = 14$

After best flip for o5

D
o1
o5
o2
o4
o3

Error
$\tau = 13$

After best flip for o1

D
o2
o5
o1
o4
o3

Error
$\tau = 14$

After best flip for o2

D
o5
o2
o1
o4
o3

Error
$\tau = 13$

After best flip for o4

D
o4
o2
o1
o5
o3

Error
$\tau = 12$

After best flip for o3

D
o4
o2
o1
o3
o5

Error
$\tau = 11$

Choose the minimum error configuration from this run!
and continue

IBF seems to outperform ADJ and do well even when we start from a random ranking.

Analysis of Aggregation Methods

- Complex aggregators incorporate little nuances about the input rankers.They use more information but are sensitive to noise.
- Simple aggregators disregard information contained in the input rankers but are less sensitive to noise.
  - For example average is more complex than median and precision optimal
  - How about pagerank and other Kendall-tau optimization based optimizers?

Input Rankers

A	B
o1	o3
o2	o1
o3	o2

Kendall-tau optimal aggregations

D1	D2	D3
o3	o1	o1
o1	o2	o3
o2	o3	o2

The question we would like to answer is which aggregator performs well under different conditions. Does reducing Kendall-tau with respect to Kendall-tau always lead to a good solution?

Statistical Model of Aggregators

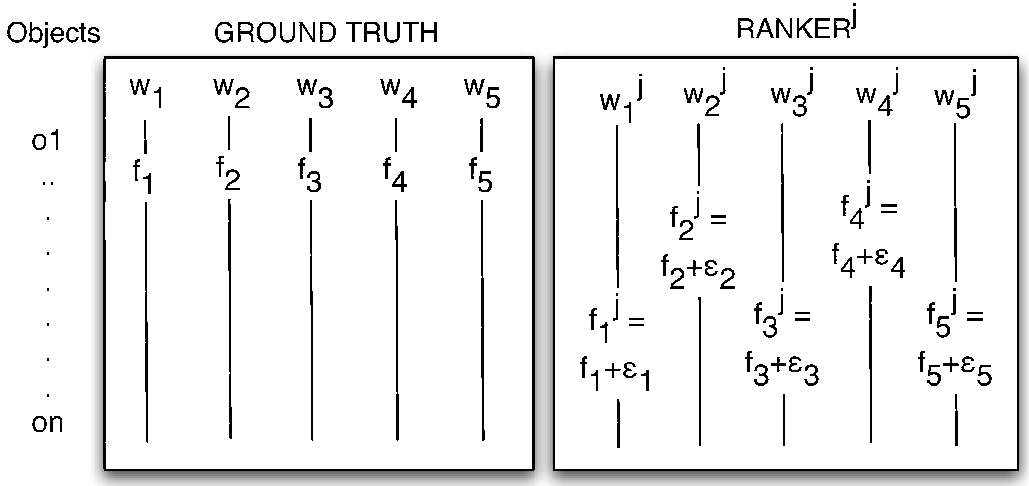
- Suppose there is a correct ranked list called the ground truth that represents the correct ordering.
- The correct ordering is computed for each object using:
  - A set of factors that measure the fit of object for a specific criteria (factors **F =f_1 ... f_F** where **f_l in [-3,3]**)
    - Examples of factors are number of occurrences of a keyword, recency of updates to a document or pagerank
  - A weight for each factor (**W= w_1,..., w_F** where **w_1 + ... + w_F = 1**)
  - The final score of each object **o_i** is computed using a linear combination function

$$V_i = \sum_{l=1}^{F} w_l f_l(o_i)$$

  - Objects are ranked with respect to the scores.

Statistical Model of Aggregators

- Each ranker produces a ranked list by using the same formula and the same factors
  - The ranker **j** tries to estimate the factors' true values for each object, produces $\mathbf{F^j}$
  - It also guesses the correct weights for the combination formula, produces $\mathbf{W^j}$



$$V_i = \sum_{l=1}^{F} w_l . f_l(o_i)$$

$$V_i^j = \sum_{l=1}^{F} w_l^j . f_l^j(o_i)$$

Statistical Model of Aggregators

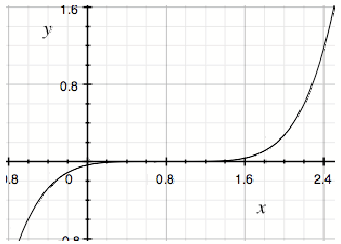
- The ranker's estimate $\mathbf{F^j}$ of a factor introduces an error $\boldsymbol{\epsilon}$, i.e. $\mathbf{F^j = F + \epsilon^j}$
  - The magnitude of error depends on a variance parameter σ^2
- The distribution of the error can be adjusted to model different types of spam

$$Var(\epsilon_{il}^{j}) = \sigma^2 \frac{(\gamma - f_l(o_i))^{\delta}.(\gamma + f_l(o_i))^{\beta}}{max_{f \in [-3,3]}(\gamma - f)^{\delta}.(\gamma + f)^{\beta}}$$

- In our model, we can model various types of correlation between the factors and the errors, but we do not report on those.

Test Setup

- We distribute the scores for each factor uniformly for 100 objects, use 5 factors and 5 rankers
- We set **γ= 1, δ = 5, β = 0.01** which models a case where rankers make small mistakes for “good” objects and make increasingly larger mistakes for “bad” objects
- We vary **σ^2** from **0.1, 1, 5** to **7**
- We set the ground truth weights to **W**



$$W = \langle \frac{1}{15}, \frac{2}{15}, \frac{3}{15}, \frac{4}{15}, \frac{5}{15} \rangle$$

- We assign 1,2,3,4, and 5 rankers to correct weights (**W**) and the remaining rankers are assigned the incorrect weights (**W^r**) (**n_{MI}** represent the number of rankers with the wrong weights)

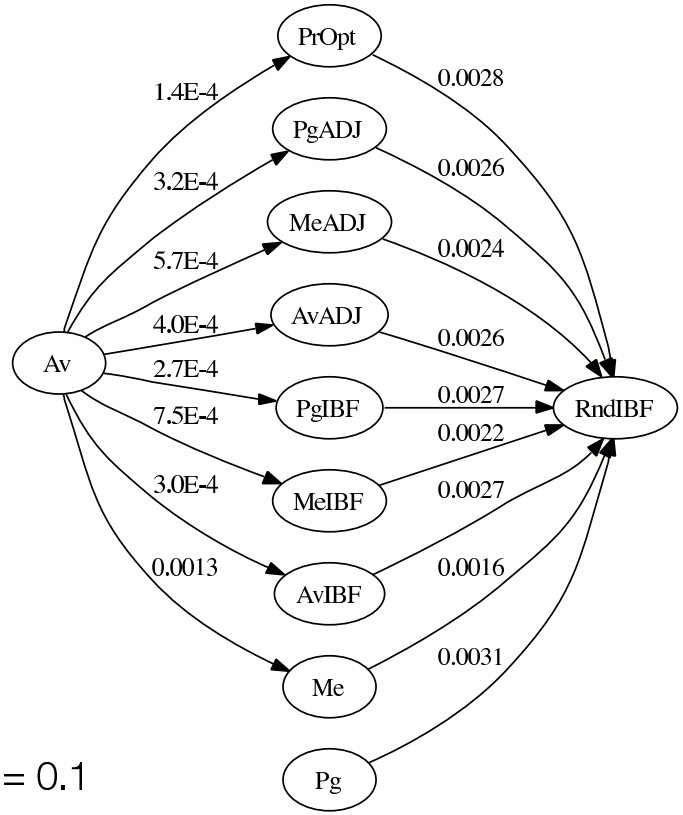
$$W^r = \langle \frac{5}{15}, \frac{4}{15}, \frac{3}{15}, \frac{2}{15}, \frac{1}{15} \rangle$$

Test Setup

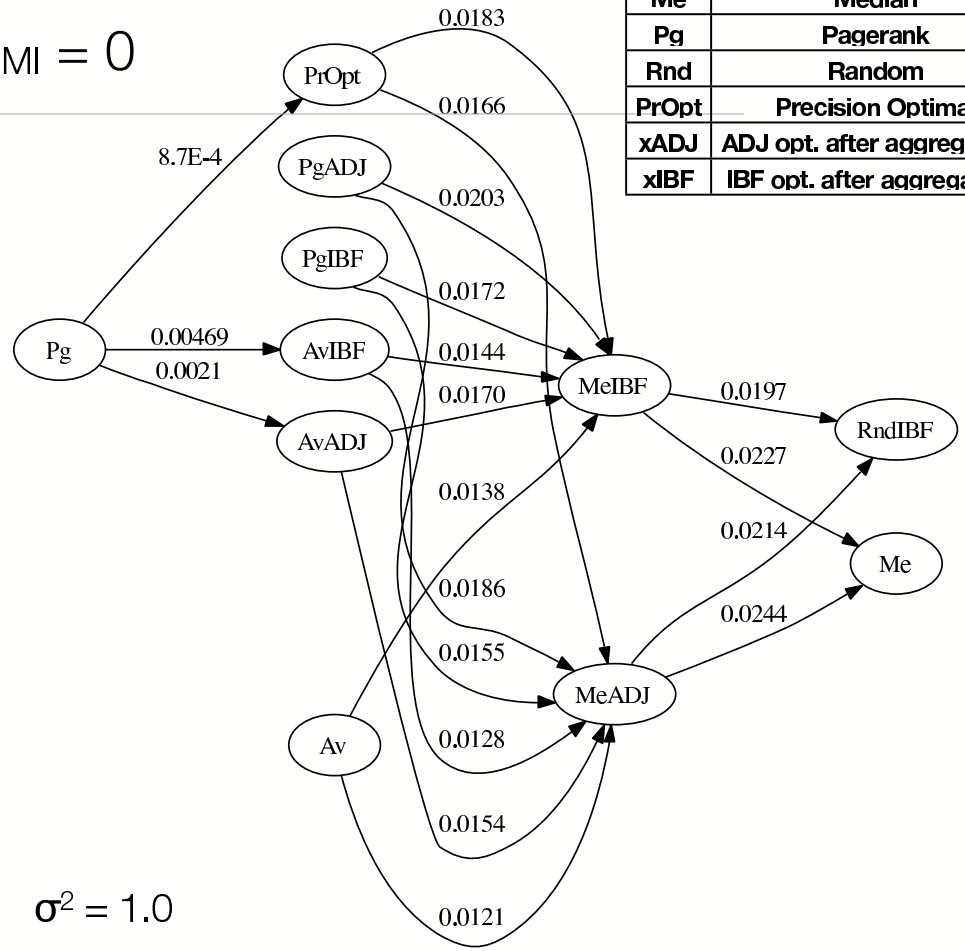
- For each setting, we construct 40,000 different data sets
  - For each dataset, we construct each aggregator for top 10 from the input rankers and output top 10
    - Compare the performance of each aggregator with respect to the ground truth using precision and Kendall-tau
    - For each error metric, we compute the difference between all pairs of aggregators
  - For each test case and error metric, we output for all pairs of aggreagators [A1, A2] a range [l, h] with 99.9% confidence
    - We assume A1 and A2 are roughly equivalent (A1 ≡ A2) if the range [l,h] crosses zero
    - Otherwise, we construct an ordering A1 > A2 or A2 < A1 based on the range and the error metric
    - We order the aggregators using topological sort based on this ordering for each test and each error metric

Results, precision for $n_{MI} = 0$

Legend	
Av	Average
Me	Median
Pg	Pagerank
Rnd	Random
PrOpt	Precision Optimal
xADJ	ADJ opt. after aggregator x
xIBF	IBF opt. after aggregator x



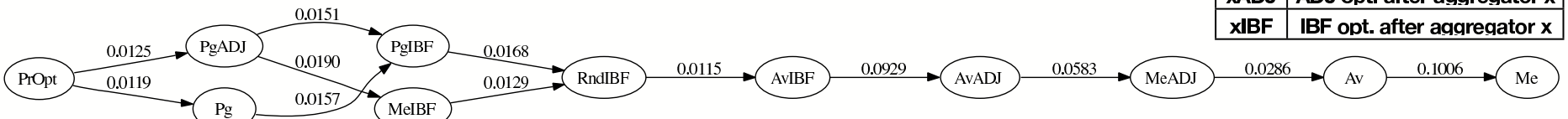
$\sigma^2 = 0.1$



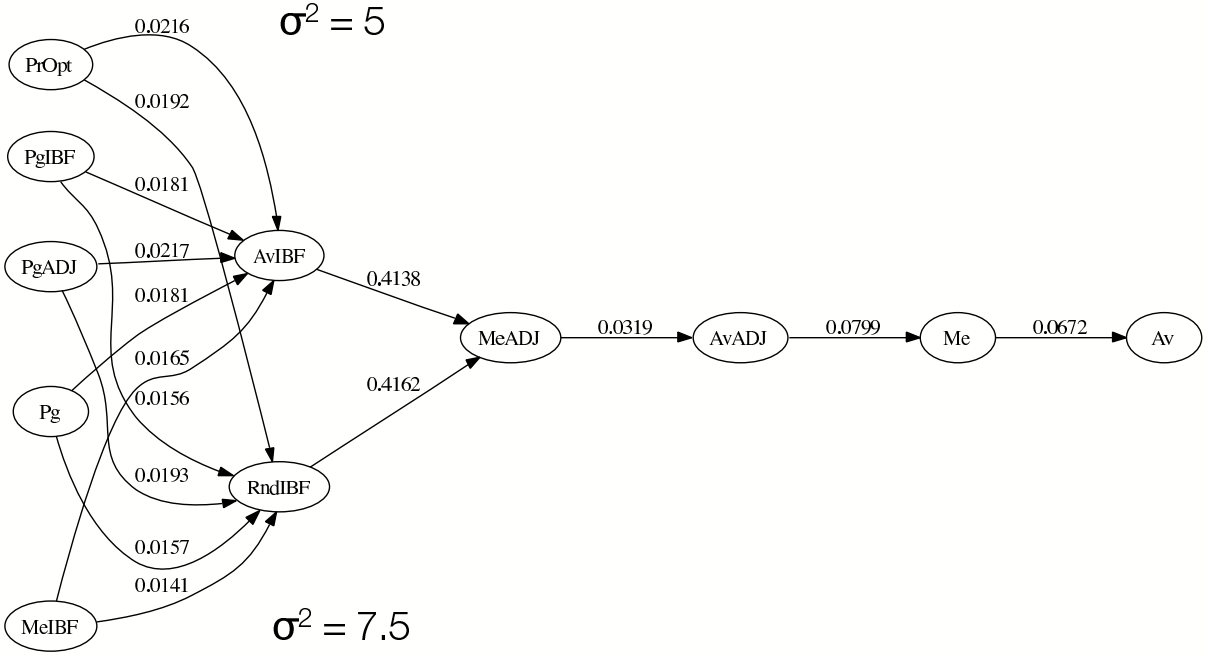
$\sigma^2 = 1.0$

Results, precision for $n_{MI} = 0$

Legend	
Av	Average
Me	Median
Pg	Pagerank
Rnd	Random
PrOpt	Precision Optimal
xADJ	ADJ opt. after aggregator x
xIBF	IBF opt. after aggregator x



$\sigma^2 = 5$

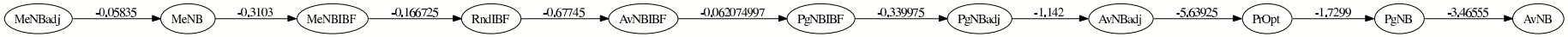


$\sigma^2 = 7.5$

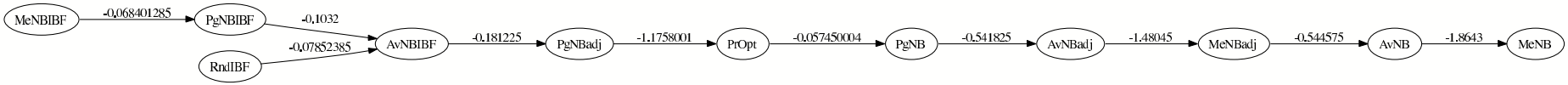
Kendall-tau results for $n_{MI} = 2$

Legend	
Av	**Average**
Me	**Median**
Pg	**Pagerank**
Rnd	**Random**
PrOpt	**Precision Optimal**
xADJ	**ADJ opt. after aggregator x**
xIBF	**IBF opt. after aggregator x**

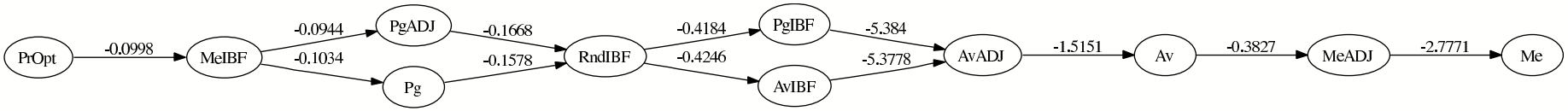
$\sigma^2 = 0.1$



$\sigma^2 = 1$



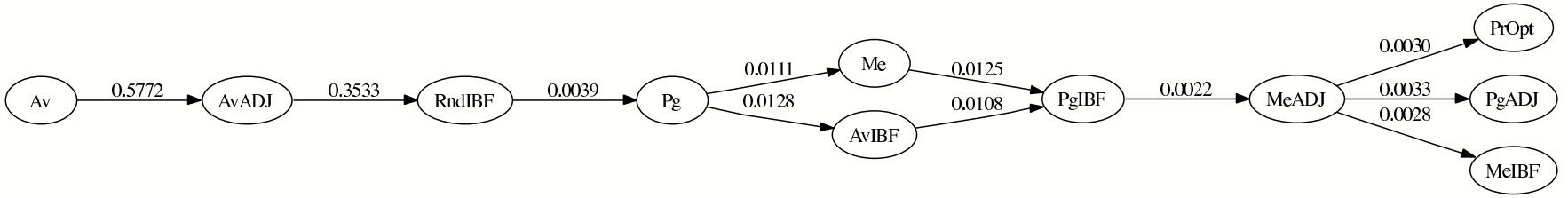
$\sigma^2 = 5$



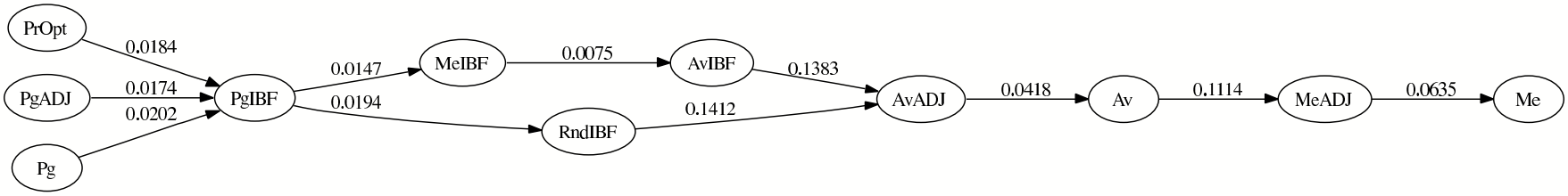
$\sigma^2 = 7.5$

Precision results for $n_{MI} = 4$

Legend	
Av	Average
Me	Median
Pg	Pagerank
Rnd	Random
PrOpt	Precision Optimal
xADJ	ADJ opt. after aggregator x
xIBF	IBF opt. after aggregator x



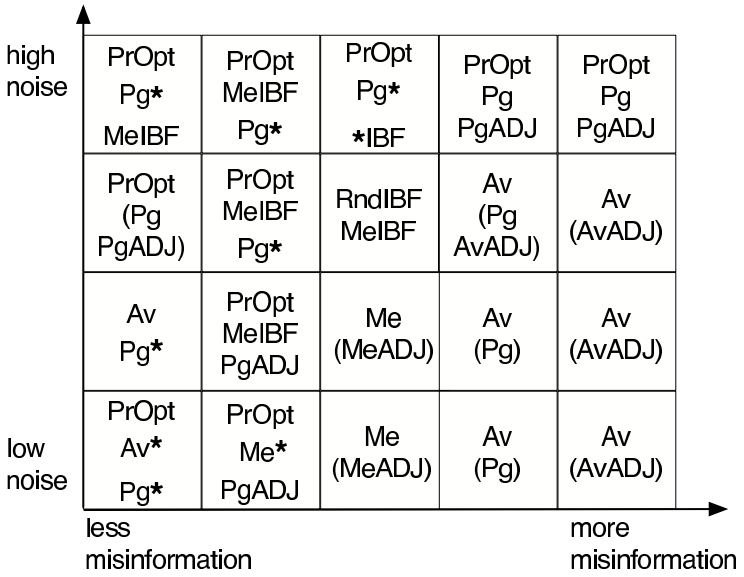
$\sigma^2 = 0.1$



$\sigma^2 = 7.5$

Result Summary

- Low noise:
  - Average is best when all the rankers are the same
  - Median is best when there is asymmetry among the rankers
- High noise
  - Robustness is needed, PrOpt, IBF and Pg are the best
  - As misinformation increases, robust but more complex rankers tend to do better



Conclusion and Future Work

- Two new aggregation methods, PrOPT and IBF that seem to do well in many cases, IBF seems to do well even starting from a random ranking
- No single rank aggregation method is best, there is a trade-off between information and robustness
- Further evaluation of rank aggregation methods is needed
  - Testing with various correlations both positive and negative between ranking factors and the errors made on these
  - Testing of the model with negative weights where misinformation is more misleading