PERFORMANCE EVALUATION OF PAGE RANK AGGREGATION ALGORITHMS

Priyanka[1], Mohinder Singh[2]
[1]M.Tech Student,
[2]Associate Professor,
Department of Computer Science & Engineering,
Maharishi Ved Vyas College of Engineering, Jagadhri, Yamunanagar

ABSTRACT The annoyance of combining the ranked possibilities of many experts is an antique and particularly deep hassle that has won renewed importance in many machine getting to know, statistics mining, and information retrieval applications. Powerful rank aggregation turns into hard in actual-international situations in which the ratings are noisy, incomplete, or maybe disjoint. We cope with those difficulties by extending numerous standard methods of rank aggregation to do not forget similarity between gadgets within the diverse ranked Lists, further to their ratings. The intuition is that comparable items must obtain similar scores, given the right degree of similarity for the domain of hobby.

Keywords: Rank Aggregation, Particle Swarm Optimization, Genetic Algorithm, Robust Rank Aggregation

1. INTRODUCTION

Rank aggregation is a vital approach for aggregating the options of a couple of retailers. The purpose of rating aggregation is to summarize a set of scores over a hard and fast of options by way of a single rating.

It is not simply the range of page links that determines the score, however additionally the quality. If a page hyperlinks to other web page that is exceptionally ranked it should receive precedence.

1.1 PageRank Algorithm

At each step in the PageRank algorithm, the score of each page is updated according to,

$$r = (1-P)/n + P*(A'*(r./d) + s/n)$$

r is a vector of PageRank scores.

P is a scalar damping factor (usually 0.85), which is the probability that a random surfer clicks on a link on the current page, instead of continuing on another random page.

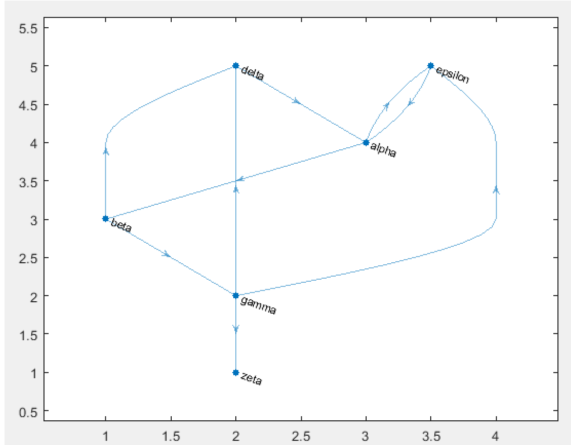
A' is the transpose of the adjacency matrix of the graph.

d is a vector containing the out-degree of each node in the graph. d is set to 1 for nodes with no outgoing edges.

n is the scalar number of nodes in the graph.

s is the scalar sum of the PageRank scores for pages with no links.

The rank of each page is largely based on the ranks of the pages that link to it. The term A'*(r./d) picks out the scores of the source nodes that link to each node in the graph, and the scores are normalized by the total number of outbound links of those source nodes. This ensures that the sum of the PageRank scores is always 1. For example, if node 2 links to nodes 1, 3, and 4, then it transfers 1/3 of its PageRank score to each of those nodes during each iteration of the algorithm.



Simulation studies of methods include: two non-optimization-based methods, mean and median from Borda's collection; two distribution-based methods, RRA and Stuart; and one optimization method GA

2. RELATED WORK

M. M. Sufyan Beg et al. [1] This NP-hard nature of (PFOA) partial foot rule ideal aggregation problem rouses to apply (GA) genetic algorithm for the PFOA issue. The GA based method may take long to figure, creator propose to settle on the number of ages of GA in view of as far as possible allowed by the client, Moreover, the inherent parallelism of GA is additionally used to accelerate the processing. Author achieve hybrid via crossover by carrying out multiplication of permutations. For transformation, the to-be-changed digit is traded with some other randomly selected digit in stage. Experimental procedure falls in accordance with the ones found in literature. Rank aggregation utilizing genetic algorithm are much better, when contrasted with the ones got utilizing the traditional Borda's technique for rank aggregation. **D. Sculley et al. [2]** propose a few algorithms for consolidating ranked lists of things with characterized comparability. Creator builds up assessment criteria for these algorithms by broadening past meanings of distance between ranked lists to incorporate the part of similitude between items. At last, creator tests these new techniques on both synthetic and real-world information, including information from an application in keywords extension for supported search advertisers. The outcomes demonstrate that incorporating similarity knowledge within rank aggregation can essentially enhance the performance of a few standard rank aggregation techniques, especially when utilized with noisy, inadequate, or disjoint rankings. **Pierre B. Borckmans et al. [3]** interested in finding the best low multilinear rank guess of a given tensor. This issue has been defined as an optimization issue over the Grassmann complex and it has been demonstrated that the objective function exhibits numerous minima. With a specific end goal to research the landscape of this cost work; writer proposes an adjustment of the Particle Swarm Optimization calculation (PSO). The Guaranteed Convergence PSO, proposed by van den Bergh, is adjusted, including a gradient component, in order to look for ideal arrangements over the Grassmann manifold. The tasks associated with the PSO algorithm are redefined using ideas of differential geometry. Creator shows some starter numerical experiments and examines the capacity of the proposed method to address the multimodal parts of the considered problem. **Lili Yan et al. [4]**Web web search tool has turned into a important tool for discovering data productively from the massive Web data. Based on Page Rank algorithm, a genetic PageRank algorithm (GPRA) is proposed. With the state of preserving PageRank algorithm points of interest, GPRA exploits genetic algorithm in order to solve web search. Experimental results have demonstrated that GPRA is better than PageRank algorithm and genetic algorithm on performance. **RaivoKolde et al. [5]** as a cure, creator proposes a novel robust rank aggregation (RRA) method. This technique recognizes qualities that are positioned reliably better than expected under invalid theory of uncorrelated data sources and allots a significance score for every quality. The fundamental probabilistic model makes the algorithm parameter free and robust to anomalies, clamor and errors. Noteworthiness scores likewise give a thorough method to keep only the statistically applicable genes in the final rundown. These properties make this approach robust and convincing for some settings. **GattacaLv et al. [6]** expand a dynamic programming algorithm initially for Kemeny scores. Creator additionally gives subtle elements on the execution of the algorithm. At long last, creator show comes about got from an experimental examination of this algorithm and two other well-known algorithms in light of genuine world and randomly generated issue occurrences. Test comes about demonstrate the usefulness and productivity of the algorithm in functional settings. **Ian Dewancker et al. [7]** propose a mechanism for looking at the execution of numerous improvement techniques for different performance metrics over a scope of optimization issues. Utilizing non-parametric factual tests to convert the measurements recorded for every issue into a partial ranking of optimization techniques, comes about from each issue are then amalgamated through a voting component to produce a final score for each optimization strategy. Mathematical investigation is given to motivate choices inside this strategy, and results comes about are given to exhibit the effect of certain ranking decisions. **Maunendra Sankar Desarkar et al. [8]** exhibit a non-regulated rank aggregation algorithm that is reasonable for metasearch and addresses the aspects specified previously. Creator likewise performs detailed test assessment of the proposed algorithm on four diverse bench-mark datasets having ground truth data. Aside from the unsupervised Kendall-Tau distance measure, a few directed assessment measures are utilized for execution correlation. Test comes about exhibit the adequacy of the proposed algorithm over benchmark strategies regarding managed evaluation metrics. Through these examinations author likewise demonstrate that Kendall-Tau remove metric may not be appropriate for assessing rank aggregation algorithms for metasearch. **Anna Korba et al. [9]** develops a statistical learning hypothesis for ranking aggregation in a general probabilistic setting (staying away from any rigid ranking model suppositions), assessing the generalization capacity of exact ranking medians. All inclusive rate limits are established and the circumstances where convergence occurs at an exponential rate are completely characterized. Minimax bring down limits are also proved, demonstrating that the rate limits got are ideal. **Xue Li et al. [10]** a methodical system is proposed to characterize diverse circumstances that may occur in view of the idea of separately positioned records. A complete recreation ponder is directed to look at the performance characteristics of a gathering of existing RA strategies that are reasonable for genomic applications under different settings simulated to mirror pragmatic circumstances. A non-little cell lung malignancy information case is accommodated encourage comparison. Based on our numerical outcomes, general rules about which strategies play out the best/most noticeably bad, and under what conditions, are gave. Likewise, creator examines key factors that generously influence the execution of the diverse strategies.

3. PROPOSED WORK

Rank aggregation is an essential approach for aggregating the preferences of multiple agents. The goal of ranking aggregation is to summarize a collection of rankings over a set of alternatives by a single ranking. Here, we focus on only the speed and accuracy which is sufficient to demonstrate the hierarchical nature of our ranking strategy. All rankers' data is provided in a single input file shown in figure:

- comma separated
- contain a header row
- first column is the object ids, assumed to be integers
- each column is a separate ranker
- if an object is not ranked by a ranker, leave that value empty.

The performance score is implemented as a score between 1 and -1. Simple aggregators pagerank and indegree are based on a graph representation of the ranks, where the weights from object i to j represents the number of rankers that rank object j higher than object i.

Iterative improvements algorithms are iterative greedy flip, igf, (flip a pair as long as improvements are made), iterative best flip, ibf, (flip a pair even when it does not improve for each possible pairs and try other greedy flips), and remove top k worst rankers, ir.

4. RESULTS AND DISCUSSION

The results show that it is not just the number of page links that determines the score, but also the quality. The alpha and gamma websites both have a total degree of 4, however alpha links to both epsilon and beta, which also are highly ranked. gamma is only linked to by one page, beta, which is in the middle of the list. Thus, alpha is scored higher than gamma by the algorithm.

Name	PageRank	InDegree	OutDegree
'http://www.example.com/alpha'	0.32098	2	2
'http://www.example.com/beta'	0.17057	1	2
'http://www.example.com/gamma'	0.10657	1	3
'http://www.example.com/delta'	0.13678	2	1
'http://www.example.com/epsilon'	0.20078	2	1
'http://www.example.com/zeta'	0.06432	1	0

```
data.csv
objects,ranker1,ranker2,ranker3,ranker4,ranker5
1,1,3,5,4,2
2,3,2,1,4,5
3,1,2,5,4,3
4,4,2,1,3,5
5,1,5,2,3,4
```

Figure 4.1: Rankers' Data

```
Indegree algorithm, score: 0.06
Iterative greedy flip with k = 1 score: 0.06
Final score: 0.06
Final ranker:
5 4 3 1 2
```

Figure 4.2: in:indegree igf: iterative greedy flip

```
Indegree algorithm, score: 0.06
Iterative best flip, score: 0.06
Final score: 0.06
Final ranker:
5 4 3 1 2
```

Figure 4.3: in:indegree ibf: iterative best flip (at most k rounds)

```
Indegree algorithm, score: 0.06
Iterative best removal with k = 1 score: 0.2
Final score: 0.2
Final ranker:
5 4 1 3 2
```

Figure 4.4: in:indegree ir: k-iterative remove up to k rankers

```
Pagerank algorithm, alpha = 0.85 , score: 0.06
Iterative greedy flip with k = 1 score: 0.06
Final score: 0.06
Final ranker:
5 4 1 3 2
```

Figure 4.5: pg: pagerank with given alpha (float between 0-1) igf: iterative greedy flip

```
Pagerank algorithm, alpha = 0.85 , score: 0.06
Iterative best flip, score: 0.06
Final score: 0.06
Final ranker:
5 4 1 3 2
```

Figure 4.6: pg: pagerank with given alpha (float between 0-1) ibf: iterative best flip (at most k rounds)

```
Pagerank algorithm, alpha = 0.85 , score: 0.06
Iterative best removal with k = 1 score: 0.2
Final score: 0.2
Final ranker:
5 4 1 3 2
```

Figure 4.7: pg: pagerank with given alpha (float between 0-1) ir: c

```
Random rank algorithm with k = 5
Iterative greedy flip with k = 1 score: 0.06
Final score: 0.06
Final ranker:
5 3 4 2 1
```

Figure 4.8: rnd: k-random with k tries igf: iterative greedy flip

```
Random rank algorithm with k = 5
Iterative best flip, score: 0.06
Final score: 0.06
Final ranker:
4 3 5 2 1
```

Figure 4.9: rnd: k-random with k tries ibf: iterative best flip (at most k rounds)

```
Random rank algorithm with k = 5
Iterative best removal with k = 1 score: 0.2
Final score: 0.2
Final ranker:
5 4 1 3 2
```

Figure 4.10: rnd: k-random with k tries ir: iterative best flip (at most k rounds)

Table 4.1:

Aggregator list	Iterative algorithms
in: indegree	igf: iterative greedy flip
pg alpha-pagerank with given alpha (float between 0-1)	ibf k-iterative best flip (at most k rounds
rnd k-random with k tries	ir k-iterative remove up to k rankers

Aggregator list	in		Pg		rnd	
Iterative algorithms	Final Score	Final Ranker	Final Score	Final Ranker	Final Score	Final Ranker
Igf	0.06	5,4,3,1,2	0.06	5,4,1,3,2	0.06	5,3,4,2,1
Ibk	0.06	5,4,3,1,2	0.06	5,4,1,3,2	0.06	4,3,5,2,1
Ir	0.2	5,4,1,3,2	0.2	5,4,1,3,2	0.2	5,4,1,3,2

Importing the rankers data in Matlab workspace as shown in figure 2

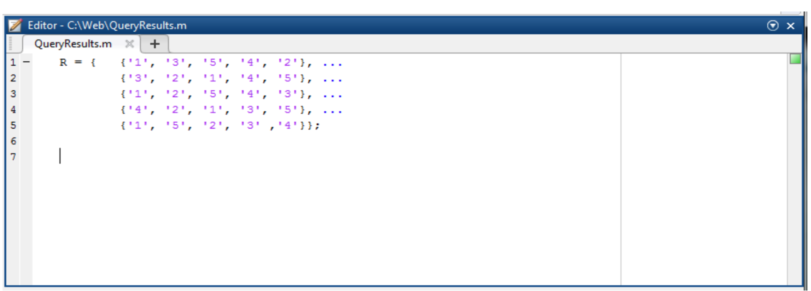


Figure 4.11: Query Results

```
[aggR] = aggregateRanks(R,5,'mean',1)
timeElapsed =
   0.0182
aggR =
   0.3600
   0.5600
   0.6400
   0.7200
   0.7200

 [aggR] = aggregateRanks(R,5,'stuart',1)
timeElapsed =
   0.4370
aggR =
   0.0163
   0.2150
   0.4531
   0.5299
   0.6662

 [aggR] = aggregateRanks(R,5,'RRA',1)
timeElapsed =
   0.0484
aggR =
   0.2579
   0.8518
   0.9957
   0.9962
   0.9968

 [aggR] = aggregateRanks(R,5,'median',1)
timeElapsed =
   1.5647
aggR =
   0.2000
   0.4000
   0.8000
   0.8000
   0.6000

 [aggR] = aggregateRanks(R,5,'ga',1)
timeElapsed =
   0.4594
aggR =
   0.3104
   0.5210
   0.5519
   0.6340
   0.6787
```

Table 4.2: Timing Analysis of various Methods

Method	**Time (seconds)**
Mean	0.0182
Stuart	0.4370
RRA	0.0484
GA	0.4594
Median	1.5647

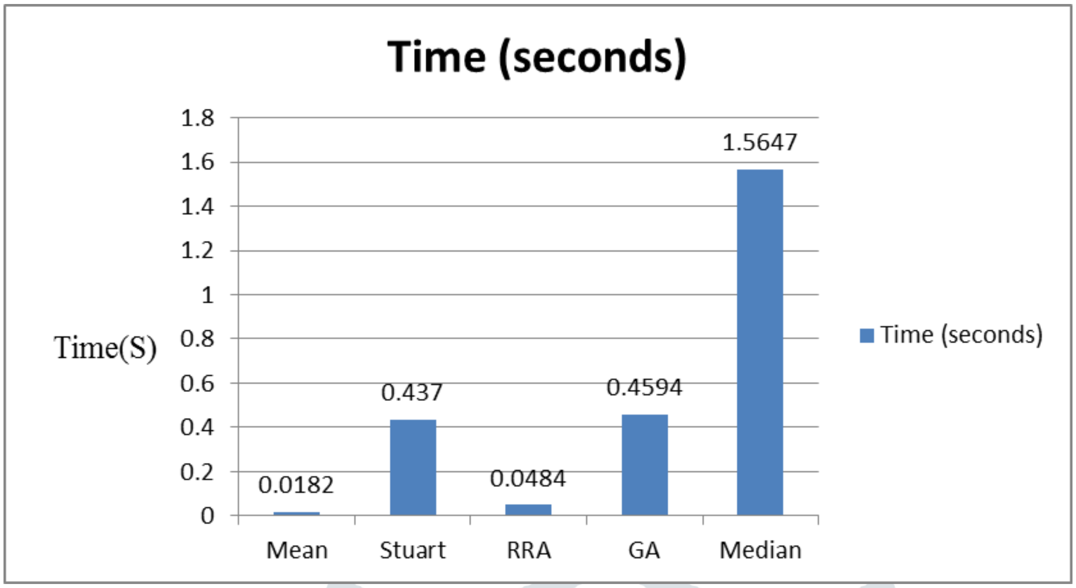


Figure 4.12 Computation Time

5. CONCLUSION

Rank aggregation is an essential approach for aggregating the preferences of multiple agents. The goal of ranking aggregation is to summarize a collection of rankings over a set of alternatives by a single ranking. Rank aggregation is an essential approach for aggregating the preferences of multiple agents. The goal of ranking aggregation is to summarize a collection of rankings over a set of alternatives by a single ranking.

REFERENCES

[1] M. M. Sufyan Beg, ”Parallel Rank Aggregation for the World Wide Web”, IEEE, 2004, pp.385-390.

[2] D. Sculley, ”Rank Aggregation for Similar Items”, Work performed at Yahoo!, Inc., in Spring of 2006, pp.1-12.

[3] Pierre B. Borckmans, MariyaIshteva, and Pierre-Antoine Absil, ”A Modified Particle Swarm Optimization Algorithm for the Best Low Multilinear Rank Approximation of Higher-Order Tensors”, Universit´ecatholique de Louvain, Louvain-la-Neuve, Belgium, 2010, pp.15-23.

[4] Lili Yana, ZhanjiGuia, WencaiDub,QingjuGuo, ”An Improved PageRank Method based on Genetic Algorithm for Web Search”, Procedia Engineering 15 ,2011, pp. 2983 – 2987

[5] RaivoKolde, Sven Laur, Priit Adler and JaakVilo, ”Robust rank aggregation for gene list integration and meta-analysis”, BIOINFORMATICS, Vol. 28 no. 4 2012, pp.573–580

[6] GattacaLv, ”An Analysis of Rank Aggregation Algorithms”, arXiv:1402.5259v5 [cs.DS] 4 May 2014, pp.1-12.

[7] Ian Dewancker ,Michael McCourt ,Scott Clark ,Patrick Hayes ,Alexandra Johnson ,George Ke, ”A Strategy for Ranking Optimization Methods using Multiple Criteria”, JMLR: Workshop and Conference Proceedings 64:pp.11–20, 2016.

[8]Maunendra Sankar Desarkar, SudeshnaSarkar,Pabitra Mitra, ”Preference relations based unsupervised rankaggregation for metasearch”,Expert Systems With Applications 49 ,2016,pp.86–98

[9] Anna Korba,Stephan Clemencon,Eric Sibony, ”A Learning Theory of Ranking Aggregation”, International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54, pp.1-10.

[10] Xue Li, Xinlei Wang, and Guanghua Xiao, ”A comparative study of rank aggregation methods for partial and top ranked lists in genomic applications”, Briefings in Bioinformatics, 2017, pp.1–12

[11] E. Dopazo, M.L. Martínez-Céspedes, "Rank aggregation methods dealing with incomplete information applied to Smart Cities", 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)

[12] Ronaldo C. Prati, "Combining feature ranking algorithms through rank aggregation", The 2012 International Joint Conference on Neural Networks (IJCNN)

[13] J. M. Maestre, Hideaki Ishii, "Node Aggregation for Enhancing PageRank", IEEE Access (Volume: 5)

[14] Randall Wald, "An extensive comparison of feature ranking aggregation techniques in bioinformatics", 2012 IEEE 13th International Conference on Information Reuse & Integration (IRI)

[15] Samuel Oliveira, "Evolutionary rank aggregation for recommender systems", 2016 IEEE Congress on Evolutionary Computation (CEC)

[16] Yan Pan, Hanjiang Lai, "Rank Aggregation via Low-Rank and Structured-Sparse Decomposition", Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013