DeepAggregation: A New Approach for Aggregating Incomplete Ranked Lists using Multi-Layer Graph Embedding

Rohith Dwarakanath Vallam
IBM Research - India
rovallam@in.ibm.com

Ramasuri Narayanam
IBM Research - India
ramasurn@in.ibm.com

Srikanth G. Tamilselvam
IBM Research - India
srikanth.tamilselvam@in.ibm.com

Nicholas Mattei
Tulane University
New Orleans, LA 70118
nsmattei@tulane.edu

Sudhanshu S. Singh
IBM Research - India
sudsing3@in.ibm.com

Shweta Garg
IBM Research - India
shgarg87@in.ibm.com

Gyana R. Parija
IBM Research - India
gyana.parija@in.ibm.com

ABSTRACT

Preference aggregation, and specifically rank aggregation, is a well known problem in the fields of computational social choice and preference handling with broad application including web search and recommendation systems. Inspired by the recent advances in the area of deep neural representation learning, for the first time in the literature, in this paper we leverage unsupervised deep learning techniques - especially graph embeddings - for aggregating a collection of incomplete rank lists and accordingly we develop an algorithm called *DeepAggregation*. It takes as input a set of incomplete rank lists and constructs a multi-layer graph wherein the nodes are the alternatives that are ranked and the edges capture information contained in the incomplete rank lists. We then compute deep neural representation vectors (i.e. embeddings) for the nodes and then derive the aggregated order using these representation vectors. Our proposed algorithm can handle incomplete rank lists with or without ties. We conduct thorough empirical analysis of the proposed DeepAggregation algorithm using various real life data sets such as TripAdvisor reviews data. We empirically observe that DeepAggregation generates impressive results in comparison with a number of well-known state-of-the-art preference aggregation methods.

KEYWORDS

Incomplete rank lists, Top-*k* rank lists, Partial rank lists, Rank aggregation, multi-layer graphs, deep neural networks, graph representation learning, graph embedding

ACM Reference Format:
Rohith Dwarakanath Vallam, Ramasuri Narayanam, Srikanth G. Tamilselvam, Nicholas Mattei, Sudhanshu S. Singh, Shweta Garg, and Gyana R. Parija. 2019. DeepAggregation: A New Approach for Aggregating Incomplete Ranked Lists using Multi-Layer Graph Embedding. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*, IFAAMAS, 3 pages.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

1 INTRODUCTION

Preference aggregation, in particular *Rank aggregation*, problem deals with finding a permutation (or ranking) of a set A of n alternatives/items by combining information from a collection of input rankings over the alternatives in A [4, 5, 10]. This problem has a rich history in the fields of information retrieval, meta-search, social choice theory, e-commerce, committee decision making, and many other fields [2, 7, 9]. There are multiple variants of rank aggregation problem in the literature such as aggregating full rank lists, partial rank lists, top-k lists [1, 14] and other models of missing or incomplete information.

There exist significant amount of work in economics and computer science in the literature that builds the formal mathematical basis for aggregation of rankings. This existing work can be largely divided into two categories: axiomatic approach for aggregation and metric approach for aggregation [14]. In the axiomatic approach, a set of axioms are formulated to essentially characterize the underlying aggregation functions [4]. Following the metric approach, one defines a metric on the given (partial, full, or top-k) and then computes aggregated rank order as a consensus ranking whose total distance to the given rankings is minimized [5, 14]. Towards addressing the rank aggregation problem, one of the fundamental aspects is to define a measure of distance among input rank orders. There exists significant amount of work on designing metrics for full as well as partial rank lists [10, 14].

Though there exist a number of well known schemes for full rank list aggregation [4], often it is not possible to rank all the alternatives in A in several real life scenarios, such as:

- In the web search engine setting, there would be several thousands of web pages (i.e. the set A) that are relevant for a given query. However, the search engine would only rank the top-k (say top 100 or 200) web pages; and
- Customers often provide reviews about a hotel where they stay or a restaurant where they dine. In any of these scenarios, in the review, the customer usually highlights the items or facilities that she/he likes or dislikes. That is, customers generally don't enumerate all the facilities or items that the hotel/restaurant provides while rating/ranking.

Algorithm	MC1	MC2	MC3	MC4	MCS1	MCS2	MCS3	MCS4	Approx	Borda	DA_DW	DA_NV	DA_Avg
DS1	41	43	42	39	37.5	42	42.5	39	36.125	38.5	37	35.5	**32**
DS2	33	36	33	**31.5**	33.5	34	33	**31.5**	32.96	33.5	42	36	43
DS3	44.5	**39**	44.5	43	44	42	44.5	40	39.115	41	39.5	46.5	45.5
DS4	40.5	40.5	40.5	40.5	46	40.5	40.5	40.5	39.655	36.5	**32.5**	39	40
DS5	39.5	41.5	39.5	40	42	38.5	40	37.5	37.54	**37**	37.5	46.5	47.5
DS6	43	39.5	43	40	38	39.5	44	36.5	37.5	**34**	43.5	40.5	36.5
DS7	44	39	39	40	40	39	41.5	40	31.43	33.5	**31**	35.5	35
DS8	48	41.5	50	44.5	34	49	43	44.5	31.22	25.5	**24**	30	25
DS9	43	46	40.5	40.5	41	38	42	40.5	35.72	36	**34**	40	**34**
DS10	36.5	37.5	37.5	**34.5**	35.5	40	37.5	**34.5**	36.505	37	38.5	41	38

Table 1: Sum Distance Metric based performance comparison of the proposed DeepAggregation algorithm vis-a-vis the well known baselines using 10 datasets (labelled as DS1, DS2, ..., DS10)

As we can see from above, in many practical settings, we often have access to only incomplete (or partially) ranked lists and thus the task of generating aggregated rank ordering of alternatives or finding the top-k alternatives should only work with incomplete ranked lists [13]. Clearly it generalizes the setting of well known *full rank list aggregation*. Note that there exist a few efforts [1] in the literature to address the partial rank list aggregation problem. In this paper, given a set A of n alternatives and a collection of m incomplete ranked lists, we focus on the problem of finding the top-k alternatives according to a desired objective function or error metric.

2 PROPOSED APPROACH

To the best of our knowledge, for the first time in the literature, in this paper we leverage unsupervised deep learning techniques [3], which have proven successful in natural language processing and a number of other scenarios, into the topic of *incomplete (partially) ranked list aggregation* and accordingly develop an algorithm, which we refer to as *DeepAggregation*. In our approach, we consider a collection of incomplete rank lists and then construct a multi-layer graph wherein *nodes* are alternatives that are being ranked and *edges* capture the information contained in the given incomplete ranked lists. We next compute low dimensional deep neural representation vectors (also usually referred to as embeddings) for the nodes in the above constructed multi-layer graph such that the representations are latent features of the nodes to capture neighborhood similarity and community membership. We finally derive the aggregated preference ordering of the alternatives using these representation vectors. Figure 1 highlights the overall scheme of our proposed approach.

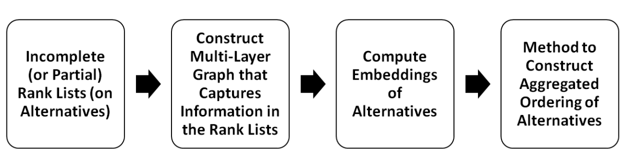

Figure 1: High level view of the proposed DeepAggregation approach

The following are a few key advantages of the proposed DeepAggregation scheme: (a) It can handle incomplete rank lists both with or without ties; and (b) Most of the state-of-the-art rank aggregation algorithms (for both full rank lists and incomplete rank lists) don't use outside heuristic information - such as similarity scores, reviews, and any other text in natural language - about alternatives. As opposed to this, our proposed DeepAggregation approach has the capability to efficiently leverage such natural language based auxiliary information about alternatives while computing the aggregated rank list.

3 EXPERIMENTAL RESULTS

We conduct thorough empirical analysis of the proposed DeepAggregation algorithm using several real life data sets. We also benchmark its performance using a number of well-known state-of-the-art preference aggregation methods as follows.

Data Sets: We consider top 10 hotels having most user reviews from TripAdvisor [11]. Each hotel results in an instance of our problem wherein *alternatives* are the facilities provided by the hotel and each incomplete rank list is generated from each type of the customer category.

Performance Measurement: Following this metric, we calculate the sum of distances from the aggregated rank list to the initial rank orders and it is inspired by the Kemeny optimal ordering principle [8]. That is, Sum Distance Metric = $\sum_{i \in N} d(r_a, r_i)$, where r_a is the aggregated rank list, for each $i \in N$, r_i is the i-th initial rank order, and $d(.,.)$ is a distance metric to measure the distance between any two bucket (rank) orders. Please refer to Section 3.1 in [14] for more details on the distance metric.

Baseline Algorithms: MC1, MC2, MC3, and MC4 algorithms from [5]; MCS1, MCS2, MCS3, and MCS4 algorithms from [15]; Approx [1]; and Borda [16].

Our Results: Using Metric 1, Table 1 shows the performance comparison of proposed DeepAggregation method vis-a-vis the well known baselines. Note that *DA_DW, DA_NV,* and *DA_Avg* refer to the proposed DeepAggregation approach with embeddings of alternatives derived using Deepwalk [12], Node2Vec [6], and averaging the embeddings by Deepwalk & Node2Vec respectively. Clearly the proposed DeepAggregation method outperforms the benchmarks in several data set instances.

REFERENCES

[1] Nir Ailon. 2010. Aggregation of Partial Rankings, p-Ratings and Top-m Lists. *Algorithmica* 57, 2 (2010), 284–300.

[2] Kenneth J. Arrow. 1951. *Social Choice and Individual Values.* Wiley, New York.

[3] Y. Bengio, A. Courville, and P. Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1798–1828.

[4] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia. 2016. *Handbook of Computational Social Choice.* Cambridge University Press, New York, USA.

[5] Moni Naor D. Sivakumar Cynthia Dwork, Ravi Kumar. 2001. Rank aggregation methods for the Web. In *Proceedings of WWW*. 613–622.

[6] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.

[7] C. A. Tovey J. Bartholdi and M. A. Trick. 1989. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare* 6 (1989), 157–165.

[8] John Kemeny. 1959. Mathematics without numbers. *Daedalus* 88 (1959), 577–591.

[9] Jon Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *J. ACM* 46, 5 (1999), 604–632.

[10] Ravi Kumar and Sergei Vassilvitskii. 2010. Generalized distances between rankings. In *Proceedings of WWW*. 571–580.

[11] Nicholas Mattei and Toby Walsh. [n. d.]. PrefLib: A Library for Preferences. ([n. d.]). http://www.preflib.org/

[12] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[13] A. Popova, M. Regenwetter, and N. Mattei. 2013. A Behavioral Perspective on Social Choice. *Annals of Mathematics and Artificial Intelligence* 68, 1–3 (2013), 135–160.

[14] Mohammad Mahdian D. Sivakumar Erik Vee Ronald Fagin, Ravi Kumar. 2006. Comparing Partial Rankings. *SIAM J. Discrete Math.* 20, 3 (2006), 628–648.

[15] D. Sculley. 2007. Rank Aggregation for Similar Items. In *Proceedings of the 7th SIAM International Conference on Data Mining (SDM)*. 587–592.

[16] H.P. Young. 1974. An axiomatization of Borda's rule. *Journal of Economic Theory* 9, 43 (1974), 43–52.