Expert Systems With Applications 49 (2016) 86–98

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Preference relations based unsupervised rank aggregation for metasearch

Maunendra Sankar Desarkar[1,*], Sudeshna Sarkar, Pabitra Mitra

Department of CSE, IIT Kharagpur, Kharagpur 721302, India

ARTICLE INFO

Keywords:
Rank aggregation
Metasearch
Information retrieval

ABSTRACT

Rank aggregation mechanisms have been used in solving problems from various domains such as bioinformatics, natural language processing, information retrieval, etc. Metasearch is one such application where a user gives a query to the metasearch engine, and the metasearch engine forwards the query to multiple individual search engines. Results or rankings returned by these individual search engines are combined using rank aggregation algorithms to produce the final result to be displayed to the user. We identify few aspects that should be kept in mind for designing any rank aggregation algorithm for metasearch. For example, generally equal importance is given to the input rankings while performing the aggregation. However, depending on the indexed set of web pages, features considered for ranking, ranking functions used etc. by the individual search engines, the individual rankings may be of different qualities. So, the aggregation algorithm should give more weight to the better rankings while giving less weight to others. Also, since the aggregation is performed when the user is waiting for response, the operations performed in the algorithm need to be light weight. Moreover, getting supervised data for rank aggregation problem is often difficult. In this paper, we present an unsupervised rank aggregation algorithm that is suitable for metasearch and addresses the aspects mentioned above.

We also perform detailed experimental evaluation of the proposed algorithm on four different benchmark datasets having ground truth information. Apart from the unsupervised Kendall-Tau distance measure, several supervised evaluation measures are used for performance comparison. Experimental results demonstrate the efficacy of the proposed algorithm over baseline methods in terms of supervised evaluation metrics. Through these experiments we also show that Kendall-Tau distance metric may not be suitable for evaluating rank aggregation algorithms for metasearch.

1. Introduction and motivation

The input to the rank aggregation problem is a number of rankings obtained from different sources. The sources can be human judges or algorithms. The task is to combine these input rankings and produce an aggregate ranking. Rank aggregation techniques have been used to solve problems from different applications domains. We mention here a few expert and intelligent systems applications or application domains where rank aggregation techniques have been used to arrive at solutions to different problems.

- *Voting:* For *voting* applications, there is a fixed list of candidates. There are voting schemes where the voters are allowed to rank the candidates based on the order of their choices. For such schemes, it is often necessary to combine the rankings provided by the voters to produce an aggregate ranking of the candidates, to obtain a *consensus ordering of the candidates*. This aggregate ranking can be determined by rank aggregation algorithms (de Borda, 1781; Davenport & Kalagnanam, 2004; Elkind & Lipmaa, 2005).
- *Metasearch:* Metasearch engines (e.g. MetaCrawler (http://www.metacrawler.com/), Dogpile (http://www.dogpile.com), Entireweb (http://www.entireweb.com/), etc.) accept queries from users, and forward that query to several *second-level* search engines. Each of these second-level search engines returns a ranked list of items for the query. The metasearch engine then combines these ranked lists to produce an aggregate list. This aggregate list is displayed to the user as a response

* Corresponding author. Tel.: +91 40 2301 8466.
E-mail addresses: maunendra@iith.ac.in (M.S. Desarkar), sudeshna@cse.iitkgp.ernet.in (S. Sarkar), pabitra@cse.iitkgp.ernet.in (P. Mitra).
[1] Present address: IIT Hyderabad, Kandi, Medak District 502 285, Telangana, India.

http://dx.doi.org/10.1016/j.eswa.2015.12.005
0957-4174/

to his/her query (Aslam & Montague, 2001; Chen, Wang, Song, & Zhang, 2008; Jansen, Spink, & Koshman, 2007; Thomas & Hawking, 2009). Thus rank aggregation is a central task for successful working of any metasearch engines.

- *Multi-criteria decision making:* There are systems where the objects (documents/products/candidates) might be scored or ranked based on multiple criteria. However, a single ordering of the objects is required as the final ranking. Rank aggregation algorithms are often used in tasks such as selecting product or services for recommendation (Shao, Chen, & Huang, 2010), combining feature based rankings for producing a single ranking for web search queries (Farah & Vanderpooten, 2007), candidate screening for hiring process in a large organization (Mehta, Pimplikar, Singh, Varshney, & Visweswariah, 2013), diversifying search results (Ozdemiray & Altingovde, 2015), etc.
- *Recommender systems:* Recommender systems have traditionally recommended items to individual users. Recently there has been a proliferation of recommender systems that recommend items to *groups of users* (Jameson & Smyth, 2007). Examples of such scenarios include a group of users listening to music, watching a movie, going to a restaurant or a museum etc. For recommending items to a group of users, Baltrunas, Makcinskas, and Ricci (2010) present a methodology where the system first gets ranked recommendation list for each member of the group, and then aggregates the individual lists to produce the recommendation for the group. Sohail, Siddiqui, and Ali (2015) aggregate user feedbacks to provide evaluation in product recommender systems.
- *Natural language processing:* For the language translation task, algorithms are suggested that for a source sentence, consider the ranked list of translations returned by different translator algorithms, and combine them to produce a final ranking of the candidate translations (Rosti et al., 2007). Similar techniques of combining ranked lists of candidate solutions for finding the final output are used for approaching the problems of syntactic dependency parsing (Sagae & Lavie, 2006) and word sense disambiguation (Brody, Navigli, & Lapata, 2006).
- *Networking:* In the networking domain, a number of metrics have been proposed to quantify the inherent robustness of a network topology against failures. However, each single metric usually offers only a limited view of network vulnerability. When applied to certain network configurations, different metrics rank the network topologies in different orders, and no single metric fully characterizes network robustness against different modes of failure. To overcome this problem, Yazdani and Leonardo Duenas-Osorio (2013) propose a multi-metric approach where the ordering of the topologies given by different individual metrics are combined to get an overall ranking of robustness for the network topologies. In social networking domain, Tabourier, Libert, and Lambiotte (2014) use rank aggregation for link prediction. Ordering of a user's neighbors according to various network-based features (such as Adamic-Adar, Jaccard Index, Katz measure, etc.) are identified. These orderings are aggregated to suggest set of possible new connections for the user.
- *Healthcare:* In Fields, Okudan, and Ashour (2013), the authors present a system where in an emergency department of a hospital, nurses provide orderings of the patients in terms of the severity of the patients' conditions. Nurses put patients requiring medical attention more urgently than others near the top of the list. Such orderings provided by multiple nurses are aggregated to produce a single ranking of patients and the patients are attended in that order.
- *Bioinformatics:* Rank aggregation algorithms are used in the bioinformatics domain also, for cluster validation in microarray data analysis (Pihur, Datta, & Datta, 2007), identifying differentially expressed genes (Fang, Feng, & Ng, 2011), high throughput screening in nanotoxicology (Patel et al., 2012), multimodal biometric systems (Monwar & Gavrilova, 2013), feature selection (Sarkar, Cooley, & Srivastava, 2014), etc.

It is evident from the discussion above that rank aggregation algorithms are used to solve problems in different expert and intelligent systems. Most of the rank aggregation algorithms discussed in literature are unsupervised in nature. This is because unsupervised methods can be easily ported across different applications. Supervised approaches to rank aggregation need supervised ranked data, which is expensive to acquire (Klementiev, Roth, & Small, 2008; Wu, Greene, & Cunningham, 2010). Therefore, unsupervised rank aggregation is an important problem to be studied. We wish to develop an unsupervised rank aggregation algorithm. Our main motivation is to work on the metasearch problem. In metasearch, ranked responses from different search engines are combined and the aggregate ranking is displayed as the output. As the aggregation is performed in runtime when the user is waiting for the final result for his query, it is essential for the algorithm to be of low complexity, and also the steps involved in performing the aggregation should involve low cost operations. Also, the quality of the input rankings given by individual search engines are affected by various factors such as indexed set of web-pages, features used for ranking, ranking function used etc. Due to this fact, qualities of the input rankings may not be equal. So, in metasearch, there is a need to identify the qualities of the input rankings and use that quality information while performing the aggregation. We have not come across any work in literature that emphasizes on *these requirements* while developing unsupervised algorithms for metasearch. The unsupervised algorithm proposed in this paper is developed keeping these aspects in mind.

Several unsupervised rank aggregation algorithms and their analysis are presented in the works (Aslam & Montague, 2001; Betzler, Bredereck, & Niedermeier, 2014; de Borda, 1781; Dwork, Kumar, Naor, & Sivakumar, 2001; Schalekamp & van Zuylen, 2009). These algorithms are widely used in metasearch engines and in metasearch literature. However, all these methods consider the input rankings to be equally good. Equal importance is given to the rankers for computing the aggregate ranking. Cohen, Schapire, and Singer (1998) learn quality weights of the rankers and uses that information for aggregation. The algorithm maintains a single query vector over the rankers at any time, which is irrespective of the query. However, relative qualities of the rankers can be different for different input queries. This is often true for metasearch, due to different sets of indexed pages maintained by different search engines. Also, the method learns the quality weights from user feedback data, which as mentioned earlier, may be difficult to get. Several other recent researches also use supervised approaches for rank aggregation (Liu, Liu, Qin, Ma, & Li, 2007; Pujari & Kanawati, 2012; Tabourier et al., 2014). The work in Rajkumar and Agarwal (2014) discusses desirable theoretical properties for rank aggregation algorithms and provides a supervised algorithm for rank aggregation.

The method proposed in this work considers each input ranking as a preference graph and aggregates the preference graphs to generate the aggregate ranking. We want to give different weights to different rankers (or the corresponding preference graphs) depending on their *qualities* or *goodness* on a given query. We want to assign the weights in an unsupervised manner. At the same time, we want the weight assignment and ranking aggregation algorithms to involve low cost operations, so that the algorithm is fast and suitable for real time processing. Although the algorithm is developed keeping in mind the metasearch problem, it can be used for all the applications/frameworks mentioned earlier in this section