Rank Aggregation for Meta-search Engines

Ka Wai, Lam
kwlam1i@ine.cuhk.edu.hk

Chi Ho, Leung
chleun1i@ine.cuhk.edu.hk

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong SAR

ABSTRACT

In this paper, we present an algorithm for merging results from different data sources in meta-search engine. We further extend one that has developed for ranking players of a round-robin tournament to a more general one when the ranking input is given from multiple sources. The problem in meta-search engine can be represented by a complete directed graph which can be used by the Majority Spanning Tree (MST) algorithm [3]. It is useful especially when the system must integrate and merge the query results that are returned from various search engines in a consistent manner.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval

General Terms

Ranking Algorithms

Keywords

Meta-search Engines, Rank aggregation

1. INTRODUCTION

An application of information retrieval for the Internet is producing the ordered documents in response to a retrieval query sent to a database or search engine - MetaSearch. It combines the output of "complete" search engines, as a kind of post-processing, value-adding stage. One problem often encountered in this environment is the procedure to rank relevant WWW pages resulting from user queries method to solve it qualitatively since in most cases the only available information returned from the sources is the ranking of the query results without any further quantitative values.

2. PRELIMINARIES

In this section, we define the terminology required for the MST algorithm and the corresponding results for ready reference. We modify the previous results to consider digraphs with arcs running from both i to j and j to i. Let $G = (V, A)$ be such a digraph with nonnegative weights on its arcs, and let T be a spanning tree of G. Let $(i, j) \in T$ defines a partition of V into V_i and V_j, where $T(V_i)$ that includes vertex i and $T(V_j)$ induces a tree on V_j containing vertex j.

1. A directed cutset (V_i, V_j) is defined as $(V_i, V_j) = \{(k, l) | k \in V_i, l \in V_j\}$ The weight $W\{(V_i, V_j)\}$ of a directed cutset (V_i, V_j) of a digraph G is defined as $W\{(V_i, V_j)\} = \sum_{k \in V_i, l \in V_j} w_{kl}$, where w_{kl} is the weight attached to the arc (k, l).
2. A cutset $[V_i, V_j]$ is defined as $(V_i, V_j) \cup (V_j, V_i)$. The weight of a cutset $[V_i, V_j]$ is defined as $W\{[V_i, V_j]\} = W\{(V_i, V_j)\} - W\{(V_j, V_i)\}$.
3. A spanning tree T of digraph $G = (V, A)$ is said to be a majority spanning tree if
$\{\forall_{i,j} (i, j) \in T\} \rightarrow \{W[V_i, V_j] \geq 0\}$

And the following results have been obtained

1. For every connected digraph $G = (V, A)$ and every non-negative weight function W on the arcs there exists a majority spanning tree. Let R be any optimal ranking of a set of documents represented by a complete digraph $G = (V, A)$. Then $G_R = (V_R, A_R)$ is an MST of G.
2. For any optimal ranking R and $1 \leq i \leq j \leq n, G_R^{ij}$ must be an MST of G_R^{ij}.

3. CONVERSION TO COMPLETE DIRECTED GRAPH

We first show how a set of ranking lists from various data sources can be transformed into a complete graph. A vertex V represents a document; the number on each arcs denotes the number of data sources that have ranked the document represented by the tail above the one represented by the head vertex. That means $W(i, j)$ equals the number of search engines which rank document i before document j. For those vertex pairs that have not been compared by any search engine, we assume they have equal chance to be ranked before each other, so we put 0.5 as the weight of these arcs for both directions.

Let us have an example where there are three data sources D_1, D_2, D_3 which return the ranked documents as follows: $D_1 = (C, B, A), D_2 = (C, E, D), D_3 = (A, C, D, E)$. Let $score(id, i)$ denotes the score for a particular document with the identification, id, from a data source, i. It is defined

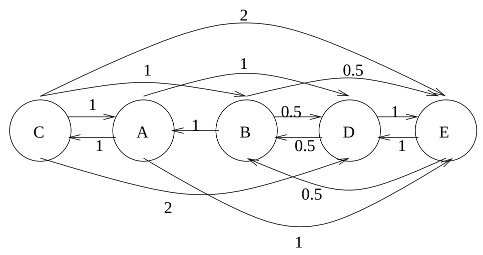

Figure 1: The complete directed graph

as $(N_i - j) - (j - 1) \equiv N_i - 2j + 1$ where j is the j-th ranking position for the document, id, in i and N_i is the length of D_i. So the overall score for a particular document is denoted by $\sum_i score(id, i)$. So for the above the scores for the documents A, B, C, D, and E are 1, 0, 5, -3, and -3. We then use this score information as the initial ranking in a non-increasing order. Thus, the initial ranking is (C, A, B, D, E), which represents document C gets the highest ranking and document E gets the lowest. The directed graph corresponding to the above data has been depicted in Figure 1.

4. THE MST ALGORITHM

The basic structure of the MST algorithm is shown in Algorithm 1. It consists of two main steps: **1. Conflicts Discovery**– Calculates the cutsets of the subgraphs. When the cutset is less than zero, by definition 2, the subgraph is not a MST. **2. Swapping**– When the cutset is less than zero, the subgraph is in wrong ordering. By swapping the vertexes we can obtain an MST of the subgraph.

The cutset of G^{AB} (which is a subgraph of G consists of documents A and B), $[A, B]$ is -1 .So, the MST algorithm will swap these vertices to obtain the ranking (C, B, A, D, E). Now we observe that there is no violating cutset in G or in any of its induced subgraphs G^{ij} implying that the MST algorithm cannot improve this ranking any further.

Algorithm 1 The MST algorithm

```
Set up G with appropriate weights
repeat
  $swap = false$
  for $i = 1$ TO $size - 1$ do
    for $j = i + 1$ TO $size$ do
      for $k = i$ TO $j - 1$ do
        if Cutset(i, k, j)< 0 then
          //it is a invalid MST then swap
          Swap $Set([i,k],[k+1,j])$
          $swap = true$
          BREAK (i-loop)
        end if
      end for
    end for
  end for
until $swap = false$
```

5. EXPERIMENTS

We have described a new method for ranking in meta-search. Now, we would like to compare it with other algorithms: **1. Preference function** [2] **2. Borda-Fuse and Weight Borda-Fuse** [1] **3. Linear** - A score, $s(d)$, is assigned to each element. It is the length of the list minus element's position, then we will calculate a sum which is taken over all constituent search engine $S_f(d) = \sum_i \alpha_i s_i(d)$ where α_i is the weight of the data source. **4. Exponential** - This is similar to the Linear method, we multiply the score by a exponential function $(score(p) = \sum_i (1 - e^{\frac{-1}{rank_i(p)}})$ where p is the page and $rank_i(p)$ is the ranking of page p in list i.).

Table 1: Results

Algorithm	k-distance	SD	Time(sec)
MST	10.56	5.49	2.1846
Linear	16.76	6.06	0.0135
Exponential	17.36	6.33	0.0157
Preference Function	21.06	5.59	0.1979
Borda-fuse	45.35	14.21	0.0129

We use four search engines; Yahoo, Alltheweb, Lycos, AltaVista; and 37 different queries – affirmative action, alcoholism, amusement parks, architecture, bicycling, blues, cheese, citrus groves, classical guitar, computer vision, cruises, death valley, field hockey, gardening, graphic design, hiv, java, lipari, lyme disease, mutual funds, national parks, parallel architecture, penelope fitzgerald, recycling cans, rock climbing, san francisco, shakespeare, stamp collecting, sushi, table tennis, telecommuting, thailand tourism, vintage cars, volcano, zen buddhism and zener – are sent to each search engine, retrieving 20 pages from each and combine their ranking by the algorithms. We measure the accuracy of the final ranking by k-distance [2]. All the experiments have been implemented in Python v1.5.2 and executed on a Sun Ultra 5 machine. The results are shown in Table 1. From the results, we know that the MST algorithm can obtain a better result, but it is quite time consuming. Borda-fuse is quick, but its results are less accuracte. It is because MST performs a larger search compared with the other methods.

6. CONCLUSION

In this paper, we propose a novel algorithm for rank aggregation in meta-search engines. We also compare our proposed strategy with other algorithms and show that the MST algorithm can obtain more accurate results.

7. ACKNOWLEDGMENT

This work is supported in part by the Earmarked Grant, CUHK #4351/02E, from the Research Grants Council (RGC) of the Hong Kong Special Administrative Region.

8. REFERENCES

[1] J. A. Aslam and M. Montague. Models for metasearch. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 276–284. ACM Press, 2001.

[2] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[3] M. Kaykobad, Q. Ahmed, A. Shafiqul Khalid, and R. Al Bakhtiar. A new algorithm for ranking players of a round robin tournament. *Computer Operations Research*, 22:221–226, 1995.