Condorcet fusion for blog opinion retrieval

Shengli Wu
School of Computer and Telecommunication
Jiangsu University, Zhenjiang, China
s.wu@uju.edu.cn

Xiaoqin Zeng
College of Computer and Information Engineering
Hehai University, Nanjing, China
xzeng@hhu.edu.cn

Abstract

Blogs have been popular social networking platforms in recent years. Blog opinion retrieval is one of the key issues that needs to be solved. In this paper, we investigate if the Condorcet fusion and the weighted Condorcet fusion can be used for effectiveness improvement of blog opinion retrieval. The experiments carried out with the data set from the TREC 2008 Blog track show that the Condorcet fusion is effective and the weighted Condorcet fusion, with its weights trained by linear discriminant analysis, is very effective. Both of them outperform the best component result by a clear margin.

1 Introduction

Blogs have been popular social networking platforms [7, 11] in recent years. Blog opinion retrieval is one of the key issues that needs to be solved. Usually, an opinion retrieval system is implemented by enhancing an ordinary information retrieval system (search engine) with an opinion finding mechanism, which may rely on a lexicon of subjective words and phrases, gathered from a variety of manually or automatically built lexical resources. Therefore, blog opinion retrieval systems are more complicated than conventional information retrieval systems, and many different kinds of techniques can be used together in any individual blog opinion retrieval systems. Previous research (e.g., in [12]) suggests that data fusion is an effective technique for blog opinion retrieval. However, the data fusion methods investigated in [12] are score-based methods, which require that all component retrieval systems (search engines) provide scores for every document retrieved. This may not always be possible in real applications. In this paper, we are going to investigate Condorcet fusion and weighted Condorcet fusion for blog opinion retrieval. Both of them are ranking-based methods and there is no requirement for scoring information.

2 Condorcet and weighted Condorcet fusion

In recent years, information retrieval has experienced a prosperous period of research and application on web search and various kinds of on-line information systems. Many different models such as the boolean model, the vector space model, the probabilistic model, and so on, have been proposed, and many other techniques such as query expansion, user feedback, using of noun or other types of phrases, structural analysis, semantic analysis, link analysis, and so on, are also very commonly used. Because many of these techniques are competitive, it is possible to combine results from different retrieval systems or different components or features in the same individual retrieval system so as to obtain more effective result. This is the primary idea behind data fusion [13].

In information retrieval, data fusion can be divided into two categories: score-based methods and ranking-based methods. Score-based methods are applicable when all retrieved documents are associated with scores. For example, if we have two component results $R_1 = <$ (d_1, 0.8), (d_2, 0.7), (d_4, 0.5), (d_3, 0.2)$>$, and $R_2 = <$(d_3, 0.7), (d_4, 0.6), (d_2, 0.5), (d_1, 0.2)$>$. Each of them is a ranked list of documents (ranked high to low from left to right). Scores which are associted with documents are also shown. We may use the averaging method that averages scores of all the documents, thus the fused result by such a method is $F = <$(d_2, 0.6), (d_4, 0.55), (d_1, 0.5), (d_3, 0.45)$>$. If no scoring information is available, then ranking-based method can be used. Among those ranking-based methods, Condorcet fusion is very distinctive.

A Condorcet method[5], named after the French mathematician and philosopher Marie Jean Antoine Nicolas Caritat, the Marquis de Condorcet, is a single-winner election method that ranks the candidates in order of preference. It is a pairwise voting, i.e., it compares every possible pair of candidates to decide the preference of them. A matrix can be used to present the competition process. [1] Every candidate appears in the matrix as a row and as a column as

[1] See http://en.wikipedia.org/wiki/Condorcet method

1529-4188/12 $26.00 © 2012 IEEE
DOI 10.1109/DEXA.2012.23

well. If there are m candidates, then we need m^2 elements in the matrix in total. Initially 0 is written to all the elements. If d_i is preferred to d_j , then we add 1 to the element at row i and column j (a_{ij}). The process is repeated until all the ballots are processed. For every element a_{ij}, if $a_{ij} > m/2$, then d_i beats d_j ; if $a_{ij} < m/2$, then d_j beats d_i; otherwise ($a_{ij} = m/2$), there is a draw between d_i and d_j. The total score of each candidate is quantified by summarizing the raw scores it obtains in all pairwise competitions. Finally the ranking is achievable based on the total scores calculated. This method can be used for data fusion in information retrieval if we regard each document as a candidate and each component result as a voter. Next let us take an example to illustrate how the Condorcet voting can work as a data fusion method.

Example 1. Let us assume that $R_1 = < d_2, d_3, d_1, d_4 >$, $R_2 = < d_3, d_4, d_1, d_2 >$, and $R_3 = < d_1, d_3, d_2, d_4 >$. Now we use the Condorcet method to fuse it. In R_1, d_2 has higher preference than d_3, d_1, and d_4; d_3 has higher preference than d_1 and d_4; and d_1 has higher preference than d_4. We add 1 to the corresponding units and the matrix looks like this:

R		Opponent			
u		d_1	d_2	d_3	d_4
n	d_1	-	0	0	1
n	d_2	1	-	1	1
e	d_3	1	0	-	1
r	d_4	0	0	0	-

We continue processing with R_2 and R_3 and the matrix is as follows:

R		Opponent				Total Scores
u		d_1	d_2	d_3	d_4	
n	d_1	-	2	1	2	2
n	d_2	1	-	1	2	1
e	d_3	2	2	-	3	3
r	d_4	1	1	0	-	0

Note that there are 3 information retrieval systems in total. For each element (a_{ij}) in the matrix, if a_{ij} is 2 or above, then d_i is preferred to d_j; if a_{ij} is 1 or less, then d_j is preferred to d_i. For the document in each row, we count how many times it wins over other documents. The total number of wins is written down on the right side of the matrix. The final fused ranking is <(d_3, 3), (d_1, 2), (d_2, 1), (d_4, 0)>. One thing needs to be noticed is: in the sum matrix, it is possible that more than one document obtains the same score, then we say there is a tie between those documents. How to rank those tied documents is also an important issue. However, in this work, we do not address it and just take a simple solution by ranking those tied documents randomly.

The above Condorcet method treats all the component results equally, or it can be regarded as an equal weight is assigned to all the component results. We can imagine that different weights might be assigned to different information retrieval systems for some reason. One possible reason is that we know the goodness of those information retrieval systems involved. On the other hand, the weighted Condorcet fusion can be regarded as a general form of the Condorcet fusion, while the Condorcet fusion is a special form of the weighted Condorcet fusion in which all weights are equal to 1.

In the above example, if retrieval results R_1, R_2, and R_3 are assigned weights of 4, 2, and 1, respectively, then the matrix is as follows when all the component results are processed.

R		Opponent				Total Scores
u		d_1	d_2	d_3	d_4	
n	d_1	-	2+1	1	4+1	1
n	d_2	4+1	-	4	4+1	3
e	d_3	4+2	2+1	-	4+2+1	2
r	d_4	2	2	0	-	0

In this example the maximum possible raw score for any item is 4+2+1 = 7. Therefore, a raw score of 4 or above means a win. The total scores for d_1, d_2, d_3, and d_4 are: score(d_1) = 1 (3:1:5), score(d_2) = 3 (4:4:5), score(d_3) = 2 (6:3:7), and score(d_4) = 0 (2:2:0). Therefore, the final ranking is changed to $< d_2, d_3, d_1, d_4 >$. □

Previous experiments [8] suggest that Condorcet fusion is a good data fusion method. When the performance of component results or the similarity among component results varies, weighted Condorcet fusion is a better option than Condorcet fusion. However, only a very primitive weighting method was mentioned in [8]: for a group of information retrieval systems, their effectiveness is evaluated over a group of training queries by a specific measure, for example, average precision, then the values obtained as such are used as their weights for the weighted Condorcet.

Next we discuss how to apply linear discrimination to training weights for weighted Condorcet fusion. Linear discriminant analysis (LDA) is a method used in statistics to find a linear combination of features that characterizes or separates two or more classes of instances [1]. This approach estimates the parameters of the linear discriminant directly from a given labelled sample through a search for the parameter values that minimize an error function. The key issue in Condorcet fusion is the pairwise document competition. If both documents involved are relevant (or irrelevant) at the same time, then how to rank them is not important. Since no matter which document wins the competition, the performance of the result will not be affected. What matters is: if a relevant document and an irrelevant document are in a pairwise competition, then we wish that the relevant document is able to win.

Suppose there are m information retrieval systems S_1, S_2, ..., S_m. For query Q, each of them returns a ranked list of documents $R_j (1 \leq j \leq m)$ and D is the set of all the documents involved. For simplicity, we assume that all R_js comprise the same group of documents. We may divide D into two sub-collections: relevant documents D_r and irrelevant documents D_i. There are $|D_r|$ documents in D_r and $|D_i|$ documents in D_i. If we choose one from each collection, then we have a total number of $2|D_r||D_i|$ ranked pairs. Note that $< d_a, d_b >$ and $< d_b, d_a >$ are different pairs. For all $2|D_r||D_i|$ pairs, we divide them into two classes: Class C_g and Class C_b. Class C_g comprises all those pairs in which a relevant document is ranked ahead of an irrelevant document, represented by '+1'; and Class C_b comprises all those pairs in which an irrelevant one is ranked ahead of a relevant one, represented by '-1'. For each ranked pair, we check every component result R_i to see if it is supported or not. If the ranked pair $< d_a, d_b >$ is supported by R_i, which means that d_a is also ranked ahead of d_b in R_i, then we use '+1' in the corresponding column f_i to represent it; If the ranked pair $< d_a, d_b >$ is not supported by R_i, which means that d_b is ranked ahead of d_a in R_i, then we use '-1' to represent it. For all the ranked pairs, we repeat this process over all component results. Thus for each ranked pair (instance), it has m features, each of which is obtained from a component information retrieval system S_i.

Example 2. Let $D_r = \{d_1, d_3, d_5\}$, $D_i = \{d_2, d_4\}$, $R_1 = < d_1, d_3, d_2, d_4, d_5 >$, $R_2 = < d_2, d_1, d_3, d_5, d_4 >$, $R_3 = < d_5, d_4, d_3, d_1, d_2 >$, then Table 1 can be used to represent all the instances with their features.

Now we want to distinguish the instances of the two classes by a linear combination of m features. Let $g(f_1, f_2, ..., f_m) = \sum_{i=1}^{n} w_i f_i + w_0$, if $g(f_1, f_2, ..., f_m) > 0$, then the instance in question belongs to Class C_g; if $g(f_1, f_2, ..., f_m) \leq 0$, then the instance belongs to Class C_b. For the above example, by using LDA[2] we obtain the weights w_1, w_2, and w_3 for f_1, f_2, and f_3 are 1.265, 1.342, and 1.897, respectively. Note that in Table 1, each feature f_i (column) is obtained from a given retrieval system S_i, thus the number of information retrieval systems is equal to the number of features and the weight obtained for f_i is for S_i as well. □

Table 1. Classification and features of all the instances (pairs) in Example 2

Number	Pair	f_1	f_2	f_3	Category
1	$< d_1, d_2 >$	+1	-1	+1	+1
2	$< d_2, d_1 >$	-1	+1	-1	-1
3	$< d_1, d_4 >$	+1	+1	-1	+1
4	$< d_4, d_1 >$	-1	-1	+1	-1
5	$< d_3, d_2 >$	+1	-1	+1	+1
6	$< d_2, d_3 >$	-1	+1	-1	-1
7	$< d_3, d_4 >$	+1	+1	-1	+1
8	$< d_4, d_3 >$	-1	-1	+1	-1
9	$< d_5, d_2 >$	-1	-1	+1	+1
10	$< d_2, d_5 >$	+1	+1	-1	-1
11	$< d_5, d_4 >$	-1	+1	+1	+1
12	$< d_4, d_5 >$	+1	-1	-1	-1

3 Experimental settings

As one of the major events in information retrieval evaluation, TREC (Text REtrieval Conference) has been held by NIST (National Institute of Standards and Technology, USA) annually since 1992. [3] In 2008, the Blog track was undertaken among several others such as enterprise, legal, million query, relevance feedback, and so on. In the TREC 2008 Blog track, "Blog06" test collection was used. Opinion retrieval was one of the tasks in the Blog track. It was used to locate blog posts that expressed an opinion about a given target. A target could range from the name of a person or organization to a type of technology, a new product, or an event. A total of 150 topics (851-950, 1001-1050) were used in the 2008 Blog track. Among them, 50 (1001-1050) were new ones, 50 (851-900) were used in the 2006 Blog track and 50 (901-950) were used in the 2007 Blog track. An example of a topic is shown below.

Topic 1001
Description: Find opinions of people who have sold a car, purchased a car, or both, through Carmax.
Narrative: Relevant documents will include experiences from people who have bought or sold a car through Carmax and expressed an opinion about the experience. Do not include posts where people obtain estimates from Carmax but do not buy or sell an auto with Carmax.

For most opinion retrieval systems, the opinion finding is a two-stage process. The first stage is to generate baseline ad hoc retrieval runs. 5 standard baselines were provided by NIST for the 2008 Blog track. Information about them can be found in [9]. Then, based on any of these baselines, the participants can submit their final runs. 19 groups submitted a total of 191 runs to the opinion-finding task.

Each submitted run consists of up to 1000 retrieved documents for each topic. The retrieval units are the documents from the permalinks component of the Blog06 test collection. The content of a blog is defined as the content of the post itself and all the comments to the post.

Analogous to other TREC tracks, the Blog track uses the pool policy for retrieval evaluation: A pool was formed

[2] IBM SPSS is used for LDA in this study. Its web site is located at http://www-01.ibm.com/software/uk/analytics/spss/

[3] Its web site is located at http://trec.nist.gov/

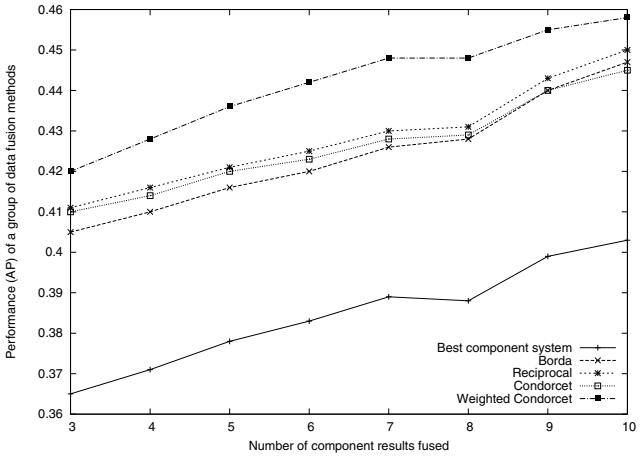

Figure 1. Performance (AP) of several data fusion methods with the TREC 2008 data set

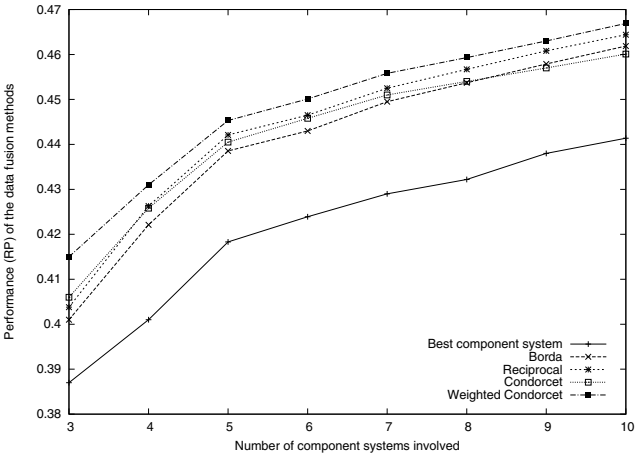

Figure 2. Performance (RP) of several data fusion methods with the TREC 2008 data set

from the submitted runs of the participants. The two highest priority runs per group were pooled to depth 100. The remaining runs were pooled to depth 10. Only those documents in the pool were judged. All the documents that were not in the pool were treated as irrelevant documents.

Apart from the Condorcet fusion and the weighted Condorcet fusion, two other ranking-based methods, Borda count [4] and the Reciprocal Rank Fusion (RRF) [3], are also tested. Borda count works like this: for a ranked list of documents, score are assigned to those documents linearly first. For example, if there are 1000 documents in the result, then 1, 0.999, 0.998,..., 0.001 will be assigned to documents at rank 1, 2, 3, ..., 1000, respectively. Then the averaging method (averaging scores for every document) will be used to combine all the component results to form the new result. Instead of using a linear method for converting rank to score as in Borda count, RRF uses a reciprocal rank function. For any document at ranking position t, $1/(k+t)$ is used to convert its ranking into score. Here k is a constant. As in [3], we also set $k = 60$ in our experiment.

For weighted Condorcet, training is needed. We divide 150 queries into two groups: odd-numbered (oq) and even-numbered (eq) queries. oq is used for weights training and eq is used for testing and vice versa. It is referred to as two-way cross validation in [8].

4 Experiments and results

The experiment is conducted to investigate the performance of the data fusion methods involved (Condorcet fusion, weighted Condorcet fusion, Borda count). All 191 runs submitted to the TREC 2008 blog opinion track are used. From all available ones, we randomly choose 3, 4, ..., 10 runs, and then fuse them using different fusion methods. The best component system is used as the baseline.

Four metrics, which are AP(average precision over all relevant document levels), RP(recall-level precision), P@10 (precision at 10 document level), and RR(reciprocal rank), are used for retrieval evaluation. Figures 1-3 show the experimental results. Each data point is the average of 200 trials. The result for P@10 is not shown because it is very analogous to that for RR. In Figures 1-3, we can see that all the data fusion methods outperform the best component system by a clear margin, and the difference between any of the data fusion methods and the best result is significant at a level of 99%. Weighted Condorcet fusion performs the best, while Borda count, Condorcet fusion, and RRF are close. When different metrics are used, the improvement rates of the fused results over the best component result are different. The improvement rate is the largest on AP and the smallest on RR. When AP is used, weighted Condorcet, RRF, Condorcet, and Borda outperform the best component system by 15.0%, 10.7%, 10.5%, and 10.1%, respectively. When RR is used, the corresponding figures are 5.2%, 4.4%, 4.3%, and 4.1%, respectively.

Finally, let us take a look at the effect of the number of component results on fusion performance. Figure 4 shows the result. This time we only consider the weighted Condorcet fusion. From Figure 4 we can see that the performance improvement over the best component result is different across different measures. AP is most successful than the three others, while PR, P10, and RR are quite close. Focused on any individual measure, we find that the percentage of improvement over the best component result is quite stable when different number of component results are fused.

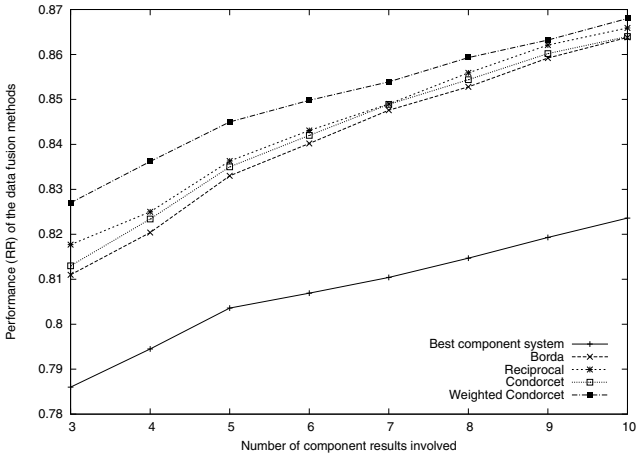

Figure 3. Performance (RR) of several data fusion methods with the TREC 2008 data set

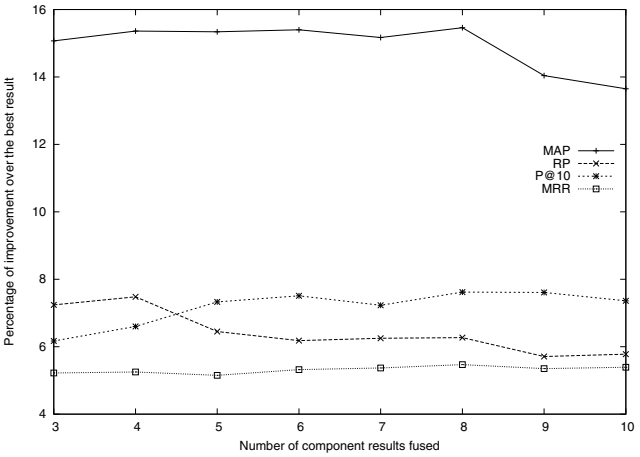

Figure 4. Improvement (percentage) of the weighted Condorcet over the best component result when different number of component results are fused

5 Conclusions

From the above discussion, we can see that the experimental result is very positive and shows that ranking-based data fusion methods such as Borda count, RRF, and Condorcet fusion are effective techniques for retrieval performance improvement. The result also shows that weighted Condorcet fusion, with its weights trained by linear discriminant analysis, is a very effective method. All these methods can be useful for improving the performance of blog opinion retrieval.

For blog opinion retrieval, the data fusion technique can be further investigated in a few directions. One is to compare score-based methods (e.g., CombSum, CombMNZ, the linear combination method) with ranking-based methods (e.g., Borda count, Condorcet fusion, RRF) [6, 10]; the second is to focus on fusing a few top systems for further performance improvement [2, 14]; finally, non-linear combination methods may be investigated. These remain to be our further work.

References

[1] E. Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2010.

[2] S. Beitzel, E. Jensen, A. Chowdhury, D. Grossman, O. Frieder, and N. Goharian. On fusion of effective retrieval strategies in the same information retrieval system. *Journal of the American Society of Information Science and Technology*, 55(10):859–868, 2004.

[3] G. V. Cormack, C. L. A. Clarke, and S. Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning mthods. In *Proceedings of the 32nd Annual International ACM SIGIR Conference*, pages 758–759, Boston, MA, USA, July 2009.

[4] J. C. de Borda. Memoire sur les elections au scrutin. 1781.

[5] M. de Condorcet. Essai sur l'application de l'analyse $\grave{a}$ la probabilit$\acute{e}$ des d$\acute{e}$cisions rendues $\grave{a}$ la pluralit$\acute{e}$ desvoix. 1785.

[6] F. Hsu and I. Taksa. Comparing rank and score combination methods for data fusion in information retrieval. *Information Retrieval*, 8(3):449–480, 2005.

[7] Y. Huang, T. Huang, and Y. Huang. Applying an intelligent notification mechanism to blogging systems utilizing a genetic-based information retrieval approach. *Expert Systems with Applications*, 37(1):705–715, January 2010.

[8] M. Montague and J. A. Aslam. Condorcet fusion for improved retrieval. In *Proceedings of ACM CIKM Conference*, pages 538–548, McLean, VA, USA, November 2002.

[9] I. Ounis, C. Macdonald, and I. Soboroff. Overview of the trec-2008 blog track. In *Proceeding of the 17th Text Retrieval Conference*, Gaithersburg, MD, USA, 2008.

[10] M. E. Renda and U. Straccia. Web metasearch: rank vs. score based rank aggregation methods. In *Proceedings of ACM 2003 Symposium of Applied Computing*, pages 841–846, Melbourne, USA, April 2003.

[11] O. Vechtomova. Facet-based opinion retrieval from blogs. *Information Processing & Management*, 46(1):71–88, 2010.

[12] S. Wu. Applying the data fusion technique to blog opinion retrieval. *Expert Systems with Applications*, 39(1):1346–1353, January 2012.

[13] S. Wu. *Data Fusion in Information Retreival*. Springer, 2012.

[14] S. Wu and S. McClean. Improving high accuracy retrieval by eliminating the uneven correlation effect in data fusion. *Journal of American Society for Information Science and Technology*, 57(14):1962–1973, December 2006.