Information Processing and Management 51 (2015) 306–328

Contents lists available at ScienceDirect

Information Processing and Management

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

Learning combination weights in data fusion using Genetic Algorithms

Kripabandhu Ghosh [a,*], Swapan Kumar Parui [a,1], Prasenjit Majumder [b,2]

[a] Indian Statistical Institute, 203 Barrackpore Trunk Road, Kolkata 700108, West Bengal, India
[b] Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Near Indroda Circle, 382007 Gujarat, India

ARTICLE INFO

Article history:
Received 2 August 2013
Received in revised form 26 November 2014
Accepted 12 December 2014
Available online 9 January 2015

Keywords:
Information retrieval
Data fusion
Linear combination
Genetic Algorithms

ABSTRACT

Researchers have shown that a weighted linear combination in data fusion can produce better results than an unweighted combination. Many techniques have been used to determine the linear combination weights. In this work, we have used the Genetic Algorithm (GA) for the same purpose. The GA is not new and it has been used earlier in several other applications. But, to the best of our knowledge, the GA has not been used for fusion of runs in information retrieval. First, we use GA to learn the optimum fusion weights using the entire set of relevance assessment. Next, we learn the weights from the relevance assessments of the top retrieved documents only. Finally, we also learn the weights by a twofold training and testing on the queries. We test our method on the runs submitted in TREC. We see that our weight learning scheme, using both full and partial sets of relevance assessment, produces significant improvements over the best candidate run, CombSUM, CombMNZ, Z-Score, linear combination method with performance level, performance level square weighting scheme, multiple linear regression-based weight learning scheme, mixture model result merging scheme, LambdaMerge, ClustFuseCombSUM and ClustFuseCombMNZ. Furthermore, we study how the correlation among the scores in the runs can be used to eliminate redundant runs in a set of runs to be fused. We observe that similar runs have similar contributions in fusion. So, eliminating the redundant runs in a group of similar runs does not hurt fusion performance in any significant way.

1. Introduction

Data fusion has been used as an effective tool for improving information retrieval performance. Several IR researchers have proposed several methods of combining two or more retrieved lists to produce a single list that contains the useful documents from all the lists at higher ranks. Given a set of retrieved lists (or runs, as they are popularly referred to) and given a fusion algorithm, it is vital to choose the combination weights which will result in improvement in performance. Previous results show that weighted fusions have scored over unweighted fusions when appropriate weights are assigned to the fused runs (Wu et al., 2009). Much research has been done on learning or optimizing the fusion weights. Vogt and Cottrell (1998) have tried to predict the fusion performance of run pairs using multiple linear regression based on several features. Wu et al.

* Corresponding author. Tel.: +91 8335045897; fax: +91 3325773035.
E-mail addresses: kripa.ghosh@gmail.com (K. Ghosh), swapan@isical.ac.in (S.K. Parui), prasenjit.majumder@gmail.com (P. Majumder).
[1] Tel.: +91 8335045897; fax: +91 3325773035.
[2] Tel.: +91 9712660746; fax: +91 07930520010.

http://dx.doi.org/10.1016/j.ipm.2014.12.002
0306-4573/

(2009) stated that power functions can be useful in finding good weights for fusion. Bartell et al. (1994) also tried to maximize fusion performance. In the last two papers, Conjugate Gradient method (Press et al., 1995) on Guttman's Point Alienation function (Guttman, 1978) was used. Vogt and Cottrell used golden search method (Press et al., 1995) for optimization.

The Genetic Algorithm (GA) has been used in finding useful solutions to optimization and search problems. GAs generate solutions to optimization problems by using techniques inspired by natural evolution. GAs can thus be used in learning the optimum fusion weights for a weighted linear combination of retrieval scores of different runs.

Next, we study how the use of the top ranked documents on linear combination of scores can be used to get an optimal retrieval performance. We consider runs at a given depth k per query and run the optimization algorithm on them. The optimum weights thus learned are tested on the runs. The idea of using the top k documents is modeled on Multi-armed Bandits problem (Auer et al., 1995) where the gambler has no initial knowledge about the levers and tries to maximize the gain based on existing knowledge of each lever. Here we attempt to use partial knowledge about the IR performance of each run to learn the optimal fusion weights. Pal et al. (2011) found that reducing pool size per topic does not have much effect on evaluation. We draw our motivation from this observation also.

We also study how the correlation between the scores of runs can help in removing the redundant runs in data fusion. We calculate the correlation values among the run pairs. First we choose the run pair with the highest correlation between them and drop the run which has the inferior retrieval performance. We fuse the remaining runs and see if there is any significant drop in fusion performance. If not, we consider the pair with the next highest correlation. We repeat the procedure until there is a significant drop in performance. This study is aimed at exploring if highly correlated runs in terms of retrieval scores have similar contribution in fusion performance.

Our contributions made in the present paper are summarized below:

1. Given a set of runs, a GA based approach to finding the optimal weights for an efficient fusion of these runs on the basis of their retrieval scores, is proposed.
2. It is shown that if the learning of the fusion weights by the GA is based only on the top-ranked documents, there is not much loss in efficiency of the resulting fusion. In other words, if only lower depths in the ranked pool of documents are used by the GA to learn the fusion weights, the performance of the resulting fusion is not hurt much.
3. A new approach to determination of the runs that make insignificant contributions in fusion, and hence to determination of the smallest subset of the runs to be fused, without much loss in efficiency, is proposed. It is based on the fusion weights learnt by the GA and the correlation coefficients between pairs of runs obtained on the basis their retrieval scores.

The rest of the paper is organized as follows:

We provide a discussion on the related works in Section 2. In Section 3, we describe the GA and discuss how this algorithm can be used in the present fusion problem. The experimental setup is described in Section 4. We present our experimental results and a comparative study in Section 5 and conclude in Section 6.

2. Related work

Work of different genres has been done on data fusion. Many new data fusion techniques have been proposed. On the other hand, approaches that focussed on improving the existing methods were also reported. Fox and Shaw proposed CombMIN, CombMAX, CombSUM, CombANZ and CombMNZ algorithms based on linear combinations of scores (Fox and Shaw, 1993). CombSUM and CombMNZ have emerged as effective methods based on linear combinations of scores. Lee (1997) performed an experiment on six submitted runs of TREC 3 and concluded that CombMNZ was slightly better than CombSUM. This claim, however, was contradicted in many works like Montague and Aslam (2001), Lillis et al. (2006), Wu et al. (2009), and no clear inference could be drawn about the supremacy of any single approach. Popular voting algorithms were also used effectively in data fusion. Montague and Aslam (2002) used popular voting method called Condorset method (named after French mathematician and philosopher Marquis de Condorset) to good effect in data fusion. The fusion algorithm was called Condorset fusion. Another voting algorithm, viz., Borda count (named after another French mathematician Jean-Charles de Borda) was used by Aslam and Montague (2001) and was referred to hitherto as Borda fusion. Cormack et al. (2009) showed that Reciprocal Rank Fusion paired with learning to rank outperforms Condorset fusion and individual rank learning method. Lillis et al. (2006) proposed a fusion algorithm named ProbFuse which estimated the probability of relevance of documents based on the position in the ranked list. Khudyak Kozorovitsky and Kurland (2011) used document cluster-based approach (named ClustFuse) to find retrieval scores in the fused list. But there is a fundamental factor differentiating the fusion algorithms. CombSUM (and other Comb variants), ProbFuse and ClustFuse make use of the relevance scores assigned to the documents in the input runs that are fused, while Condorset, Borda and Reciprocal Rank fusions use the rank of the documents in the ranked lists.

In the case of the algorithms that use the relevance scores, Lee (1997) introduced a score normalization scheme to the algorithms proposed by Fox and Shaw and showed that score normalization is important in data fusion. Similarly, Savoy (2004) used a new score normalization formula on linear combination fusion algorithms. He called the normalized score Z-Score. Montague and Aslam (2001) also studied different score normalization schemes.