# ABSTRACT

Information retrieval (IR) is the activity of obtaining information resources relevant to an information need from a collection of information resources. Searches can be based on metadata or on full-text (or other content-based) indexing. Most IR systems compute a numeric score on how well each object in the database matches the query, and rank the objects according to this value. The top ranking objects are then shown to the user.

Learning to rank represents an important class of supervised machine learning tasks with the goal of automatically constructing ranking functions from training data. Learning to rank or machine-learned ranking (MLR) is the application of machine learning, typically supervised, semi-supervised or reinforcement learning, in the construction of ranking models for information retrieval systems. Training data consists of lists of items with some partial order specified between items in each list. This order is typically induced by giving a numerical or ordinal score or a binary judgment (e.g. "relevant" or "not relevant") for each item. The ranking model's purpose is to rank, i.e. produce a permutation of items in new, unseen lists in a way which is "similar" to rankings in the training data in some sense.

Learning to Rank algorithms is broadly classified into three approaches namely Point-wise approach, Pair-wise approach and List-wise approach respectively.

In Point-wise approach it is assumed that each query-document pair in the training data has a numerical or ordinal score. Then learning-to-rank problem can be approximated by a regression problem — given a single query-document pair, predict its score. A number of existing supervised

machine learning algorithms can be readily used for this purpose. Ordinal regression and classification algorithms can also be used in point wise approach when they are used to predict score of a single query-document pair, and it takes a small, finite number of values.

In Pair-wise approach learning-to-rank problem is approximated by a classification problem, learning a binary classifier that can tell which document is better in a given pair of documents. The goal is to minimize average number of inversions in ranking. List-wise algorithms try to directly optimize the value of one of the evaluation measures, averaged over all queries in the training data. This is difficult because most evaluation measures are not continuous functions with respect to ranking model's parameters, and so continuous approximations or bounds on evaluation measures have to be used.

With the emergence and rapid explosion of web applications, ranking over various data sources is becoming more and more significant for several applications. Conventional learning-to-rank issue majorly focuses on one single type of objects. On the other hand, with the rapid development of the Web 2.0, ranking over several interrelated and heterogeneous objects becomes a general condition, e.g., the heterogeneous academic network. Most learning to rank algorithms are heavily dependent on a large amount of editorial labelled training data, which is time consuming and costly to obtain. Furthermore, collecting specific labelled data for different domains or different ranking applications is not scalable. To solve this problem, in this work, a cross domain ranking scenario1 and cross domain ranking scenario2 are proposed to simultaneously reduce loss functions associated to cross domains.

Cross domain ranking scenario 1 deals with in making use of the labelled information from existing source domain to build an accurate ranking model for the target domain. Here the dataset with low level features are used for training and dataset with high level features are used for testing. These feature spaces of the source domain and target domains may be different. Ranking models are constructed using learning to Rank algorithms. Rank Support Vector Machine (Rank SVM), RankBoost, RankNet (pair-wise approach) and AdaRank, (list-wise approach) are used as the basic learners for the cross domain ranking. The experiments were set up to simulate the cross domain ranking scenario using Learning to Rank for Information Retrieval (LETOR) dataset. Results showed that RankNet produced more precise results than other baselines.

The main goal of cross domain ranking scenario2 is to improve the ranking performance of documents even in the presence of documents from the multiple cross domains. Usually, in the learning based ranking algorithms, it is expected the training and testing data are drawn from the identical domain. Equally the inadequate data from cross domain cannot improve the exactness of ranking. Instead of creating a new model for target domain, an attempt is made to use existing model to suit for the source and target domain requirements. In this work, the learning of a cross domain ranking scenario2 in the target domain is done by exploiting the information from another domain data. The feature level information allocation and instance level information relocation are achieved with four learners namely RankNet, Ranking SVM, RankBoost and AdaRank. A learning algorithm would train a ranking model so as to enhance its performance. In this investigation, the suggested algorithms are evaluated using benchmark datasets for the information retrieval. The results presented show that the AdaRank algorithm achieves higher performance values in terms of Mean

Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG).

Also after applying Correlation based Feature Selection(CFS) algorithm in both cross domain ranking scenario1 and cross domain ranking scenario2 , it is observed that performance has been improved further.