Chapter V

MARS: Multiplicative Adaptive Refinement Web Search

Xiannong Meng
Bucknell University, USA

Zhixiang Chen
The University of Texas - Pan American, USA

ABSTRACT
This chapter reports the project MARS (Multiplicative Adaptive Refinement Search), which applies a new multiplicative adaptive algorithm for user preference retrieval to Web searches. The new algorithm uses a multiplicative query expansion strategy to adaptively improve and reformulate the query vector to learn users’ information preference. The algorithm has provable better performance than the popular Rocchio’s similarity-based relevance feedback algorithm in learning a user preference that is determined by a linear classifier with a small number of non-zero coefficients over the real-valued vector space. A meta-search engine based on the aforementioned algorithm is built, and analysis of its search performance is presented.

INTRODUCTION
Vector space models and relevance feedback have long been used in information retrieval (Baeza-Yates & Ribeiro-Neto, 1999; Salton, 1989). In the n-dimensional vector space model, a collection of n index terms or keywords is chosen, and any document d is represented by an n-dimensional vector \( d = (d_1, ..., d_n) \), where \( d \) represents the
relevance value of the $i$-th index term in the document. Let $D$ be a collection of documents, $R$ be the set of all real values, and $R^+$ be the set of all positive real values. It has been shown in Bollmann and Wong (1987) that if a user preference relation is a weak order satisfying some additional conditions then it can be represented by a linear classifier. That is, there is a query vector $q = (q_1, ..., q_n) \in R^n$ such that:

$$\forall d, d' \in D, d \rightarrow d' \iff q \cdot d < q \cdot d'. \tag{1}$$

Here, “$\cdot$” denotes the inner product of vectors. In general, a linear classifier over the vector space $[0,1]^n$ is a pair of $(q, \theta)$ which classifies any document $d$ as relevant if $q \cdot d > \theta$, or irrelevant otherwise, where the query vector $q \in R^n$, the classification threshold $\theta \in R^+$, and $[0,1]$ denote the set of all real values between 0 and 1. Recall that $q \cdot d$ is usually used as the relevance rank (or score) of the document $d$ with respect to user preference.

Let $D_r$ be the set of all relevant documents in $D$ with respect to a user’s information needs (or search query). Assume that a user preference relation has a simple structure with only two levels, one level consisting of all relevant documents and the other consisting of all irrelevant documents, and within the same level no preference is given between any two documents. Then, finding a user preference relation satisfying the expression (1) is equivalent to the problem of finding a linear classifier $(q, \theta)$ over $[0,1]^n$ with the property:

$$\forall d \in D, d \in D_r \iff q \cdot d > \theta, \tag{2}$$

where $q \in R^n$ is the query (or weight) vector.

The goal of relevance feedback in information retrieval is to identify a user preference relation with respect to his/her information needs from documents judged by that user. Since user preference relations vary between users and may have various unknown representations, it is not easy for an information system to learn such relations. The existing popular relevance feedback algorithms basically use linear additive query expansion methods to learn a user preference relation as follows:

- Start with an initial query vector $q_0$.
- At any step $k \geq 0$, improve the $k$-th query vector $q_k$ to:

$$q_{k+1} = q_k + \alpha_1 d_1 + \ldots + \alpha_s d_s, \tag{3}$$

where $d_1, ..., d_s$ are the documents judged by the user at this step, and the updating factors $\alpha_i \in R$ for $i = 1, ..., s$.

One particular and well-known example of relevance feedback is Rocchio’s similarity-based relevance feedback (Rocchio, 1971). Depending on how updating factors are used in improving the $k$-th query vector as in expression (3), a variety of relevance feedback algorithms have been designed (Salton, 1989). A similarity-based relevance feedback algorithm is essentially an adaptive supervised learning algorithm from examples (Chen & Zhu, 2000, 2002; Salton & Buckley, 1990). The goal of the algorithm is
Enhanced Utility in Search Log Publishing of Infrequent Items

www.igi-global.com/article/enhanced-utility-in-search-log-publishing-of-infrequent-items/115935?camid=4v1a