Can We Get A Better Retrieval Function From Machine?

Li Wang
Ross School of Business
University of Michigan
Ann Arbor, MI
wang@umich.edu

Weiguo Fan
Pamplin College of Business
Virginia Tech
Blacksburg, VA
wfan@vt.edu

Wensi Xi
Department of Computer Science
Virginia Tech
Blacksburg, VA
{xwensi, fox}@vt.edu

Edward A. Fox

Abstract

The quality of an information retrieval system heavily depends on its retrieval function, which returns a similarity measurement between the query and each document in the collection. Documents are sorted according to their similarity values with the query and those with high rank are assumed to be relevant. Okapi BM25 and their variations are very popular retrieval functions and they seem to be the default retrieval function for the IR research community; and there are many other widely used and well studied functions, for example, Pivoted TFIDF and INQUERY. Most of these retrieval functions being used today are made based on probabilistic theories and they are adjusted in real world according to different contexts and information needs. In this paper, we propose the idea that a good retrieval function can be discovered by a pure machine learning approach, without using probabilistic theories and knowledge-based techniques. Two machine learning algorithms, Support Vector Machine (SVM) and Genetic Programming (GP) are used for retrieval function discovery, and GP is found to be a more effective approach. The retrieval functions discovered by GP might be hard for human interpretation, but their performance is superior to Okapi BM25, one of the most popular functions. The new retrieval function is combined with query expansion techniques and the retrieval performance is improved significantly. Based on our observations in the empirical study, the GP function is more reliable and effective than Okapi BM25 when query expansion techniques are used.

Keywords

Machine Learning, Retrieval Function, Query Expansion

1. Introduction

Retrieval function is one of the most important components in an information retrieval system. For a query submitted by the user, retrieval function is used to measure the similarity between query and each document in the collection. Then documents are sorted according to their similarity measurements and the documents ranked to the top are assumed to be relevant to this specific query. Although there are many other criteria for evaluating an information retrieval system, such as response time of the search engine and volume of its collection, the accuracy of returned documents is a critical index for the performance of an IR system. The accuracy can be measured by recall, precision, or mean average precision (MAP) which is a compromise between recall and precision. Under the same conditions, such as the same document collection, preprocessing procedure for documents, indexing method and etc., the accuracy of an IR system completely depends on how effective its retrieval function is.

There are many popular and well established retrieval functions, such as Pivoted TFIDF [1] and Okapi BM25 [2]. They have been thoroughly studied, widely used in real world and proved to be effective. These functions are invented by information retrieval experts, guided by probabilistic theories and their prior experience. They share the same property that the function has simple format and can be easily interpreted. According to different contexts and information needs, many variations of these functions have been created in practice, by either adjusting parameter values or modifying some fragments in the function. The first approach can not alter the framework of a retrieval function. The later one is conducted either based on theories or by trial-and-error approach. The methodology of discovering new functions is not changed. The machine/statistical learning algorithms have been used; however their success is limited on parameter tuning and estimation. A new function or model still has to be proposed by human experts, and then its effectiveness is fine-tuned by a machine learning approach for different tasks.

Can we completely rely on machines to construct retrieval functions for us? Can the functions discovered by machine beat those made by human experts? This is the basic incentive of our research. We use two popular machine learning algorithms, kernel based Support Vector Machine (SVM) [3] and genetic programming (GP) [4], in our experiments for retrieval function discovery. The new retrieval functions found by each approach are compared with Okapi BM25, which has the best performance among traditional functions in our experiment. Kernel based SVM is found not an

effective tool for retrieval function discovery task in our experiment; but genetic programming gives inspiring results. Many retrieval functions constructed by GP are able to outperform Okapi BM25.

Based on the best GP function, several automatic query expansion algorithms are used to further improve the retrieval performance. An empirical study is conducted to explore how GP-discovered function works with query expansion algorithms. Large scale experiments are done for this purpose. When combined with query expansion, the GP function can get a significantly better performance, measured as MAP, than the baseline system using Okapi BM25. From the observation of MAP surface, the GP function is more robust to the parameter settings than Okapi BM25; and with GP function people have more chance to reach the optimal settings of query expansion using design of experiment (DOE) approach in practice.

The reminder of the paper is organized as follows: Section 2 gives a brief description of our automatic retrieval function discovery mechanism and the retrieval function performance comparison; Section 3 introduces the empirical study on query expansion algorithms; Section 4 concludes this paper.

2. Automatic Retrieval Function Discovery

The problem we are studying is the traditional 2-class classification problem. Given a user query and a document, we need to predict whether this document belongs to class 0 (irrelevant document) or class 1 (relevant document), according to predictors extracted from the query and document. The training data can be obtained from previous TREC results. For the predictors, we take those used in traditional retrieval functions, such as term frequency within the document (tf), term frequency within the query (qtf), document frequency for a term (df) and etc.

The 2-class classification problem has been well studied and there are quite a few existing algorithms suitable for this task, such as linear regression, logistics regression, Linear Discrimimnant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) [5]. All of these methods need a given model; then they minimize the misclassification rate or other object function by adjusting parameter values within that framework. A trial-and-error approach has to be used to explore the infinite model space, which consists of all possible combinations of predictors. Therefore these algorithms largely depend on human input and it is hard to arrive at a potentially useful model which might has extremely complex format. Support Vector Machine is another effective tool for 2-class classification problem and it also requires a given model. However when using the kernel method,

SVM essentially transforms the given linear predictor space into a nonlinear higher-dimensional or even infinite-dimensional predictor space, reproducing kernel Hillbert space (RKHR). Therefore it has potential to provide a good retrieval function which is flexible and complex.

Genetic programming is another algorithm that can be used. Different from the approaches above, there is no sound mathematical explanation for its success in practice. GP algorithm often shows its edge where the solution has high complexity and when the usual analytical methods fail. The details of experiment setting can be found in [6].

We used kernel-based SVM and GP in our experiments for retrieval function discovery. Some mechanisms are needed to prevent over-fitting and reduce predicting error. K-fold cross-validation is a popular method for this purpose. However, the training process for retrieval function discovery is already computational intense, we can not afford with the K-fold cross-validation here. Instead, we use the setup with independent training, validation and testing data to control over-fitting and choose models. These three independent data are randomly picked from previous TREC queries and data sets.

After choosing different kernels for SVM and manually expanding the predictor space, the SVM algorithm still can not provide a retrieval function with satisfactory performance on the testing data. But the GP method gives us pretty inspiring results. It generates a group of models that have better MAP than Okapi BM25 on validation and testing data sets. These GP functions are further tested using the 150 queries from TREC 6 - 8 and 50 new queries from Robust Track of TREC 2003. Figure 1 shows the performance comparison between Okapi BM25 and the GP function used in our retrieval system on different fields of queries.

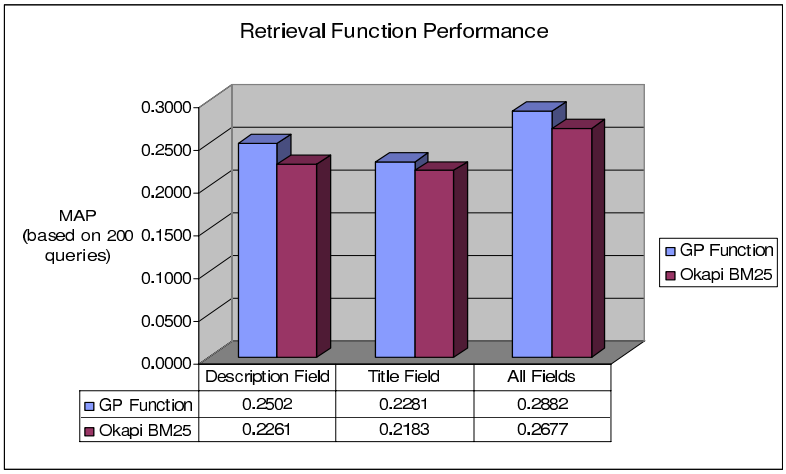
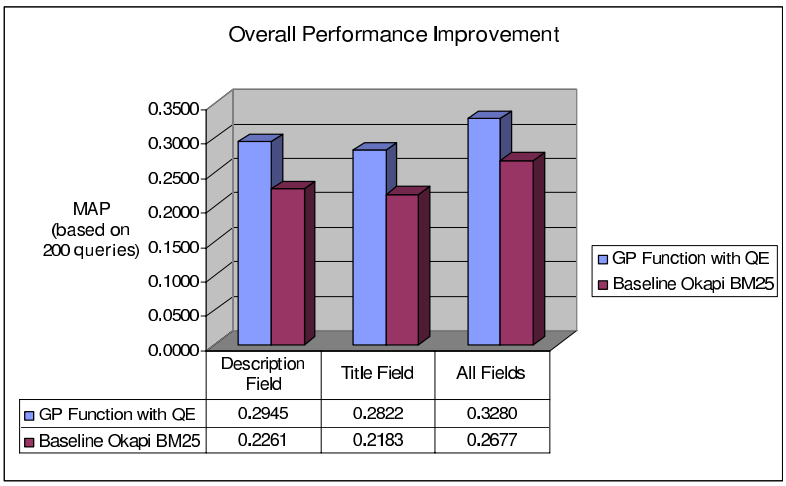


Figure 1

Based on 200 queries in previous TRECs, the GP function improves the MAP performance by 10.6%, 4.5% and 7.7%, on description field, title field and all fields, respectively. The performance improvement is significant using an ANOVA test.

3. Empirical Study on Query Expansion

Query expansion, or blind feedback, is a useful technique for boosting retrieval performance. It uses a two stage retrieval strategy. In the first stage, a preliminary rank list for the query is returned using the retrieval function; then without actual user feedback, it assumes the top *D* documents in that rank list are relevant to the query and uses an algorithm to pick *T* words from the words contained in original query as well as in those *D* documents. There are many popular query expansion techniques, such as Rocchio, Ide Dec-Hi, KLD, RSV and CHI [7], which use different algorithms to pick the new query with *T* words in it. Figure 2 shows the performances of our baseline system with Okapi BM25 and our submissions, where GP function and query expansion technique are combined.



	Description Field	Title Field	All Fields
GP Function with QE	0.2945	0.2822	0.3280
Baseline Okapi BM25	0.2261	0.2183	0.2677

Figure 2

Evaluated by 200 queries from TREC 6-8 and TREC 2003, combining GP function and query expansion technique provides 29.3%, 30.2% and 22.5% improvements over our baseline system using Okapi BM25 on description field, title field and all fields, respectively.

Figure 3 shows the comparison between the performance of our submissions and the best performance achieved for Okapi BM25 in our extensive experiments where query expansion

techniques are applied. GP function has 8.4%, 8.6% and 6.8% performance improvements over Okapi BM25 on different fields of queries. However the performances shown for Okapi BM25 might not be achieved in practice. It will be explained in the next section.

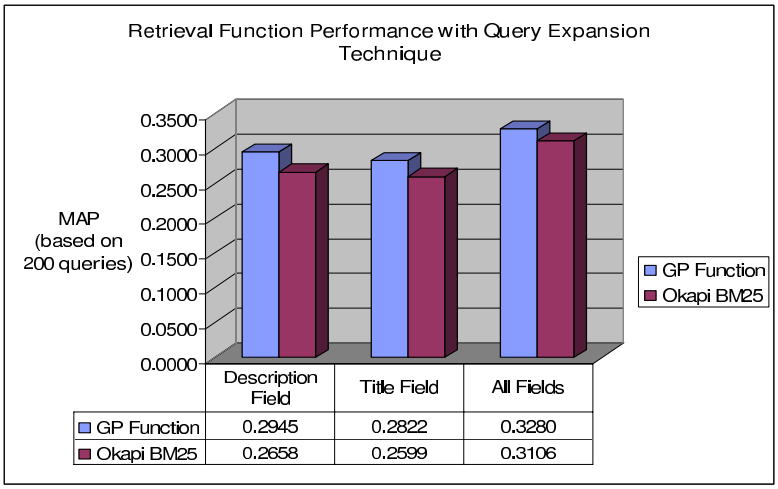
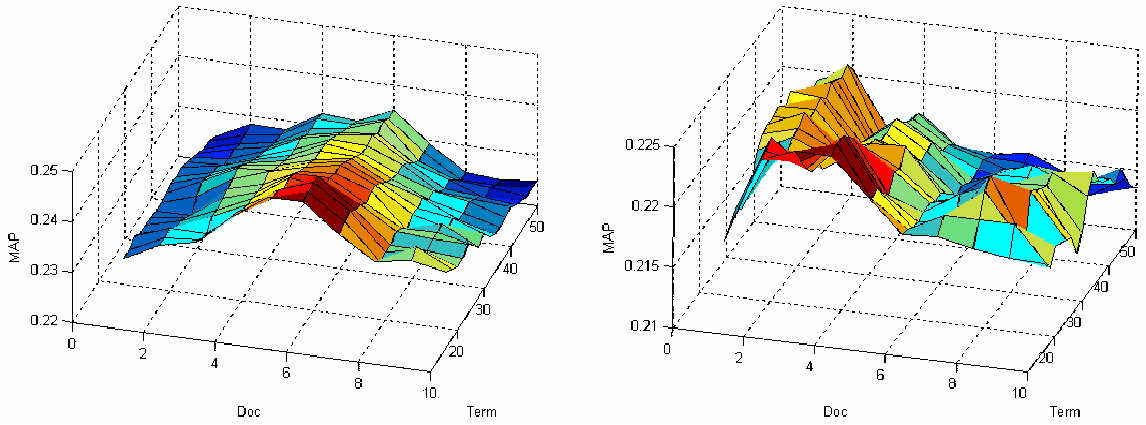


Figure 3

All of the blind feedback algorithms need two common parameters, *D* and *T*. According to our experience, the performance of these query expansion techniques is rather sensitive to the chosen values for *D* and *T*. Theoretically it is better to use an adaptive approach for choosing *D* and *T* values for each query, where a machine/statistical learning algorithm can be used to determine the best *D* and *T* combination from predictors, but we suspect the effectiveness and robustness of such a strategy in our experiments. Instead, we simply use a constant (*D*, *T*) pair within each query expansion algorithm. Since every point (*D*, *T*) corresponds to a performance measurement that the system achieves with that setup, we need to search the optimal point in this 2-dimensional space. However we can not find any theories to describe and predict the shape of such surface. It can only be learned via experiments. Since the computational cost for getting the performance measurement at a single (*D*, *T*) point is not cheap, in practice the design of experiment (DOE) approach should be used in order to avoid extensive searching and to achieve a relatively optimal solution.

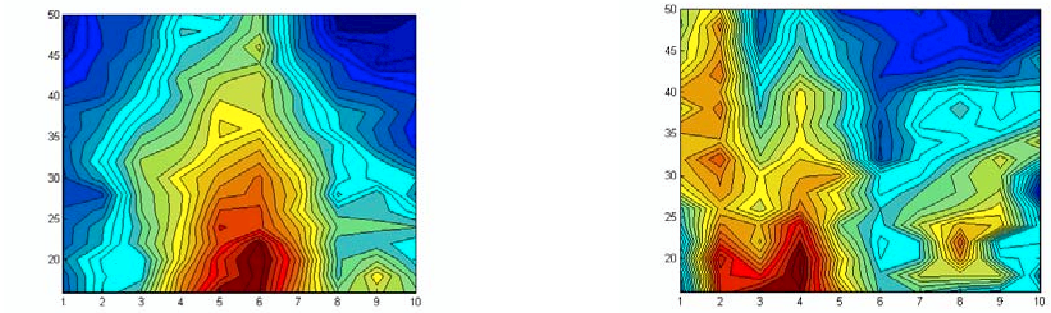
Large scale experiments are conducted in our research to learn the big picture of performance (measured as MAP) surface for each query expansion algorithm. For both GP function and Okapi BM25, we measure the MAP value at any point within the region {(*D*, *T*) | *D* = 1,2,…10; *T* = 10,11,…50}, because outside that scope the performance deteriorates for most of the query

expansion algorithms. According to the experiment results, Rocchio and Ide Dec-Hi query expansion algorithms provide much better performance on both 150 queries from TREC 6-8 and the 50 new queries from TREC 2003 than KLD, CHI and RSV do. Therefore they are used for the submissions of Robust Track, TREC 2004. Figure 4 shows the 3-D performance surfaces for GP function and Okapi BM25 combined with Dec-Hi blind feedback algorithm. Figure 5 shows the corresponding contour plots.



(a) GP function (b) Okapi BM25

Figure 4 3-D Performance Surface for GP function and Okapi BM25



(a) GP Function (b) Okapi BM25

Figure 5 Contour Plot for GP Function and Okapi BM25

From the 3-D performance surface plots, we can easily find: (1) Surfaces generated by GP function are rather smooth, while the corresponding surfaces generated by Okapi BM25 are rough. This observation suggests the property of robustness for GP function. (2) Although the surfaces of GP function are not strictly concave, it is not too far from it; however the Okapi BM25 surfaces are pretty irregular, with many local maximums. This implies that in practice it is more likely and easier to achieve the optimal setting for GP function, while using Okapi BM25 people can be trapped into local maximums. In Figure 3, it shows the performance comparison between GP function and Okapi BM25 when query expansions are applied to both functions. We take the best performance of Okapi BM25 based on our extensive experiments. However these performances can be hardly achieved in practice when large scale experiments are not affordable.

The contour plots confirm our conclusions in 3-D plots. We can find that the area of the highest performance regions for GP function is more regular and larger than area of the corresponding Okapi BM25 in contour plots. That is the indicator for robustness. In the contour plots of GP function, a global maximum is surrounded by nearly parallel contour lines; but in those of Okapi BM25, multiple local maximums exist and the contour lines are jerky. Therefore starting from an arbitrary setting, it is easier for GP function to reach the global maximum if we follow the direction suggested by maximal gradient at each step.

From our empirical study, we can conclude that when combined with blind feedback techniques: (1) GP function is more robust to parameter settings than Okapi BM25; (2) We have more chance to find the optimal parameter setting with GP function than Okapi BM25 in practice.

4. Conclusion

In this paper, we used a non-knowledge based technique to construct the retrieval function and compare its performance with the popular retrieval functions made by experts. Our retrieval function itself is proved to be more effective than any of these existing functions. The blind feedback techniques were further combined and large scale experiments were conducted to test performances of our function and Okapi BM25 under various settings. The new retrieval function discovered by GP has superior performance to Okapi BM25 when blind feedback is applied. From the empirical study of performance surfaces, we find many pleasing properties of the GP-learned function.

5. Acknowledgement

We are grateful to Ming Luo and Ye Zhou for their programming support.

6. References

[1] A. Singhal, G. Salton, M. Mitra, and C. Buckley, "Document length normalization," *Information Processing and Management*, vol. 32, pp. 619-633, 1996.

[2] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, "Okapi at TREC-4," in *the Proceedings of the Fourth Text Retrieval Conference*, D. K. Harman, Ed., 1996, pp. 73-97.

[3] V. N. Vapnik, *Statistical Learning Theory*: Wiley, 1998.

[4] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: John Wiley & Sons, 2001.

[6] W. Fan, M. Luo, L. Wang, W. Xi, and E. A. Fox, "Tuning before feedback: Combining ranking discovery and blind feedback for robust retrieval," in *the Proceedings of the27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*: ACM, 2004.

[7] W. Fan, M. D. Gordon, and P. Pathak, "Effective profiling of consumer information retrieval needs: a unified framework and empirical comparison," *Decision Support Systems*, pp. in press, 2004.