# A Hybrid Relevance-Feedback Approach to Text Retrieval

Zhao Xu[1], Xiaowei Xu[2], Kai Yu[3], and Volker Tresp[4]

[1] Tsinghua University, Beijing, P.R. China
xuzhao00@mails.tsinghua.edu.cn
[2] University of Arkansas at Little Rock, Little Rock, USA
xwxu@ualr.edu
[3] University of Munich, Munich, Germany
yu_k@dbs.informatik.uni-muenchen.de
[4] Siemens AG, Corporate Technology, Munich, Germany
volker.tresp@mchp.siemens.de

**Abstract.** Relevance feedback (RF) has been an effective query modification approach to improving the performance of information retrieval (IR) by interactively asking a user whether a set of documents are relevant or not to a given query concept. The conventional RF algorithms either converge slowly or cost a user's additional efforts in reading irrelevant documents. This paper surveys several RF algorithms and introduces a novel hybrid RF approach using a support vector machine (HRFSVM), which actively selects the uncertain documents as well as the most relevant ones on which to ask users for feedback. It can efficiently rank documents in a natural way for user browsing. We conduct experiments on Reuters-21578 dataset and track the precision as a function of feedback iterations. Experimental results have shown that HRFSVM significantly outperforms two other RF algorithms.

## 1 Introduction

The World Wide Web continues to grow at an amazing speed, as does the number of text and hypertext documents in organizational intranets. These documents represent the accumulated knowledge that becomes more and more important for an organization's success in today's information society. Search Engines (SE) became important tools in order for people to use the information on the Internet or intranets. But generally, they return a small number of relevant web pages with a large number of irrelevant web pages. Therefore, much effort has been aimed at improving the precision of SE–a challenging task due to the web's huge size, high dynamics, and large diversity.

In using a SE, users usually enter keywords that are often ambiguous and that may have different meanings in different contexts. For instance, the term "java" means "a kind of programming language" for some users, while for others it may mean "an island in Indonesia." Therefore, users may find it difficult to formalize their query concepts clearly by just using simple key words. Furthermore, because of the often long lists of results, users, who want to spend as little

time possible, may browse only the first dozen or so results. Hence, they expect powerful retrieval technology.

Traditionally an iterative and interactive process, relevance feedback (RF) improves the quality of the information retrieval [9], [10], [5]. After the initial user query, the system returns a set of ranked documents from the text base. Although the system may retrieve many documents, it only presents one screen of documents at a time. Search engines usually use screens of 10-20 documents. We assume that the abstracts of returned documents on the initial screen have enough information for the user to gauge whether a document is relevant. The user gives relevance feedback to the results on the screen. Users can give their relevance feedback by either explicitly voting or implicitly clicking documents. This way, the system learns a query concept model from the feedback and generates another list of ranked documents. This interactive process continues until the user terminates it.

The conventional RF algorithms converge slowly because users are led to label only the most relevant documents, which is usually not informative enough for systems to improve the learned query concept model. Recently, active learning algorithms have been proposed to speed up the convergence of the learning procedure [11], [12]. In active learning, the system has access to a pool of unlabelled data and can request the user's label for a certain number of instances in the pool. However, the cost of this improvement is that users must label documents when the relevance is unclear or uncertain for the system. These "uncertain documents" are also proven to be very informative for the system to improve the learned query concept model quickly.

From a machine learning point of view, both RF and active learning are two extreme instance selection schemas: the RF selects the most probably relevant instances (documents), while active learning selects the most uncertain yet informative ones. The conventional RF algorithms converge slowly, while active learning costs a user's additional efforts in reading "uncertain documents". Therefore, both methods are far from being optimal in information retrieval.

In this paper, we compare several relevance feedback algorithms for document retrieval and summarize their strengths as well as weaknesses; secondly, we introduce a novel Hybrid Relevance Feedback approach using Support Vector Machines (HRFSVM), which takes a heuristic strategy to overcome weaknesses and achieve optimal performance.

## 2 Relevance Feedback Algorithms for Document Retrieval

In the past 30 years, relevance feedback (RF) has become an effective way of modifying and expanding user queries for improving the quality of retrieval systems. Various approaches were proposed and investigated. In this section we give a survey on representative approaches. By analyzing their strengths and weaknesses, we show the reasons of why to propose a new approach.

### 2.1 The Rocchio Algorithm

One of the earliest RF algorithms was proposed by J.J. Rocchio [9]. The feedback iteration using Rocchio's algorithm expands the query in the following way:

$$Q_j = \alpha Q_{j-1} + \frac{\beta}{N_r} \sum_{i=1}^{N_r} R_i - \frac{\gamma}{N_s} \sum_{i=1}^{N_s} S_i \tag{1}$$

where, $\alpha$, $\beta$, $\gamma$ are three constants, $Q_j$ is the vector for the $j$-th updated query, $R_i$ is the vector for relevant document $i$, $S_i$ is the vector for irrelevant document $i$, $N_r$ is the number of labelled relevant documents, and $N_s$ is the number of labelled irrelevant documents.

Here $Q$, $R$, and $S$ are all represented as vectors of terms within the framework of a vector space model. All negative components of the resulting optimal query are assigned a zero weight. Once the new query $Q_j$ is obtained, we compute its inner product with each unlabelled document and form a ranking of them. The higher inner product indicates a higher relevance. The intuitive idea of Rocchio's algorithm is to iteratively increase the weights of those terms contained in labelled relevant documents while penalizing the terms in irrelevant documents. In practice, how to determine the optimal values of three constants, as in Equation (1), is always a problem.

Later, many extensions or modifications of Rocchio's RF algorithm were proposed, like Ide Regular algorithm and Ide dec-hi algorithm [10], [5]. The basic operational procedure in these methods is the merging of labeled relevant document vectors and original query vectors. Queries are automatically expanded by adding the terms not in the original query but in the labeled relevant documents. On the other hand, the term weights are increased or decreased based on whether the terms are coming from relevant or irrelevant documents. Rocchio's RF algorithm and its extensions perform well when a user gives enough feedback. No optimizing mechanism exists in them to guarantee an optimal retrieval quality in different situations, especially when few relevant or irrelevant documents are obtained from user feedback. A recent study [3] revealed that Rocchio's algorithm has a poor performance when the proportion of relevant documents in the whole corpus is quite low.

### 2.2 The SVM Relevance Feedback Algorithm (SVMRF)

In general, any classification algorithm can be applied for RF. The retrieval system iteratively obtains positive (relevant) and negative (irrelevant) instances (documents) from a user and trains a classifier at each iteration; it then uses the classifier to predict any unlabelled document as relevant or not. Due to its strong mathematic foundations and excellent empirical successes, support vector machines (SVM) [13] recently gained wide attention in the research communities of machine learning and information retrieval. Drucker, Shahraray, and Gibbon [3] applied a SVM classifier in RF and reported a much better retrieval performance, especially when just a few relevant feedbacks were obtained at the beginning iterations.

In simplest form, SVMs are hyperplanes that separate the training data by a maximal margin (see Fig. 1). All vectors lying on one side of the hyperplane are labelled as "cross" (e.g. positive), and all vectors lying on the other side are labelled as "circle". The training instances that are closest to the hyperplane are called "support" vectors. Generally, SVMs allow one to project the original training data in space $X$ to a higher dimensional feature space $F$ via an operator $K$. In other terms, we consider the set of classifiers of the form:

$$f(x) = (\sum_{i=1}^{n} \alpha K(x_i, x)) \tag{2}$$

When $K$ satisfies Mercer's condition [1], we can write:

$$K(u, v) = \Phi(u) \bullet \Phi(v) \tag{3}$$

where $\Phi : X \rightarrow F$. We can rewrite $f$ as:

$$f(x) = w \bullet \Phi(x) \tag{4}$$

where $w = \sum_{i=1}^{n} \alpha_i \Phi(x_i)$.

By choosing different kernel functions, we can implicitly project the training data from $X$ into spaces $F$ for which hyperplanes in $F$ correspond to more complex decision boundaries in the original space $X$. Commonly used kernel functions include linear kernel, polynomial kernel, and radials basis function kernel.

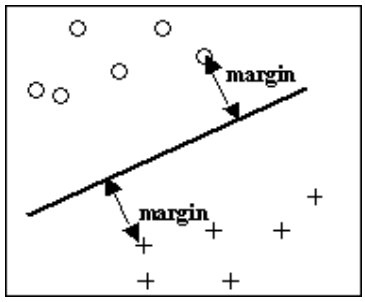



**Fig. 1.** A support vector machine with its corresponding margin and two support vectors

It has been reported that SVMs achieved notable success in the task of text classification [6], [4]. Furthermore, by calculating distances of document vectors to the trained hyperplane, SVMs give a straightforward ranking of documents, which is more desired in document retrieval than a hard class decision. A reasonable intuition is that those remote documents far away from the decision boundary can be judged relevant or irrelevant with a high confidence. Drucker et al. [3] applied this idea to document retrieval and proposed a SVM relevance feedback scheme. SVMRF proceeds in the following iterative way:

1. Users label some of documents in the top ranked list to be relevant or irrelevant according to their query concepts.
2. The system trains a new SVM model using the relevance-feedback data gathered so far.
3. The system presents a ranking of documents to users, according to the documents' distances to the newly trained hyperplane. The remotest one on the positive side is ranked at the top of the list, while the remotest one on the negative side is ranked at the bottom.
4. If the user is satisfied with current ranking, the system ends this section; otherwise, it goes to Step 1.

SVMRF was found much more effective than the Rocchio algorithm, especially in the case of searching the least frequently occurring topics [3]. This excellent performance is mainly due to the maximum margin optimization of hyperplane, which promises good generalization ability even when observed training data are quite limited. SVMRF can be viewed as an approach of passively obtaining information from a user and then learning the user's query concept (represented by a SVM model). Normally, a user gives relevance feedback in the order of ranking, e.g. rating the documents in the first screen, and thus, the system obtains feedback on those documents that are likely to be truly relevant. Normally, a learning system gets little information from the data familiar to the system, i.e. the correctly classified data. Our concern regarding this method is mainly the low learning rate, since relevance feedback on the most certainly relevant documents are not actually informative to the system for improving the model. In a machine learning community, it has been revealed that a learning system can always gain maximum information by learning from the most uncertain instances [2], [7], [12]. For the purpose of document retrieval, there should be a more effective way to actively obtain information from a user and thereby speed up the learning process. We emphasize that improving the learning rate is an essential issue in document retrieval, since people are always impatient.

### 2.3 SVM Active Learning for Relevance Feedback (ActiveSVMRF)

The idea of active learning has been widely studied in a machine learning community. Pool-based active learning was introduced by Lewis and Gate [7]. They applied a Naïve Bayesian classifier combined with logistic regression to choose the instance (a document) in which the class is most uncertain for the current classifier.

Later, two other studies [11], [12] independently investigated a similar idea of uncertainty sampling using SVM, and both applied it for document classification. In particular, Tong and Koller [12] theoretically analyzed the learning process of SVM in the framework of shrinking version space [8], and this led to three interesting and important active learning schemes: *Simple Margin*, *MaxMin Margin* and *Ratio Margin*. These methods all substantially outperform standard passive learning (e.g. random sampling) in text classification tasks. In the following, we will briefly introduce the *Simple Margin* since it has good performance and is

the simplest scheme in terms of computational cost and mathematical complexity. It was also independently proposed [7], [11]. For simplicity, we will skip the theoretical details of *Simple Margin* and give only the algorithm and intuitive explanation. (Fig. 2 presents an illustration of this algorithm.)

The key idea of *Simple Margin* is that the unlabelled vector closest to current decision boundary in $F$ is the most uncertain one and should be queried for labels, e.g. positive or negative. Under some simplifying assumptions [12], this approach leads to minimizing the version space [8]. A version space of SVM can be viewed as a space of hyperplanes that can perfectly classify labelled training instances. Imagine that in the version space there is one SVM classifier that is the target of our learning task; shrinkage of such a space means reducing our uncertainty of knowledge regarding the target classifier.

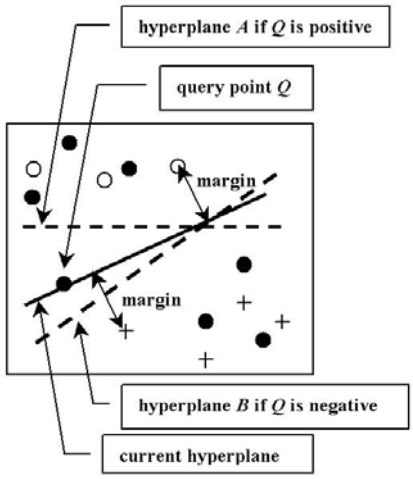



**Fig. 2.** SVM active learning algorithm - *Simple Margin* (The hollow points are negative-labelled vectors; cross points are positive vectors; and solid points, unlabelled vectors. The unlabelled vector Q is the selected query vector which is the one closest to the current hyperplane. If Q is positive, then the next trained SVM hyperplane is A, otherwise B.)

The unlabelled vector that is closest to the current boundary is the most uncertain one for classification. Intuitively, a sensible learning method should actively learn knowledge from the instances that the learning system is currently not familiar with.

In document classification tasks, a model is to be built for classifying documents in a digital library. It is always expensive for human experts to label a large number of example documents. Active learning provides a principled way to reduce the cost. However, this idea is not suitable for the tasks of document

retrieval where users search documents that are relevant to their query concepts. Our considerations are mainly based on two reasons:

1. It is unnatural to require users to rate a certain number of uncertain documents, e.g. the system has to supply one window to browse and another one to ask questions. Furthermore, most documents are likely to be irrelevant ones (see the experimental section of this paper). Users may lose patience if they read many uncertain or irrelevant documents in browsing initial screens. A classification system can explicitly require some experts to label a number of documents and then train a model. However, a retrieval system should present many relevant results to users and meanwhile gather information for further improving the ranking of unlabelled documents.
2. ActiveSVMRF, like *Simple Margin*, is actually a "myopic" algorithm. The idea is to select the only one optimal vector to label. There is no clear way to address the question of how to select the best $n$ vectors. But in practice, a document retrieval system typically presents a screen of documents to users. In many studies, researchers simply extend the conclusions of single-point optimization to multi-point optimization. For example, *Simple Margin* algorithm has been extended to select the five vectors closest to the boundary as query points [12]. In our study, we found that this extension was not effective.

## 3 A Hybrid SVM RF Approach to Document Retrieval (HRFSVM)

The following is a review of the two analyzed algorithms:

1. SVMRF: It is a natural way for getting user feedback and presents many relevant documents when gathering information from the user. However, its learning rate is not optimal since it does not select the most informative documents to label.
2. ActiveSVMRF: It is suitable for document classification rather than document retrieval. It has a fast learning rate but requires users to read many irrelevant or uncertain documents (or their abstracts). Also, it has no substantial justification on how to present several query documents.

As mentioned above, SVMRF and ActiveSVMRF each have their own strengths and weaknesses. Combinations of these may overcome their drawbacks and achieve an optimal performance for document retrieval. Therefore, we propose a hybrid approach, which presents both the most likely relevant documents and the most uncertain documents on one screen and obtains feedback about them from the user. In detail, the proposed relevance feedback approach proceeds in the following way:

1. The system learns a SVM model from a user's initial query.

2. The system then presents a screen of $M$ (e.g. 10) documents consisting of $K$ ($K \leq M$) with the remotest ones on the positive side of the SVM hyperplane and $M - K$ closest ones to the hyperplane.
3. The user returns relevance feedback to the system.
4. The system trains a new SVM model using all the obtained training data and goes to Step 2 until the user feels satisfied.

HRFSVM has a faster learning rate than SVMRF, since it integrates the mechanism of active learning into the relevance feedback process. On the other hand, compared with ActiveSVMRF, HRFSVM presents more relevant documents to users by retaining the advantages of SVMRF. While HRFSVM is going on, the trained SVM hyperplane is getting more precise to model the user's query concept; we accordingly reduce the proportion of uncertain documents on the presented screen. The reason is that at the beginning, when the SVM model is not precise enough to return many really relevant documents, we take the opportunity to gain more information about the query concept by including more uncertain documents on the screen. We found that this idea resulted in a better performance than the constant proportion of uncertain documents across iterations (see experimental section below).

## 4 Experiments

### 4.1 Experiment Description

To enable an objective evaluation of system performance and simulate the behavior of users, we assume that a query concept is a document topic. The goal of a system is to learn a given query concept through an interactive process. We assume that at each step the system returns $M = 10$ documents to a user and the user evaluates them by labelling them as "relevant" or "irrelevant," according to the query concept. Then, the system uses the total, cumulated, labelled documents so far to rebuild a new model (a SVM hyperplane), which is then used to predict all of the unlabelled documents in which the top-$K$ most relevant documents as well as the top-$(M - K)$ most uncertain documents are returned to the user. As mentioned in Section 3, we decrease the number of returned uncertain documents while iteration is going on. In our work, we empirically set $K = 6$ in the first 4 iterations and 10 in later iterations.

HRFSVM is compared with SVMRF and ActiveSVMRF. We did not consider Rocchio's RF as a competitive method since it has been demonstrated that SVMRF impressively outperforms it [3].

### 4.2 Performance Metrics

There are many ways to measure the effectiveness of the feedback process. We use *precision* since it not only indicates the accuracy of ranking but also reflects the user's satisfaction [3]. In particular, we investigate $P50$ and $P100$, the precision

of the top 50 and 100 documents respectively, which are defined in the following way:

$$PN_i = \begin{cases} \frac{n_{R_i}}{N}, n_{R_i} \leq N \\ 1, \quad \text{otherwise} \end{cases} \tag{5}$$

where $n_{R_i}$ is the total number of relevant documents in the first $N$ (50 or 100) documents at iteration $i$. $n_{R_i}$ consists of two parts: one includes the $n^1_{R_i}$ user-labelled relevant documents, and the other includes the relevant documents in the top $N - n^1_{R_i}$ unlabelled ones ranked by the system. Therefore, the adopted $P50$ and $P100$ not only show the accuracy of the predicting-model but also demonstrate how many relevant documents a user has browsed during the interactions. Furthermore, more detailed distribution of relevant documents in top 100 ones can be discovered through combining $P50$ and $P100$ together. For example, if $P50$ is larger than $P100$, it means there is more fraction of relevant documents in the first 50 ones than the later 50 ones.

### 4.3 Experimental Data Set

For evaluating the performance of the proposed approach (HRFSVM) as opposed to SVMRF and ActiveSVMRF, a test data set needs to be a set of documents labeled with topics. Therefore, we use the Reuters-21578 database (http://www.daviddlewis.com/resources/testcollections/reuters21578/), a collection of news documents that have been assigned a single topic, multiple topics, or no topic. By eliminating documents without topics, titles, or texts, we finally retain 10369 documents.

As in Drucker et al. [3], we use the visibility to indicate the percentage of a topic in the corpus:

$$visibility = \frac{n_R}{N} \tag{6}$$

where $N$ is the total number of documents in Reuters database; $n_R$ is the number of documents having the given topic.

As shown in Table 1, we select 5 topics for experiments. We will track $PN$ as a function of iteration in cases of different visibility. Since in real situations it is common that the percentage of documents meeting query concept is rather low in the documents retrieved by a search engine, we mainly investigate the low-visibility topics. Note that searching for the low-visibility topics is a more challenging task.

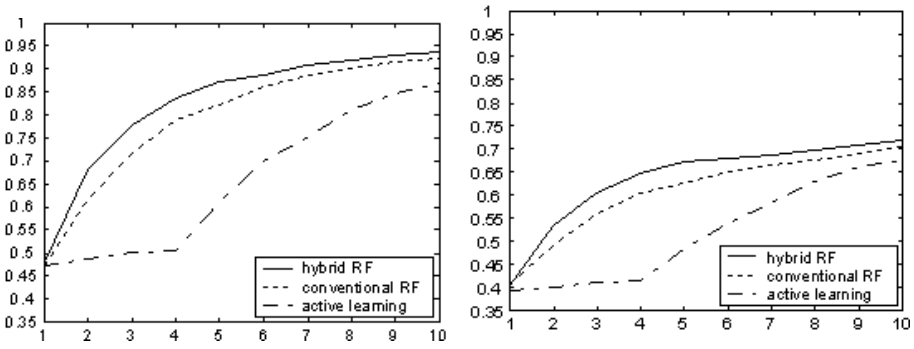
As with Drucker et al. [3], the number of returned relevant documents in the initial search is restricted to one. We assume that the system starts with one relevant document having been labelled by a user. A user has to go to subsequent screens to find a relevant document. The number of screens one has to search will depend on the sophistication of the preliminary query.

We ran 30 trials for each topic and reported the averaged precisions. Each run of the 30 experiments started with a set of random preliminary documents.

**Table 1.** Topics selected for experiments

| Topic | Number | Visibility (%) |
|---|---|---|
| Earn | 3775 | 36.4 |
| Gnp | 153 | 1.5 |
| Soybean | 111 | 1.1 |
| Iron-steel | 65 | 0.6 |
| Palm-oil | 42 | 0.4 |

### 4.4 Experiment Results

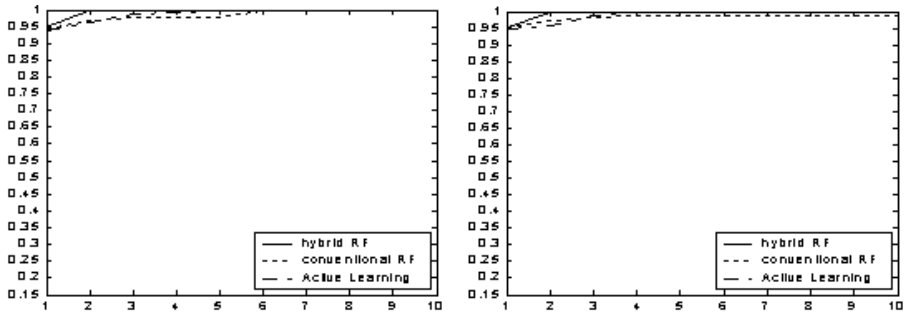



(a) $P50$ (b) $P100$

**Fig. 3.** Average value of five different visibilities versus number of iterations

In Fig. 3, each curve presents the averaged test results over the 5 topics, including 150 trials. We evaluate 3 methods in terms of $P50$ and $P100$. It has been shown that our method significantly outperforms SVMRF and ActiveSVMRF. In particular, a more impressive performance of HRFSVM is observed in initial iterations. For example, at the third iteration, HRFSVM, SVMRF, and ActiveSVMRF respectively achieve 77.6%, 71.8%, and 50.1% in the case of $P50$. It has also been observed that their differences are getting smaller, which shows all three methods get sufficient training data. We emphasize that a good performance at the early stage of relevance feedback is highly preferred by users. As shown in Fig. 3, the increase of ActiveSVMRF is slow at the beginning because a user labels few relevant documents.

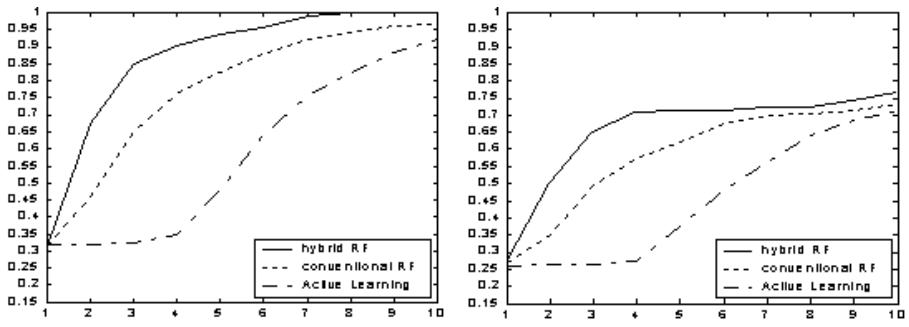
For the topics with high visibility such as *earn* with visibility 36.4% (Fig. 4), all three methods have good performance. However, our approach still outper-




(a) $P50$ (b) $P100$

**Fig. 4.** Average value of 30 experiments versus number of iterations for 36.4% visibility

forms the other two. After the first iteration, the $P50$ of HRFSVM reaches 100% three steps earlier than SVMRF.

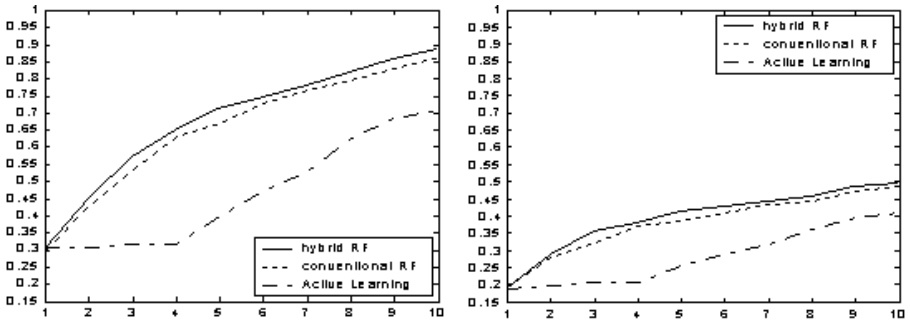



(a) $P50$ (b) $P100$

**Fig. 5.** Average value of 30 experiments versus number of iterations for 1.1% visibility

In the case of topics with small visibility such as *soybean* with visibility 1.1% (Fig. 5), our method demonstrates many advantages over SVMRF and ActiveSVMRF. After 6 iterations, proposed work reaches 98.6% with $P50$ and 72.1% with $P100$, while SVMRF does 87.8% and 67.4% and ActiveSVMRF does 64.2% and 48.3%.

For the topics with extremely low visibility such as *iron-steel* with visibility 0.6% (Fig. 6), our methods also have better performance compared with the




(a) $P50$ (b) $P100$

**Fig. 6.** Average value of 30 experiments versus number of iterations for 0.6% visibility

other two methods. The $P50$ of the proposed method reached 70% at 5th iteration, while ActiveSVMRF only does 30%. The $P100$ of HRFSVM finally reaches 49.6%. In fact, as this topic just has 65 documents, 77% of relevant documents have been found.

## 5 Conclusion

We analyzed performance of SVM-based hybrid Relevance Feedback (RF), conventional RF, and active learning. The initial relevant documents retrieved are one in every ten. We tested 5 different visibilities from 36.4% to 0.4%. The experimental results reveal that the hybrid RF outperforms the other two methods regardless of visibilities. Our method efficiently satisfies a user in a natural browsing way by mixing the most uncertain documents and the most relevant ones together to ask for user labelling.

Now the method is a heuristic. In the future, we will improve it by using optimization technology.

## References

1. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, (2):121–167, 1998.
2. D. Cohn and Z. Ghahramani. Active learning with statistical models. *Journal of Artificial Intelligence Research*, (4):129–145, 1996.
3. H. Drucker, B. Shahraray, and D. Gibbon. Relevance feedback using support vector machines. In *Proceedings of the 18th International Conference on Machine Learning*, pages 122–129, 2001.
4. S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the Seventh International Conference on Information and Knowledge Management*. ACM Press, 1998.

5. D. Harman. Relevance feedback revisited. In *Proceedings of the Fifth International SIGIR Conference on Research and Development in Information Retrieval*, pages 1–10, 1992.
6. T. Joachims. Text categorization with support vector machines. In *Proceedings of the European Conference on Machine Learning*. Springer Verlag, 1998.
7. D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1994.
8. T. Mitchell. Generalization as search. *Artificial Intelligence*, (28):203–226, 1982.
9. J. J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.
10. G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society of Information Science*, 41:288–297, 1990.
11. G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.
12. S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, (2):45–66, 2001.
13. V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer Verlag, 1982.