Interactive Document Retrieval with Active Learning

Takashi ONODA, Hiroshi MURATA
Central Research Institute of Electric Power Industry
2-11-1 Iwato-kita, Komae, Tokyo 201-8511, JAPAN
E-mail: {onoda, murata}@criepi.denken.or.jp

Seiji YAMADA
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda, Tokyo 101-8430, JAPAN
E-mail: seiji@nii.ac.jp

Abstract

We investigate the following data mining problem from Information Retrieval: From a large data set of documents, we need to find those that bind to human interesting in as few iterations of human testing or checking as possible. In each iteration a comparatively small batch of documents is screened for binding the human interesting. We apply active learning techniques for selecting successive batches.

1 Introduction

As progression of the internet technology, accessible information by end users is explosively increasing. In this situation, we can now easily accsess a huge document database through the WWW. However it is hard for a user to retrieve relevant documents from which he/she can obtain useful information, and a lot of studies have been done in IR(Information Retrieval), especially document retrieval[19]. Active works for such document retrieval have been reported in TREC(Text Retrieval Conference)[16] for English documents, IREX(Information Retrieval and Extraction Exercise)[4] and NTCIR(NII-NACSIS Test Collection for IR System)[7] for Japanese documents.

In most frameworks for information retrieval, a vector space model in which a document is described with a high-dimensional vector is used[12]. An IR system using a vector space model computes the similarity between a query vector and document vectors by cosine of the two vectors and indicates a user a list of retrieved documents.

In general, since a user hardly describes a precise query in the first trial, interactive approach to modify the query vector by evaluation of the user on documents in a list of retrieved documents. This method is called *relevance feedback*[11] and used widely in IR systems. In this method, a user directly evaluates whether a document is relevant or no-relevant in a list of retrieved documents, and a system modifies the query vector using the user evaluation. A traditional way to modify a query vector is a simple learning rule to reduce the difference between the query vector and documents evaluated as relevant by a user.

Another approach has been proposed that classification learning with relevant and no-relevant document vectors as positive and negative examples for a target concept[8]. Some studies proposed SVM(Support Vector Machine) with excellent ability to classify examples into two classes is applied to classification learning of relevance feedback[15][3].

We propose a relevance feedback framework with SVM as *active learning*. In contrast that a conventional relevance feedback system indicates a user a list of the most relevant documents, our system provides a user a list of documents which are hard for SVM to classify them. This is a kind of active learning approach and we consider it promising for relevance feedback.

Okabe and Yamada[8] proposed a frame work in which relational learning to classification rules was applied to interactive document retrieval. Since the learned classification rules is described with symbolic representation, they are readable to our human and we can easily modify the rules directly using a sort of editor. However we consider SVM dealing with continuous values can do more precise classification than symbolic classification rules.

The relevance feedback is similar to what is termed active learning in that we try to maximize test performance using the smallest number of documents in the training set[15]. In active learning, we are interested in maximizing learning performance. Drucker et al.

applied SVM to the relevance feedback[3]. They are interested in maximizing the number of relevant documents which are displayed to users at each feedback iteration. However, we are interested in satisfying both the aim of active learning and the aim of increasing the number of relevant documents which are displayed to users at each feedback iteration. The detail of this difference will be described in the third section.

In the remaining parts of this paper, we explain a SVM algorithm in the second section briefly, and an active learning with SVM for the relevance feedback in the third section. In the fourth section, in order to evaluate the effectiveness of our approach, we made experiments using a TREC data set of Las Angels Times and discuss the experimental results. Eventually we conclude our work and discuss open problem in the fifth section.

2 Support Vector Machines

Formally, the Support Vector Machine (SVM) [17] like any other classification method aims to estimate a classification function $f : \mathcal{X} \to \{\pm 1\}$ using labeled training data from $\mathcal{X} \times \{\pm 1\}$. Moreover this function f should even classify unseen examples correctly.

In order to construct good classifiers by learning, two conditions have to be respected. First, the training data must be an unbiased sample from the same source (pdf) as the unseen test data. This concerns the experimental setup. Second, the size of the class of functions from which we choose our estimate f, the so-called capacity of the learning machine, has to be properly restricted according to statistical learning theory [17]. If the capacity is too small, complex discriminant functions cannot be approximated sufficiently well by any selectable function f in the chosen class of functions – the learning machine is too simple to learn well. On the other hand, if the capacity is too large, the learning machine bears the risk of overfitting.

In neural network training, overfitting is avoided by early stopping, regularization or asymptotic model selection [1, 6, 9, 10].

For SV learning machines that implement linear discriminant functions in feature spaces, the capacity limitation corresponds to finding a large margin separation between the two classes. The margin ϱ is the minimal distance of training points $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_i, y_i), \mathbf{x}_i \in \mathbf{R}, y_i \in \{\pm 1\}$ to the separation surface, i.e.

$$\varrho = \min_{i=1,\ldots,\ell} \rho(\mathbf{z}_i, f) \tag{1}$$

where $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ and

$$\rho(\mathbf{z}_i, f) = y_i f(\mathbf{x}_i), \tag{2}$$

and f is the linear discriminant function in some feature space

$$f(\mathbf{x}) = (\mathbf{w} \cdot \Phi(\mathbf{x})) + b = \sum_{i=1}^{\ell} \alpha_i y_i (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})) + b, \tag{3}$$

with $\mathbf{w}$ expressed as $\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \Phi(\mathbf{x}_i)$. The quantity Φ denotes the mapping from input space $\mathcal{X}$ by explicitly transforming the data into a feature space $\mathcal{F}$ using $\Phi : \mathcal{X} \to \mathcal{F}$. (see Figure 1). SVM can do so implicitly. In order to train and classify, all that SVMs use are dot products of pairs of data points $\Phi(\mathbf{x}), \Phi(\mathbf{x}_i) \in \mathcal{F}$ in feature space (cf. Eq. (3)). Thus, we need only to supply a so-called kernel function that can compute these dot products. A kernel function k allows to implicitly define the feature space (Mercer's Theorem, e.g. [2]) via

$$k(\mathbf{x}, \mathbf{x}_i) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_i)). \tag{4}$$

By using different kernel functions, the SVM algorithm can construct a variety of learning machines, some of which coincide with classical architectures:

Polynomial classifiers of degree d:

$$k(\mathbf{x}, \mathbf{x}_i) = (\kappa \cdot (\mathbf{x} \cdot \mathbf{x}_i) + \Theta)^d \tag{5}$$

Neural networks(sigmoidal):

$$k(\mathbf{x}, \mathbf{x}_i) = \tanh(\kappa \cdot (\mathbf{x} \cdot \mathbf{x}_i) + \Theta) \tag{6}$$

Radial basis function classifiers:

$$k(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{\sigma}\right) \tag{7}$$

Note that there is no need to use or know the form of Φ, because the mapping is never performed explicitly The introduction of Φ in the explanation above was for purely didactical and not algorithmical purposes. Therefore, we can computationally afford to work in implicitly very large (e.g. 10^{10}- dimensional) feature spaces. SVM can avoid overfitting by controlling the capacity and maximizing the margin. Simultaneously, SVMs learn which of the features implied by the kernel k are distinctive for the two classes, i.e. instead of finding well-suited features by ourselves (which can often be difficult), we can use the SVM to select them from an extremely rich feature space.

With respect to good generalization, it is often profitable to misclassify some outlying training data points

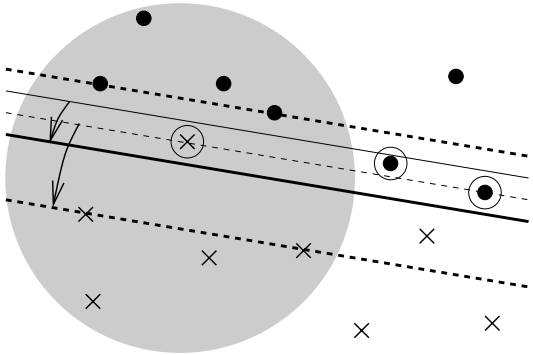

Figure 1: A binary classification toy problem: This problem is to separate black circles from crosses. The shaded region consists of training examples, the other regions of test data. The data can be separated with a margin indicated by the slim dashed lines, implicating the slim solid line as discriminate function. Misclassifying one training example(a circled white circle) leads to a considerable extension(arrows) of the margin(fat dashed and solid lines) and this fat solid line can classify two test examples(circled black circles) correctly.

in order to achieve a larger margin between the other training points (see Figure 1 for an example).

This soft-margin strategy can also learn non-separable data. The trade-off between margin size and number of misclassified training points is then controlled by the regularization parameter C (softness of the margin). The following quadratic program (QP) (see e.g. [17, 14]):

$$\begin{array}{lrll} \min & \|\mathbf{w}\|^2 + C\sum_{i=1}^{\ell}\xi_i & & \\ \text{s.t.} & \rho(\mathbf{z}_i, \boldsymbol{\alpha}) \geq 1 - \xi_i & \text{for all } 1 \leq i \leq \ell & (8) \\ & \xi_i, \alpha_i \geq 0 & \text{for all } 1 \leq i \leq \ell & \end{array}$$

leads to the SV soft-margin solution allowing for some errors.

3 Active learning with SVM in IR

In this section, we describe the information retrieval system using relevance feedback with SVM from an active learning point of view. In relevance feedback, the user has the option of labeling some of the top ranked documents according to whether they are relevant or not. The labeled documents along with the original request are then given to a supervised learning procedure to produce a new classifier. The new classifier is used to produce a new ranking, which retrievals more relevant documents at higher ranks than the original did. In relevant feedback method, the user have to judge the feedback documents. Hence, it is difficult to use a large number of user judged documents for supervised learning procedure because the user needs much effort to judged the documents. The SVMs have a great ability to discriminate even if the training data is small. Consequently, we propose to apply SVMs as the classifier in relevance feedback method. The retrieval steps of proposed method perform as follows:

Step 1: Preparation of documents for the first feedback
The conventional information retrieval system based on vector space model displays the top N ranked documents along with a request query to the user.

Step 2: Judgement of documents
The user then classifiers these documents into relevant or irrelevant. The relevant documents and the irrelevant documents are labeled.

Step 3: Determination of the optimal hyperplane
The optimal hyperplane is determined by using SVM which is learned by labeled documents.

Step 4: Discrimination all test collection and information retrieval
The SVM learned by previous step classifies the whole documents as relevant or not. The documents which are discriminated relevant and in the margin area of SVM are shown to user as the information retrieval results of the system. If the number of feedback iterations is more than m, then go to next step. Otherwise return to Step 2. The m is a maximal number of feedback iterations.

Step 5: Display of the final retrieved documents
The retrieved documents are ranked by the distance between the documents and the hyperplane which is the discriminant function determined by SVM. The retrieved documents are displayed based on this ranking.

The feature of our SVM-feedback is the selection of displayed documents to users in Step 4. Our proposed method select the documents which are discriminated relevant and in the margin area. In the reference [3], Drucker selects the higher ranked documents which are relevant and far from the discriminant function. This selection can not keep efficient learning from an active learning poin of view. In the reference [15], Tong selects the documents which are on or near the

discriminant function. This selection can make efficient learning. However, users feel stress of the selection because it is difficult to display the relevant documents by the selection. Our selection can be expected that the efficient learning can be kept and users do not need to feel stress.

4 Experiments

4.1 Experimental settings

We made experiments for evaluating the utility of our interactive document retrieval with active learning of SVM in §3. The document data set we used is a set of articles in the Los Angels Times(about 130 thousands articles, the average number of words in a article is 526) which is widely used in the document retrieval conference TREC[16]. This data set includes not only queries but also the relevant documents to each query. Thus we used the queries for experiments.

We used TFIDF[19], which is one of the most popular methods in IR to generate document feature vectors, and the concrete equations[13] are in the following.

$$
\begin{aligned}
w_t^d &= L * t * u \\
L &= \frac{1+\log(tf(t,d))}{1+log(\text{average of } tf(t,d)ind)} \quad (tf) \\
t &= \log(\frac{N+1}{df(t)}) \quad (idf) \\
u &= \frac{1}{0.8+0.2\frac{uniq(d)}{\text{average of } uniq(d)}} \quad (normalization)
\end{aligned}
$$

- w_t^d : Weight of a term t in a document d.
- $tf(t,d)$: Frequency of a term t in a document d.
- N : Total number of documents in a data set.
- $df(t)$: The number of documents including a term t.
- $uniq(d)$: The number of different terms in a document d.

The size N of retrieved results developed in **Step 1** in §3 was set as 20. The feedback iterations was 1, 2, 3 and 4. In order to investigate the influence of feedback iterations on accuracy of retrieval, we used plural feedback iterations.

In our experiments, we used the linear discriminant function for the classifier in SVM. The VSM of documents is high dimensional space. Therefore, in order to classify the labeled documents into relevant or irrelevant, we do not need to use the kernel trick and the the regularization parameter C(see §2). The VSM consists of TFIDF. Drucker et al. did not use TFIDF representation for SVM learning[3].

For comparison with our approach, two IR system were used. The first is a IR system that does not use feedback. The second is a IR system using traditional Rocchio-based releveance feedback[11] which is widely used in IR.

The Rocchio-based releveance feedback modifies a query vector Q_i by evaluation of a user using the following equation.

$$
Q_{i+1} = Q_i + \alpha \sum_{x \in R_r} x - \beta \sum_{x \in R_n} x, \qquad (9)
$$

where R_r is a set of documents which were evaluated as relevant by a user at the $i$the feedback, and R_n is a set of documents which were evaluated as no-relevant in the i feedback. α, β are weights for relevant, no-relevant documents respectively. we set $\alpha = 1.0$, $\beta = 0.5$ which are known adequate experimentally.

In general, retrieval accuracy significantly depends on the feedback iterations. Thus we changed feedback iterations for 1, 2, 3, 4 and investigated the accuracy for each iterations.

We utilized *precision* and *recall* for evaluating the three IR systems[5][18]. The following equations are used to compute precision and recall. Since a recall-precision curve is investigated to each query, we used the average recall-precision curve over all the queries as evaluation.

$$
\begin{aligned}
\text{Precision} &= \frac{\text{The No. of retrieved relevant doc.}}{\text{The No. of retrieved doc.}} \\
\text{recall} &= \frac{\text{The No. of retrieved relevant doc.}}{\text{The total No. of relevant doc.}}
\end{aligned}
$$

4.2 Experimental results

4.2.1 Comparing of recall-precision performance curve

In this section, we investigated the effectiveness of proposed method, when the user judged the top of twenty ranked documents at each feedback iteration. In the first iteration, twenty ranked documents were retrieved by VSM, which is represened by TFIDF.

Figure 2 show a recall-precision performance curve of SVM-based method, after four feedback iterations. For comparison, this figure also show the results of the conventional feedback method (i.e., Rocchio-based method) and VSM (i.e., without feedback). The thick solid line is the proposed method, the broken line is

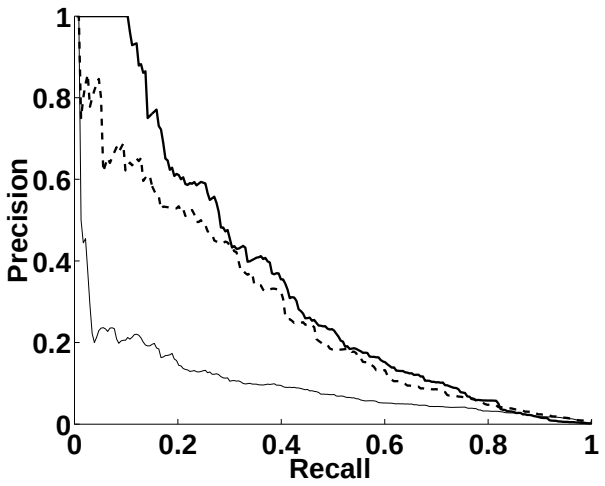

Figure 2: The effectiveness of SVM-feedback: The lines show recall-precision performance curve by using 20 feedback documents on the set of articles in the Los Angels after 4 feedback iterations. The wide solid line is proposed method, the broken line is conventional feedback method (i.e. Rocchio-based method), and the solid line is the VSM without feedback.

the conventional feedback method, and the thin solid line was the VSM without feedback.

This figure show that the retrieval effectiveness of both feedback methods, i.e., proposed and conventional feedback method, is improved compared with that of the non-feedback. In this result, we could confirm that the relevance feedback was useful technique for improving the performance of VSM.

Furthermore, this figure also show that the proposed feedback method improves the performance compared with conventional feedback method at all recall points. Consequently, we conclude that the SVMs are useful relevant feedback technique improving performance of VSM in this experiment.

4.2.2 Relationships between the performance and the number of feedback iterations

Here, we describe the relationships between the performances of proposed method and the number of feedback iterations. Table 1 gave the average precision result as a function of the number of feedback iterations. At each feedback iteration, the system displays twenty ranked relevant documents. We also show the average precision of Rocchio-based method for comparing to proposed method in table 1.

We can see from this table that the SVM feedback gives the higher performance in proportion to increase feedback iterations. On the other hand, the Rocchio-based feedback method degrades the retrieval performances nevertheless the number of feedback iterations increased from three to four. Hence, we can consider that the more feedback iterations, the better relevance documents they can obtain by using SVM-feedback method. Especially, the proposed method can improve the performance of conventional feedback method at each feedback iteration. We believe that the reason of these results was that the SVM can find a more suitable hyperplane for discriminating between relevant and irrelevant documents as increasing the feedback iterations. After all, we can believe that the proposed method can keep effective learning from active learning point of view.

Table 1: Average precision using SVM-feedback and Rocchio-feedback

No. of feedback iterations	Average precision	
	SVM	Rocchio
1	0.2625	0.2250
2	0.3500	0.2500
3	0.6125	0.2350
4	0.6375	0.2250

Furthermore, we compare the performances of proposed method to those of of Rocchiobased feedback method from an IR point of view. Table 2 shows the relationship between no. of feedback iterations and no. of actual relevant documents in 20 higher ranked relevant documents in a special case. In this special case, five documents were labeled as relevant documents in twenty documents at the first iteration. In almost case, one or two documents were labeled as relevant documents in twenty documents at the first iteration. We can see from this table that our SVM feedback can increase the number of actual relevant documents which are displayed to the user in proportion to increase feedback iterations. On the other hand, the Rocchio-based feedback method degrades the number of actual relevant documents nevertheless the number of feedback iterations increased from three to four. Hence, we can consider that the proposed method can give the suitable number of actual relevant documents to the user.

5 Conclusion

In this paper, we proposed the relevance feedback method with support vector machines (SVMs) for the information retrieval. Because the SVMs have an excellent ability to discriminate even if the training data is small, we applied the SVMs to relevance feedback

Table 2: In a special case, the relationship between no. of feedback iterations and no. of actual relevant documents in 20 higher ranked relevant documents.

No. of feedback iterations	No. of relevant documents	
	SVM	Rocchio
1	9	11
2	18	13
3	20	12
4	20	11

method. Experimental results on a set of articles in the Los Angels Times showed the proposed method gave a consistently better performance than the conventional feedback method. In our experiments we use TFIDF documents representation as VSM. Drucker et al. use binary documents representation and TF representation for estimating the performance of their proposed method. We plan to apply our proposed method to the binary representation and TF representation and compare our method with other SVM based methods(Drucker's method and Tong's method) experimentally.

In this paper, we proposed that the system should display the documents which are discriminated relevant and in the margin area of SVM at each feedback iteration. However, we do not discuss how the selection of documents influence both the effective learning and the performance of information retrieval theoretically. We would like to propose it as an open problem.

References

[1] C.M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[2] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *5th Annual ACM Workshop on COLT*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.

[3] Harris Drucker, Behzad Shahrary, and David C. Gibbon. Relevance feedback using support vector machines. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 122–129, 2001.

[4] IREX. http://cs.nyu.edu/cs/projects/ proteus/irex/.

[5] D. Lewis. Evaluating text categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 312–318, 1991.

[6] N. Murata, S. Yoshizawa, and S. Amari. Network information criterion - determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks*, 5(6):865–872, 1994.

[7] NTCIR. http://www.rd.nacsis.ac.jp/˜ntcadm/.

[8] M. Okabe and S. Yamada. Interactive document retrieval with relational learning. In *Proceedings of the 16th ACM Symposium on Applied Computing*, pages 27–31, 2001.

[9] T. Onoda. Neural network information criterion for the optimal number of hidden units. In *Proc. ICNN'95*, volume 1, pages 275–280, 1995.

[10] J. Orr and K.-R. Müller, editors. *Neural Networks: Tricks of the Trade*. LNCS 1524, Springer Verlag, 1998.

[11] G. Salton, editor. *Relevance feedback in information retrieval*, pages 313–323. Englewood Cliffs, N.J.: Prentice Hall, 1971.

[12] G. Salton and J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[13] R.E. Schapire, Y. Singer, and A. Singhal. Boosting and rocchio applied to text filtering. In *Proceedings of the Twenty-First Annual International ACM SIGIR*, pages 215–223, 1998.

[14] B. Schölkopf, A. Smola, R. Williamson, and P.L. Bartlett. New support vector algorithms. Technical Report NC-TR-1998-031, Department of Computer Science, Royal Holloway, University of London, Egham, UK, 1998. *Neural Computation 2000*.

[15] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *Journal of Machine Learning Research*, volume 2, pages 45–66, 2001.

[16] TREC Web page. http://trec.nist.gov/.

[17] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[18] I. Witten, A. Moffat, and T. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Van Nostrand Reinhold, New York, 1994.

[19] R. B. Yates and B. R. Neto. *Modern Information Retrieval*. Addison Wesley, 1999.