Using the $\alpha\beta$-Neighborhood for Adaptive Document Filtering

Adrian Fonseca-Bruzón, Reynaldo Gil-García, and Aurora Pons-Porrata

Center for Pattern Recognition and Data Mining
Universidad de Oriente, Santiago de Cuba, Cuba
{adrian,gil,aurora}@cerpamid.co.cu

Abstract. In this paper, we address the problem of adaptive document filtering. Traditionally, user profiles are represented by the centroid of the available examples, assuming that these are homogeneously distributed around this centroid. However, these examples may be irregularly distributed, being some areas more populated than others. While, in this case, the homogeneity assumption may not be globally true, it may still hold locally. In order to handle this phenomenon, we introduce a new approach in which a binary classifier for each user profile is used and more than one document is considered in the classification task. To decide whether a new document is relevant to the user or not, our approach uses a Nearest Neighbor classifier based on a neighborhood which inspects a sufficiently small area surrounding the new document. Experiments carried out on the TREC-11 collection show the effectiveness of the proposed method.

Keywords: adaptive filtering, nearest neighbor classifier.

1 Introduction

Information filtering systems monitor through an information stream to find the documents that satisfy the information need of a user. These systems keep a *profile* for each user representing this information need. For each incoming document, the system must make a binary decision, namely, to accept or reject the document. An information filtering system is said to be *adaptive* if it receives periodical feedback from the user indicating whether a delivered document is relevant or not. This feedback provides the system with training examples for online learning. Common information filtering systems require a considerable number of documents in the training set in order to build a classifier. However, an adaptive filtering system is expected to be able to start classification using a very small number of training examples, and to increase its knowledge based on the feedback received from the user.

Several approaches have been reported in the adaptive filtering literature, including the Rocchio algorithm, logistic regression and support vector machines. Commonly, the systems represent the user profile by a single vector assuming that all documents in the profile are homogenously distributed around this

J. Ruiz-Shulcloper and W.G. Kropatsch (Eds.): CIARP 2008, LNCS 5197, pp. 783–790, 2008.

vector. In our opinion, such systems may fail to capture the situation where documents satisfying a user's information need are irregularly distributed.

In order to tackle this problem, we present a new adaptive filtering approach where profiles are represented using a set of documents which is updated according to the feedback provided for newly retrieved documents. This allows us to model the true distribution of documents in the profile. Classification is carried out using a Nearest Neighbor (NN) rule over this set of documents. This rule is based on the $\alpha\beta$-neighborhood [1], in which the number of neighbors is not fixed and documents whose similarity to the new document is too low are ignored. We also define a voting scheme and a decision rule to use the NN classifier in the adaptive filtering problem.

The proposed approach is evaluated on the TREC-11 benchmark collection. We carry out a set of experiments to compare its performance to that of several systems over this collection. Experimental results show that our method performs better than other approaches when profiles contain internal divergences, while achieves comparable results for profiles which are close to be homogeneous.

The remainder of the paper is organized as follows: in Section 2 we describe some related approaches. Section 3 presents the adaptive filtering system using the NN classifier based on the $\alpha\beta$-neighborhood. Experimental results and a comparison with existing algorithms are shown in Section 4. Finally, conclusions are presented in Section 5.

2 Related Work

According to Zhang [2], there are two main types of approaches to adaptive document filtering: Retrieval + thresholding and text classification.

In the first type of approaches, systems build an adaptive filtering strategy using on algorithms originally designed for Information Retrieval (IR). These systems use an IR algorithm to score each incoming document and deliver as relevant those for which the score is higher than a threshold. A number of scoring algorithms and threshold updating methods have been explored. For instance, Yang and Kisiel [3] propose a margin-based local regression algorithm for updating threshold and apply it to the Rocchio algorithm. Tebri et. al. [4] propose a system based on the Rocchio algorithm and use a reinforcement learning algorithm for updating the weights in the vector profile.

Another type of approaches consists on systems that treat filtering as a text classification task by defining two classes: *Relevant* and *Non-Relevant*. The filtering system learns a user profile as a classifier and delivers a document if it assigned to the class *Relevant*. Several text classification algorithms are used to solve this binary classification task, such as Support Vector Machines (SVM)[5,6,7], logistic regression [8], neural networks [9] and k Nearest Neighbors (k-NN) [10,12]. Ault and Yang [10] use the k-NN algorithm considering each user profile as a class. In their work, for each new document, its neighbors are globally calculated over the documents in all profiles. Then it is delivered for all profiles represented by classes for which the relevance score assigned by

the algorithm is greater than a threshold. In this case, profiles may be seen as competing for their examples to be included among the k selected neighbors. In real environments, no relations may be assumed between profiles, therefore interactions between actions taken for different profiles should be avoided. Our method follows this second type of approach. In our case, a profile is represented by a set of documents. For determining if a new document is relevant or not for each profile, we use a version of the NN classifier.

3 Our Approach

Several algorithms represent each user profile using a single vector calculated from the centroids of the Relevant and Non-Relevant classes, assuming that all documents in each class are homogeneusly distributed around its centroid. This assumption in the general case is not necessarily true. In a profile, differences may exist between documents that are relevant to it. These differences may be the result of an inherent clustering structure of these documents. This situation is likely to occur when documents that satisfy the user's information need come from different sources, when this information need changes over time or when the user's judgments are not completely reliable. We tackle this situation by using a classifier that takes into account a set of documents for representing the user profile in order to enable the system to learn the distribution of the documents in the profile. The purpose of representing a profile by means of a set of documents is to allow the method to take into account the diversity between documents in the profile.

As mentioned before, in adaptive filtering each incoming document is classified into one of two classes: Relevant or Non-Relevant. In this work, the classification is carried out using a NN classifier based on the $\alpha\beta$-neighborhood [1]. This is a version of the well known NN rule [13]. To classify a new document d, the algorithm follows three steps.

First, it determines the $\alpha\beta$-neighborhood of d. This method inspects a sufficiently small and near area to d. In contrast to k-NN algorithm, the number of neighbors is not fixed, but rather the neighborhood radius is automatically adjusted from the nearest neighbor of d. This radius is the addition of the similarity between d and its nearest neighbor, and the threshold α. This neighborhood encloses all documents within a spherical region defined from the nearest neighbor. Candidates whose similarity to d is less than β are discarded. This neighborhood is illustrated in Figure 1. In the figure, the shaded region represents the neighborhood of d and the white point represents its nearest neighbor. Notice that the parameters α and β provide a convenient way of obtaining such a neighborhood.

The second step consists on a voting process. We define the vote for each class as the sum of the similarity values between d and the class documents belonging to the $\alpha\beta$-neighborhood of d, i.e.,

$$V(c) = \sum_{d_j \epsilon N(c)} sim(d, d_j)$$

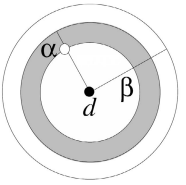


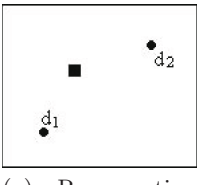
Fig. 1. $\alpha\beta$-neighborhood

where c is the class (Relevant or Non-Relevant), $N(c)$ is the set of the nearest neighbors belonging to the class c and $sim(d,d_j)$ is the similarity between the two documents.

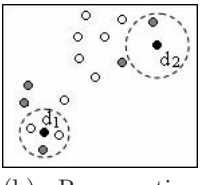
Finally, the document d is delivered to the user if

$$V(Relevant) > V(Non - Relevant)$$

When a document is delivered to the user, the system receives feedback from the user, if it is available. According to this information, the document is added either to the Relevant or to the Non-Relevant set. Note that the voting scheme and decision rule are different to those used in [1].



(a) Representing a profile with a single vector



(b) Representing the profile with a set of documents

Fig. 2. The effect of using a set of documents for representing a profile

We use the $\alpha\beta$-neighborhood, because it is homogeneously distributed around the new document, excluding candidates that are not similar enough. Thus, when a new document is being classified, the algorithm will take into account only those documents that are very similar to the new document.

Figure 2 illustrates this situation. In the figure, d_1 and d_2 represents two possible documents to be classified. In 2 (a) the profile is represented by a single vector (black square in the figure) as happens, for example, in the Rocchio approach, and d_1 and d_2 are equidistant from this vector. In this case, both documents are treated exactly in the same way, either as relevants or as non-relevants. On the other hand, in 2 (b) the same profile is represented by a set of documents. White dots represent the relevant documents whereas gray dots represent the non-relevant ones. In this case, d_1 and d_2 are classified taking into account only those documents that are close to it. The circles around d_1 and d_2

represent their neighborhoods. Thus, d_1 will be classified as relevant document whereas d_2 will be classified as non-relevant one.

To sum up, the steps of our approach is shown in Algorithm 1.

Algorithm 1. Determining the profiles for which an incoming document d is relevant.

1. For each profile P:
   (a) Build $N_\beta = \{d_j \mid sim(d, d_j) \geq \beta\}$
   (b) if N_β is empty :
      i. Reject d for the profile P
   (c) else
      i. Let max be the similarity between d and its nearest neirghbor
      ii. Let $N = \{d_j \mid d_j \epsilon N_\beta \; and \; sim\,(d, d_j) \geq max - \alpha\}$
      iii. V(Relevant) $= \sum_{d_j \epsilon N \cap R_P} sim\,(d, d_j)$
      iv. V(Non-Relevant) $= \sum_{d_j \epsilon N \cap NR_P} sim\,(d, d_j)$
      v. if $V(Relevant) > V(Non - Relevant)$, deliver d to the user for the profile P; reject it otherwise

In it, R_P and NR_P denotes the set of relevant documents and non-relevant documents for the profile P, respectively. It is worth mentioning that at the beginning, the algorithm requires some relevant examples for each profile. However, the algorithm can start with no non-relevant examples.

4 Experimental Results

The aim of these experiments is to evaluate the efectiveness of the proposed adaptive filtering approach. They are undertaken following the evaluation scheme of the TREC-11 competition.

4.1 TREC-11 Dataset

TREC-11 dataset is a subset of the RCV1 corpus. This collection includes about 800000 news stories. It covers a time period from 1996 to 1997. A set of 100 topics was defined over this corpus for the Filtering Track [14] in the TREC-11 competition. The first fifty topics (R101 - R150) were constructed by human assessors. The relevance judgements for these topics were collected using extensive searches with several rounds of multiple retrieval systems after an initial definition of the topics. The remaining fifty topics (R151 - R200) were constructed as intersections of pairs of Reuters categories. In these topics, documents belonging to both categories were considered as relevant. Non-relevant documents were chosen randomly from those assigned either of the categories, but not both.

For each topic only three relevant documents are available for building the initial user profile, and the filtering systems may used the relevance judgement from any document that has been retrieved.

4.2 Evaluation Measures

A linear utility function is used to evaluate a filtering system. A general form of the linear utility function used in the TREC-11 Filtering task is shown below

$$T11U = 2 * R^{+} - N^{+}$$

where R^{+} represents the number of relevant documents delivered by the system and N^{+} represents the number of non-relevant documents delivered by the system. A normalized version of T11U was also used in TREC-11:

$$T11SU = \frac{max\left(\frac{T11U}{MaxU}, MinNU\right) - MinNU}{1 - MinNU}$$

where $MaxU = 2*(R^{+}+R^{-})$ is the maximum possible utility, R^{-} represents the number of relevant documents not delivered by the system and $MinNU = -0.5$. If the system retrieves no documents, this measure takes the value 0.33, which was considered as baseline in the TREC-11 competition.

4.3 Results

Experiments are conducted to compare the performance of our approach ($\alpha\beta$-neighborhood for adaptive filtering) against the two systems that obtained the best results in the TREC-11 Filtering Track: ICT [11] and KerMIT [6], and a later system (Reinforcement) which implements profile learning based on a reinforcement method proposed by Tebri et. al. [4]. The results of these systems are taken from [11,6,4]. We also compare our approach against the traditional k-NN classifier using the same voting scheme and decision rule.

In our experiments, the parameter β varies in the range between 0.15 and 0.35, α takes values between 0.02 and 0.15, and the parameter k varies between 1 and 100. Then, we chose the parameters with the best performance according to the utility function to represent each algorithm. In this paper, we adopt the traditional vector space model, in which a document d_j is represented as a vector of term weights. The selection of terms includes removing tags and stop words, lemmatization and proper name recognition. Term weigths are computed using the standard *ltc* variant of TF-IDF [15], i.e.:

$$(1 + log\,(TF_i)) * log\left(\frac{N}{DF_i}\right)$$

where TF_i is the frecuency of the term i in the document, DF_i is the number of documents in the profile where the term i occurs and N is the total number of documents in the profile. IDF weights are initialized on the training corpus and are updated as more and more test documents come in. As similarity measure we use the traditional cosine measure.

Table 1 shows the average utility obtained for the algorithms on all topics of the TREC-11 benchmark collection. As we can observe, our approach obtains similar results to those of the first three systems for the assesor topics

Table 1. Comparison of the performance of the algorithms

Algorithm	T11SU	
	R101-R150	R151-R200
ICT	0.475	0.335
KerMIT	0.458	0.285
Reinforcement	0.462	—
k-NN (k =3)	0.115	0.156
Our Approach ($\alpha = 0.14$, $\beta = 0.2$)	0.464	0.490

(R101-R150). In the last 50 topics no results are shown for Reinforcement system, because these are not reported by the authors. Over these topics all the systems presented in TREC-11 competition obtained results under the baseline. However, our approach outperforms both the results obtained by the systems in the TREC-11 competition and the baseline. We consider that these topics are less homogeneus than the assessor topics and represent the situation where the user's information need is satisfied by documents coming from different sources. This situation may be common in real environments, and therefore systems must attempt to correctly handle it.

Regarding k-NN method, we can observe that its results are lower than our approach. This can be explained by the fact that k-NN algorithm delivers to the user many non-relevant documents because the k nearest neighbors may be too far from the document to classify. On the contrary, in the $\alpha\beta$-neighborhood far documents are excluded and do not influence in the classification.

These results show us that, using more than one document for representing the profile, a sufficiently small and homogeneous neighborhood, and a binary classifier for each profile, enables the system to better adapt to different enviroments. However, its time complexity is higher, because in building the neighborhood, it is necessary to compare the new document to all documents that belong to the profile.

5 Conclusions

In this paper, a new approach to adaptive filtering has been proposed. This approach uses a NN classifier based on the $\alpha\beta$-neighborhood on each user profile for determining whether a new document is relevant to the user or not. This approach is based on the idea that the documents belonging to a user profile may form internal subdivisions. In this situation, the homogeneity assumption on the document distribution may hold locally but not globally. For this reason, it is necessary for classification to take into account more documents belonging to the profile and to use only those documents that are close to the incoming document.

The experiments were carried out on TREC-11 benchmark collection. These experiments show that our approach overcomes the best ranked adaptive filtering systems. This leads us to conclude that the proposed method is able to model

the profiles where the documents that satisfy the user's information need show internal subdivisions in a more adequate fashion than the previous methods.

Nevertheless, our method can produce large profiles, and consequently the performance decreases. For that reason, future work include employing some condensing technique to reduce the number of documents in the profile. We also plan to study how sensible is our approach to imbalanced classes and to the variation of α and β thresholds.

References

1. Gil-García, R., Pons-Porrata, A.: A new nearest neighbor rule for text categorization. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 814–823. Springer, Heidelberg (2006)
2. Zhang, Y.: Bayesian Graphical Models for Adaptive Filtering. PhD thesis, Carnegie Mellon University, Pittsburgh, USA (2005)
3. Yang, Y., Kisiel, B.: Margin-based local regression for adaptive filtering. In: CIKM 2003, pp. 191–198. ACM Press, New Orleans (2003)
4. Tebri, H., Boughanem, M., Chrisment, C.: Incremental profile learning based on a reinforcement method. In: Liebrock, L.M. (ed.) 2005 ACM Symposium on Applied Computing, pp. 1096–1101. ACM Press, Santa Fe (2005)
5. Lewis, D.: Applying Support Vector Machines to the TREC-2001 Batch Filtering and Routing Tasks. In: TREC 2001, pp. 286–292. NIST (2001)
6. Cancedda, N., Cesa-Bianchi, N., Conconi, A., Gentile, C., Goutte, C., Graepel, T., Li, Y., Renders, J.M., Taylor, J.S., Vinokourov, A.: Kernel methods for document filtering. In: TREC 2002, NIST (2002)
7. McNamee, P., Piatko, C., Mayfield, J.: JHU/APL at TREC 2002: Experiments in Filtering and Arabic Retrieval. In: TREC 2002, pp. 358–363. NIST (2002)
8. Zhang, Y., Xu, W., Callan, J.: Exploration and exploitation in adaptive filtering based on bayesian active learning. In: Fawcett, T., Mishra, N. (eds.) ICML 2003, Washington, DC, pp. 896–903 (2003)
9. Kassab, R., Lamirel, J.C.: A new approach to intelligent text filtering based on novelty detection. In: 17th Australasian Database Conference, pp. 149–156. Australian Computer Society, Hobart (2006)
10. Ault, T., Yang, Y.: kNN at TREC-9. In: TREC 2000, NIST (2000)
11. Xu, H., Yang, Z., Wang, B., Liu, B., Cheng, J., Liu, Y., Yang, Z., Cheng, X., Bai, S.: TREC 11 Experiments at CAS-ICT: Filtering and Web. In: TREC 2002, NIST (2002)
12. Ault, T., Yang, Y.: kNN, Rocchio and Metrics for Information Filtering at TREC-10. In: TREC 2001, pp. 84–93. NIST (2001)
13. Duda, R., Hart, P., Stark, D.G.: Pattern Classification. Wiley-Interscience, Chichester (2000)
14. Robertson, S., Soboroff, I.: The TREC 2002 Filtering Track Report. In: TREC 2002, NIST (2002)
15. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management 24, 513–523 (1988)