Fourth International Conference on knowledge-Based Intelligent Engineering Systems & Allied Technologies, 30th Aug-1st Sept 2000, Brighton,UK

A content-based collaborative recommender system with detailed use of evaluations

Kaname Funakoshi, Takeshi Ohguro
{kf,ohguro}@cslab.kecl.ntt.co.jp

NTT Communication Science Laboratories
2-4 Hikaridai, Seika, Soraku, Kyoto 619-0237 JAPAN

Abstract.
In this paper, we present a hybrid recommender model that combines the benefits of both content-based filtering and collaborative filtering. In this model, each document profile is represented as a pair of a keyword vector and an evaluation vector. Each user profile, on the other hand, is represented as a matrix of dependency values in relation to other users according to each keyword. This type of recommender system can provide more appropriate documents to suit a user's personal information need. The simulation results showed that our model can provide appropriate documents to users with higher precision than other non-hybrid information filtering models.

1 Introduction

High performance information searches are required to cope with today's growing information overload. Recommender systems, also called information filtering, search appropriate documents or filter out inappropriate documents from several information streams in order to match with a user's general interests [2]. In this paper, we present a recommender model that searches documents with high precision.

Two major types of recommender systems have been proposed: content-based filtering and collaborative filtering. In both types, documents that have high similarities with a user profile are provided to the user. Content-based filtering selects documents based on the contents of documents and each user's preference. In content-based filtering, users can acquire appropriate documents that match with their interests. However, even if the index accurately represents the content of the document, the quality of the document cannot be considered in content-based filtering.

On the other hand, collaborative filtering selects documents based on users' evaluations of the documents. The evaluations possibly indicate the quality of documents so that the quality can be considered in collaborative filtering. However, since collaborative filtering uses the evaluations of other users, it takes some time until the documents can be provided. Moreover, the contents of each document are not referred in the selection.

Both content-based and collaborative filtering have advantages and disadvantages. Thus, they can be complementary. In this paper, we propose a recommendation model "Nakif", which is a hybrid of content-based and collaborative filtering. Nakif uses both the content of each document and a user's evaluation of the document to increase the performance of recommendations.

2 Conventional Approaches

A general conventional model for recommender systems consists of four parts: a set of users P, a set of documents D, a matching (filtering) facility, and a feedback facility. There exist several types of recommender systems with the formations of each part. Here we use the vector space model because it is widely used. In the vector space model, user profiles and document profiles are represented as vectors. The relevance of each document to each user is calculated according to the similarity between the user profile vector and the document profile vector [6].

Let $\boldsymbol{w}_i \in P$ be the profile vector of user i and $\boldsymbol{d}_h \in D$ be the profile vector of document h. Matching is implemented by a matching function $match : P \times D \rightarrow \mathcal{R}$. Documents that have a high-value mark in the matching function are regarded as relevant and are provided to the user. After the user evaluates whether the document provided is appropriate or not, the feedback is used to modify the user profile, the document profile, and, in some cases, the matching function. The modifications aim at increasing the performance of a recommendation in the subsequent filtering.

Content-based filtering

In the content-based filtering model, the user profile and the document profile are represented as weighted vectors of keywords. Let N_{term} be the number of keywords in the system. The pro-

0-7803-6400-7/00/$10.00 ©2000 IEEE

files of user i and document h are written as N_{term} dimension vectors $\boldsymbol{w}_i = (w_{i1}, \ldots, w_{iN_{term}})$ (user's preference for each keyword) and $\boldsymbol{d}_h = (d_{h1}, \ldots, d_{hN_{term}})$ (keywords of the document). The feedback in content-based filtering only refines the user profile to increase the performance in the next selection [4].

Collaborative filtering

In collaborative filtering, a document profile is represented as a vector of user evaluations. A user profile is represented as a vector of dependencies on other users' evaluations. We expand the term "user profile" to represent user features that directly match document profiles. Note that this the term was only used as the user preference in previous studies [7]. Let N_{user} be the number of users in the system. The profiles of user i and document h are written as vectors $\boldsymbol{w}_i = (w_{i1}, \ldots, w_{iN_{user}})$ (dependencies on other users' evaluations) and $\boldsymbol{d}_h = (d_{h1}, \ldots, d_{hN_{user}})$ (evaluations of the document). There are various ways to determine the elements of user profiles; for example, elements may be constructed from the similarity between vectors of users' interests [7] or from the tendency of the user's evaluations [5].

The feedback in collaborative filtering rewrites document profiles as well as user profiles. After each evaluation, the system refines both the document profiles and the user profile.

3 Nakif

Hybrid models

Models for hybrid recommender system are constructed as an extension of the general model discussed previously. The profile of document h is represented as a pair of vectors $\langle \boldsymbol{kw}_h, \boldsymbol{ev}_h \rangle$, where $\boldsymbol{kw}_h = (kw_{h1}, \ldots, kw_{hN_{term}})$ is a keyword vector for the content of the document h, and $\boldsymbol{ev}_h = (ev_{h1}, \ldots, ev_{hN_{user}})$ is an evaluation vector for user-provided evaluation. An element in $\boldsymbol{kw}_h$ and $\boldsymbol{ev}_h$ has a real value.

In a simple hybrid model, a user profile would be represented as a pair of two vectors: a keyword vector and an evaluation vector, i.e., the same form as document profiles. We call this model the "two-vector profile". A keyword vector represents the user's interest and an evaluation vector represents the dependencies on other users' evaluations. In this case, the similarity between keyword vectors of the user profile and document profile is considered as a content-based filtering, while the similarity for evaluation vectors is considered as a collaborative filtering. The system can recommend documents using the sum of these two similarities.

However, since keyword vector and evaluation vector are used independently to select the documents, information such as whether a person is good or weak in some specific subjects cannot be used in the two vector model. Thus, the search is not expected to produce information with high precision.

In order to overcome this problem, we extend the profile of user i to the $N_{user} \times N_{term}$ matrix W^i. This form of a profile makes it possible to treat a person's dependencies on other persons in specific subjects. A very detailed recommendation can be constructed, which cannot be accomplished by simply adding content-based filtering and collaborative filtering.

Nakif in detail

In the proposed hybrid recommendation, the matrix profile of user i has the following form.

$$W^i = \begin{pmatrix} w^i_{11} & \cdots & w^i_{1N_{term}} \\ \vdots & & \vdots \\ w^i_{N_{user}1} & \cdots & w^i_{N_{user}N_{term}} \end{pmatrix}$$

where $w^i_{jk} \in \mathcal{R}$ refers to whether a user i thinks another user j's evaluation of a keyword k is reliable. There are several ways to initialize user profiles. We can initialize W^i with a matrix in which every element has the same values or set at random. To overcome the cold-start problem, we can even start with W^i as $w^i_{jk} = v^i_k$ for all j ($j = 1, \ldots, N_{term}$) by using an initialization vector $\boldsymbol{v}^i \in \mathcal{R}^{N_{term}}$ that represents user i's interests. In this case, in the initial state, the system is regarded as a content-based filtering system.

Matching is implemented by a matching function that should return the similarity between a document profile and user profile. If a document that includes keyword k has been positively evaluated by user j, the document is scored high (or low) in case w^i_{jk} is high (i.e., user i trusts (does not trust) user j's evaluations on keyword k). The matching function is written as follows.

$$\begin{aligned} & match(W^i, \langle \boldsymbol{kw}_h, \boldsymbol{ev}_h \rangle) \\ = & \ \boldsymbol{ev}_h W^i \boldsymbol{kw}^t_h \\ = & \sum_j^{N_{user}} \sum_k^{N_{term}} W^i_{jk} ev_{hj} kw_{hk} \end{aligned} \quad (1)$$

We have two functions for feedback: one modifies $\boldsymbol{ev}_h$ and the other modifies W^i. The feedback on $\boldsymbol{ev}_h$ is the user's evaluation of each document in the set of documents provided. The feedback on

$\boldsymbol{ev}_h$ modifies the document profiles according to the user's evaluation.

User i inputs an evaluation of document h into the system as $e_{hi} \in \{good, bad\}$, where two constants *good* and *bad* are real values such that $good > bad$. Immediately after the evaluation is given, the evaluation part of document profile $\boldsymbol{ev}_h$ is modified as $\boldsymbol{ev}'_h = (ev_{h1}, \ldots, e_{hi}, \ldots, ev_{hN_{user}})$. That is, ev_{hi}'s evaluation of the document is replaced by e_{hi}. This is the result of the first feedback function.

Next, the system modifies the user profile. In general, feedback to the user profiles is described by a feedback function $feedback : P \times D \rightarrow P$. This function rewrites the user profile W^i from the current W^i and the document profile that was just modified by the user evaluation. Although several definitions of *feedback* are possible, we adopted the following definition for its simplicity.

$$W^{i'} = W^i + \delta_{good} \sum_{ev_{hi}=good} \boldsymbol{x}_h^t \boldsymbol{kw}_h - \delta_{bad} \sum_{ev_{hi}=bad} \boldsymbol{x}_h^t \boldsymbol{kw}_h \qquad (2)$$

where δ_{good} and δ_{bad} are constant values, and a vector $\boldsymbol{x}_h$ is as follows.

$$x_{hj} = \begin{cases} 1, & \text{if } ev_{hj} = good \\ 0, & \text{otherwise} \end{cases}$$

When user i evaluates document h that includes keyword k, the feedback process is as follows. If user i positively evaluates the document ($e_{hi} = good$), then for each user j who has evaluated the document positively, w^i_{jk} is increased. Furthermore, if user i negatively evaluates the document ($e_{hi} = bad$), then for each user j who has evaluated the document negatively, w^i_{jk} is decreased.

4 Experiments

We conducted the following simulation to evaluate the proposed model.

Simulation algorithm
0. Set D as an initial set of documents.
1. Add n documents to D.
2. Remove n old documents from D.
3. For each user in P:
  3-1. Calculate the matching function for all documents in D and store the returned values.
  3-2. The documents are sorted and provided according to the stored values. In this experiment, the system provides a limited number of documents within some threshold [1].
  3-3. The user gives either a *good* or *bad* evaluation of the provided documents.
  3-4. Given the evaluation from the user, the system refines the evaluation vector of the document profile and the user profile matrix.
4. Go to step 1. ∎

[1] The threshold is determined to be 20% of documents with a high value of matching function.

In the initial state, all user profile matrices are randomly created. The keyword vector $\boldsymbol{kw}_h$ of each document profile is also randomly created, and the evaluation vector $\boldsymbol{ev}_h$ is prepared so that all of the elements have the value $\epsilon = \frac{good+bad}{2}$, which means "unevaluated". We choose $\delta_{good} = 0.10$ and $\delta_{bad} = 0.01$ because the recommendation performance (i.e., precision) is increased only if $\delta_{good} > \delta_{bad}$.

We introduce *real profile* and *appropriate documents* to simulate user evaluations. Each user has a real profile, which has the same form as the user profile matrix. The real profile indicates if the user trusts another user's evaluation of a keyword. For each round of filtering, a user's real profile is matched with a document profile to construct ranked document list. The document ranked above the threshold, which is called an appropriate document, is evaluated as *good*. We evaluate the system by calculating if the set of provided documents and the set of appropriate documents have the same elements.

Real profiles are randomly created so that some elements are weighted because users' interests are weighted in the real world. We set $N_{user} = 100$ and $N_{term} = 100$. The number of documents in one matching is fixed to 100.

Results

We used (1) content-based filtering, (2) collaborative filtering, (3) the two-vector profile model, and (4) the Nakif (matrix profile) model as mentioned in Section 3. The precision and recall of these recommender systems were measured: precision was the rate of appropriate documents in the set of provided documents, and recall was the rate of appropriate documents provided in all appropriate documents.

Figure 1 shows the feedback-precision graph when recall was set to 80% (200 feedbacks). This figure shows that precision increased earlier and maintained high values in our model. After 120 feedbacks, all four models seemed to perform with high precisions, with our model being the most stable. Figure 2 shows the precision-recall graph after 120 feedbacks; the proposed model maintained high precision even when recall was high.

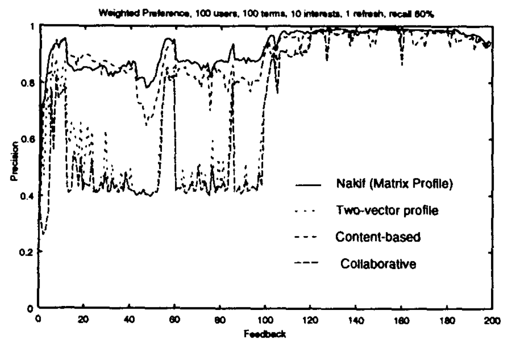

Figure 1: Precision at 80% recall vs. feedback times

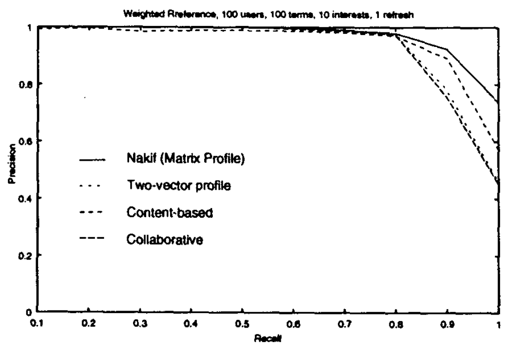

Figure 2: Precision vs. recall after 120 feedbacks

5 Related work

Several recommender systems that use both content-based and collaborative filtering have been proposed. Fab is a content-based collaborative filtering system [1], in which two types of agents (collection agents and selection agents) provide documents to users. Here both types of agents are content-based filtering, but users collaboratively feedback to the collection agents. In contrast to Fab, our model recommends documents in one matching that is both collaborative and content-based.

Delgado et al. have also developed a content-based collaborative recommender system named RAAP [3]. RAAP recommends documents based on two relationships: correlations (symmetric) and confidences (asymmetric) between two users. Correlations between two users are constructed from the "user-category matrix", which has the same form as our user matrix profile and indicates if the user is interested in categories. However, the user-category matrix is used to calculate the correlation between every pair of users. In contrast to RAAP, Nakif's user matrix profile is directly used for recommendations.

6 Conclusion

We have proposed a recommendation model that uses both the content of documents and evaluations of other group members. This model considers the relation between users' evaluations and keywords. Our simulation results have shown that the model provides document recommendations with high precision.

In this work, we assumed that all users have fixed preferences for documents. However, user interest does vary from time to time in the real world. Whether our model can be applied to real world applications is the subject of a future work, and we are planning to experiment with actual documents and actual users.

References

[1] M. Balabanović and Y. Shoham. Content-based, collaborative recommendation. *Communications of the ACM*, vol.40, no.3, pp.66–72 (1997).

[2] N. J. Belkin and W. B. Croft. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, vol. 35, no.12, pp.29–38 (1992).

[3] J. Delgado, N. Ishii and T. Ura. Content-based collaborative information filtering: actively learning to classify and recommend documents. M. Klusch and G. Weiß(eds.) *Cooperative information agents II, 2nd Int. Workshop, CIA '98 Proceedings.* LNAI 1435, Springer, 1999, pp.206–215.

[4] D. Harman. Relevance feedback and other query modification techniques. W. B. Frakes and R. Baeza-Yates (eds.) *Information retrieval: data structures & algorithms.* Prentice Hall, 1992, pp.241–263.

[5] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon and J. Riedl. Applying collaborative filtering to Usenet news. *Communications of the ACM*, vol.40, no.3, pp.77–87 (1997).

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

[7] U. Shardnand and P. Maes. Social information filtering: algorithms for automating "word of mouth". *Proceedings of CHI'95*, pp.210–217.