Content-based Collaborative Filtering for News Topic Recommendation

Zhongqi Lu[†], **Zhicheng Dou**[*], **Jianxun Lian**[‡], **Xing Xie**[‡] and **Qiang Yang**[†]
[†]Hong Kong University of Science and Technology, Hong Kong
[*]Renmin University of China, Beijing, China
[‡]Microsoft Research, Beijing, China
[†]{zluab, qyang}@cse.ust.hk, [*]dou@ruc.edu.cn, [‡]{v-lianji, xing.xie}@microsoft.com

Abstract

News recommendation has become a big attraction with which major Web search portals retain their users. Content-based Filtering and Collaborative Filtering are two effective methods, each serving a specific recommendation scenario. The Content-based Filtering approaches inspect *rich contexts* of the recommended items, while the Collaborative Filtering approaches predict the interests of *long-tail users* by collaboratively learning from interests of related users. We have observed empirically that, for the problem of news topic displaying, both the rich context of news topics and the long-tail users exist. Therefore, in this paper, we propose a Content-based Collaborative Filtering approach (CCF) to bring both Content-based Filtering and Collaborative Filtering approaches together. We found that combining the two is not an easy task, but the benefits of CCF are impressive. On one hand, CCF makes recommendations based on the rich contexts of the news. On the other hand, CCF collaboratively analyzes the scarce feedbacks from the long-tail users. We tailored this CCF approach for the news topic displaying on the Bing front page and demonstrated great gains in attracting users. In the experiments and analyses part of this paper, we discuss the performance gains and insights in news topic recommendation in Bing.

Introduction

News topic recommendation has been an important attraction offered by many major web portals, such as Yahoo! (*www.yahoo.com*), Excite (*www.excite.com*), MSN (*www.msn.com*), Bing (*www.bing.com*) and AOL (*www.aol.com*). News displaying provides an important service that can help keeping the users on the portals. Therefore, it is important for these search portals to understand users' interests and display the "right" news at the right time.

Brute-force recommendation methods such as expert edited rules had been adopted traditionally (Bobadilla et al. 2013), but had shown poor performance due to two major challenges. The first and traditional challenge is that users often prefer to view new items. When a piece of news is just emerging, the recommender system may find it hard to tell whether users are interested in it or not, as the topic may not be in the users' history. A typical solution is Content-based Filtering, where the contents of the news are analyzed before presented to the users. When the users have sufficient news reading records, Content-based Filtering approaches can usually perform well. The second major challenge is due to the scarce clicking feedbacks from users. Because news reading usually may not be the main functions for the web portals, users do not intend to see the news at these portals, even though the news may be of interest to them. A solution to this problem lies in the fact that there are always users who read some news, and these users may serve as a basis to help predict the interests of the long-tail users. This intuition leads to the Collaborative Filtering approach.

Web search portals are starting to recognize the importance of news topic recommendation, where both previously mentioned challenges exist. We propose to bring both Content-based Filtering approach and Collaborative Filtering approach together to make recommendations. Intuitively, the key to both approaches is to find similarities and do clustering implicitly. Content-based Filtering approach relies on the similarity of contexts and clusters the items, while Collaborative Filtering approach finds similarity in user-item links and clusters the links. A difficulty is that these two kinds of similarities are measured under absolutely different scales and cannot be simply summed up.

In this paper, we propose to combine both similarities by a neighborhood model. Specifically, from the contexts of news, we obtain similarity between items. In the optimization phase, we cluster the user-item links by jointly considering the similarity between items. In our proposed model, we combine both advantages of the Content-based Filtering approach and the Collaborative Filtering approach. We present a kernel to capture the contextual information in the news and then integrate the kernel into a Collaborative Filtering framework. On one hand, the proposed model try to understand the emerging news by learning the contextual meanings. On the other hand, by clustering with the similarity of the user-item links, the long-tail users are properly handled.

In the followings, we start by discussing the related works. Then we introduce some preliminary notations and frameworks. We illustrate the motivation by presenting an unique Bing news topic recommendation setting, and then propose our Content-based Collaborative Filtering (CCF) approach, focusing on the two previously mentioned challenges. Finally, we demonstrate the experimental results on

[*]Corresponding author
Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

the Bing portal and discuss our insights from the application.

Related Works

This paper proposed Content-based Collaborative Filtering to recommend news for users. In particular, we would like to use both the contexts of news and the interests of related users to predict a user's news reading interests, so that the proper piece of news could be recommended. The work is mostly related to content based recommendation, collaborative filtering and implicit feedbacks.

Collaborative Filtering is an approach of making automatic prediction (filtering) about the interests of a user by collecting interests from many related users (collaborating). Some of the best results are obtained based on matrix factorization techniques (Marlin 2003; Koren, Bell, and Volinsky 2009; Singh and Gordon 2008). Collaborative Filtering methods are usually adopted when the historical records for training are scarce.

Content-based Filtering Content-based recommender systems try to recommend items similar to those a given user has liked in the past(Lops, De Gemmis, and Semeraro 2011). The common approach is to represented both the users and the items under the same feature space. Then a similarity score could be computed between an user and an item. The recommendation is made based on the similarity scores of a user towards all the items. The Content-based Filtering methods usually perform well when users have plenty of historical records for learning.

Hybrid of CF and Content-based Filtering As a first attempt to unify Collaborative Filtering and Content-based Filtering, (Basilico and Hofmann 2004) proposed to learn a kernel or similarity function between the user-item pairs that allows simultaneous generalization across either user or item dimensions. This approach would do well when the user-item rating matrix is dense (e.g. 6% as reported by (Basilico and Hofmann 2004)). However in most current recommender system settings, the data are rather sparse, which would make this method fail.

The uses of item contents to help the recommendation tasks are widely adopted in various recommendation settings recently. (San Pedro and Karatzoglou 2014) proposed to extend the supervised Latent Dirichlet Allocation model to model expertise in collaborative question answering communities with application to question recommendation. (Liu et al. 2014) proposed to augment item features by a virtual profile based on observed user-item interactions in LinkedIn. As far as we are concerned, our work is the pioneer to retrieve the rich contexts as recommended items to users.

Implicit Feedback is originated from the area of information retrieval and the related techniques have been successfully applied in the domain of recommender systems (Kelly and Teevan 2003; Rendle et al. 2009; Koren 2008; Oard, Kim, and others 1998). The implicit feedbacks are usually inferred from user behaviors, such as browsing items, marking items as favourite, etc. Intuitively, the implicit feedback approach is based on the assumption that the implicit feedbacks could be used to regularize or supplement the explicit training data. From this point of view, our work proposes an effective implicit feedback for the recommender systems during the sale events.

News Recommendation News Recommendation is an application of recommender system techniques and/or Natural Language Processing(NLP) techniques. As a popular service and an important application to retain users, the industry puts much efforts in News Recommendation researches (Das et al. 2007; Li et al. 2010). The initial attempts (Mittermayer and Knolmayer 2006) are NLP oriented. Some works (Kompan and Bieliková 2010) adopt the content-based approach. Some (Das et al. 2007) adopt the Collaborative Filtering approach.

Preliminary

Notations

In this section, we first introduce some general notations in this paper. Other specific notations of the proposed method will be further introduced in the subsequent sections. We adopt special indexing letters for distinguishing users from items: For users, we use u, v; For news items, we use i, j. The set of users is denoted by U, while the set of the news items is denoted by I. A rating r_{ui} indicates the preferences by user u of news item i, where higher values mean stronger preferences. The observed value for the user u over item i, i.e. (u, i), is represented as r_{ui}, and the predicted value is represented as $\hat{r}_{ui}$. We use the superscript $^\top$ to denote the transport of a matrix.

Latent Factor Models

Latent factor models comprise an important approach to Collaborative Filtering. A major advantage of the latent factor models is to tackle the data sparseness issue. We will focus on the models that are induced by the Singular Value Decomposition (SVD) on the user-item preference matrix. A typical model associates each user u with an user-factor vector p_u, and each item i with an item-factor vector q_i. The prediction is given by

$$\hat{r}_{ui} = b_{ui} + p_u^\top q_i$$

where b_{ui} denotes a baseline estimate for an unknown rating r_{ui}:

$$b_{ui} = \mu + b_u + b_i$$

and μ is the overall average rating, b_u and b_i indicate the observed deviations of user u and item i, respectively.

Neighborhood Models

The neighborhood models estimate unknown ratings based on recorded ratings of either like-minded users or similar items. While the neighbor selection could be either item-oriented or user-oriented, in our work, we focus on an item-oriented method. The user-oriented method could be derived in a similar way. As suggested by (Koren 2008), we model the latent factors of an item i by its neighbors $N(\theta, i)$, based on the similarity measure θ. Specifically, $N(\theta, i)$ could be a neighborhood selection function, which returns the neighbors of i when their similarity measured by θ exceeds certain

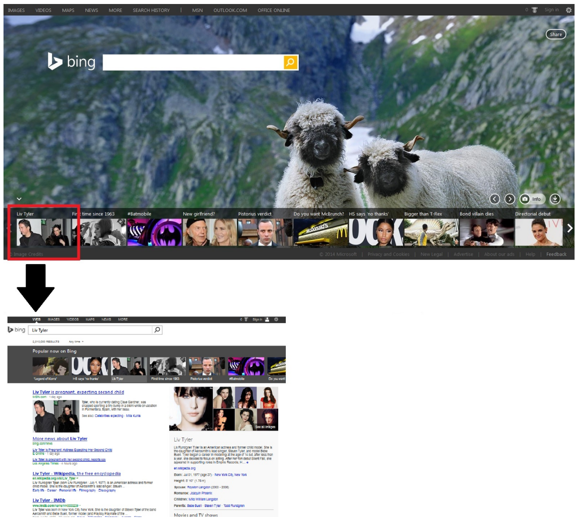

Figure 1: Front page of Bing in the U.S. market, the bottom of which is used for news topic recommendation.

boundary. The prediction is given by

$$\hat{r}_{ui} = b_{ui} + p_u^\top (q_i + \mid N(\theta, i) \mid^{-\frac{1}{2}} \sum_{j \in N(\theta, i)} \theta_{ij} y_j) \qquad (1)$$

where the term $q_i + \mid N(\theta, i) \mid^{-\frac{1}{2}} \sum_{j \in N(\theta,i)} \theta_{ij} y_j$ is defined as the latent factors of an item i, y_j is another set of latent factors to describe the neighbor items of i, and θ_{ij} is the interpolation weights to measure the similarity between the item i and j.

News Topic Recommendation

Bing Search and News Reading

While providing the web search function, the Bing[1] also makes news topic recommendations in the bottom of the webpage. Figure 1 demonstrates a snapshot for the front page of Bing. As we could see in the snapshot, each piece of news is represented by a few keywords decorated by a picture, which are chosen by the human editors as a summary of the news. When clicking, the users will be leaded to a search result page, where the queries are the keywords about the news.

From our statistics, more than half of the recommended news topics will be replaced the next day. Therefore, the recommendation of news is challenged by the cold start issue. However the cold start issues about the news may not affect human readers. Although the news is emerging from time to time, users could have a big picture in mind about the briefing (and/or the backgrounds) of the news by reading the keywords for the news. This big picture is instantiated by the documents in the search result page. A proper recommender system should also take this big picture into consideration when making recommendations.

[1]The news topic recommendation function is currently only available in the U.S. market - http://www.bing.com/?mkt=en-us

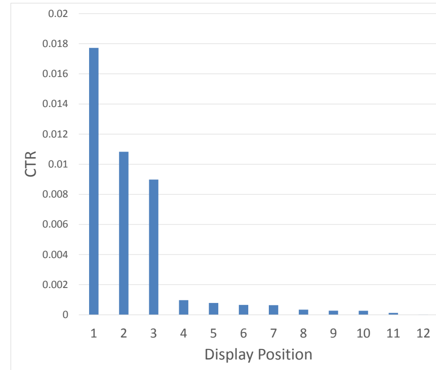

Figure 2: Click Through Rate (CTR) for different positions in news topic recommendation.

Data Analysis of News Reading at Bing

In order to make proper news topic recommendations, we first conduct the data analysis in the news reading settings. We sampled $30,000$ users in Bing and randomly provided news topic recommendations to them in a time span of 5 days, so that the users' feedbacks truly reflect the user's preferences, without the affects of display positions or news rankings etc.

From the statistics, we found the average visit of the front page of Bing is less than 1.5 per day, while the average Click Through Rate (CTR) for any news items is 0.007. Empirically, this indicates that the collected data may not be able to construct traditional recommendation models effectively, due to the sparseness of training data.

Further, we investigate the CTR on different positions. In Figure 1, we have already demonstrated the layout of the news displays. Starting from the left most news item, we mark its position as 1 and the rest of the positions are marked in order. Then we calculate the CTR for each of the positions. The statistics is shown in Figure 2. Because the news items are randomly presented to the users in each position, the content in each position shall not be biased. We found the left most position have the highest CTR. Besides, most of the click behaviours happen in the left three positions.

In order to better serve the users, we would like them to see and click on the news of their concerns. Based on the above observations, it is desirable to place the most interested news on the left most position, while the displays of news in other positions may not be much important. For the ease of experiments and evaluations, we set our objective of recommendation in this work as recommending the most interesting news to users in the left most position.

Content-based Collaborative Filtering Approach

In the following, we describe our proposed method for making news topic recommendations based on both the rich contexts and the collaborative filtering.

Based on the neighborhood models as described in Section *Neighborhood Models*, we derive the collaborative approach. The prediction of user u's interests on an item i is given by

$$\hat{r}_{ui} = b_{ui} + p_u^\top(q_i + \eta) \quad (2)$$

where $b_{ui} = \mu + b_u + b_i, \eta = | N(\theta, i) |^{-\frac{1}{2}} \sum_{j \in N(\theta,i)} \theta_{ji} y_j$. In general, θ is the similarity measure between two items. Under the Bing news topics recommendation setting, θ measures the similarity between two rich bodies of contexts (two sets of documents), which should consider both the semantics and the word frequencies. Fisher kernel would be a desirable choice of such measure. We will discuss the details about this measure of similarity between queries in the next section.

Now we propose to model the latent feature of item i as $q_i + | N(\theta, i) |^{-\frac{1}{2}} \sum_{j \in N(\theta,i)} \theta_{ji} y_j$. We use the item vector q_i to represent the latent feature from the item i itself, and the latent feature vector is complemented by a weighted sum $| N(\theta, i) |^{-\frac{1}{2}} \sum_{j \in N(\theta,i)} \theta_{ji} y_j$, which describes the item i by the latent features of those items from general search logs.

The model parameters associated with the prediction rule in 2 are learnt by solving the regularized least squares problem

$$\begin{aligned} \min_{b_\star, p_\star, q_\star, y_\star} \quad \textstyle\sum_{(u,i)} \quad & \Big(\big(r_{ui} - \mu - b_u - b_i - p_u^\top(q_i + \eta)\big)^2 \\ & + \lambda_1 b_u^2 + \lambda_2 b_i^2 + \lambda_3 \|p_u\|^2 + \lambda_4 \|q_i\|^2 \\ & + \lambda_5 \sum_{j \in N(\theta,i)} \|y_j\|^2 \Big) \end{aligned} \quad (3)$$

where $\lambda_\star$ are regularization constants.

We estimate the model parameters by minimizing the regularized squared error function through stochastic gradient descent. To ease the presentation, we define $e_u i = r_u i - \hat{r}_u i$. We randomly shuffle the user-item pairs in the training set κ and loop over κ to update the parameters. For a particular user-item pair (u, i), we update the parameters by moving in the opposite direction of the gradient, yielding:

$$\begin{aligned} b_u &\leftarrow b_u + \gamma_1(e_{ui} - \lambda_1 b_u) \\ b_i &\leftarrow b_i + \gamma_2(e_{ui} - \lambda_2 b_i) \\ p_u &\leftarrow p_u + \gamma_3(e_{ui}(q_i + \eta) - \lambda_3 p_u) \\ q_i &\leftarrow q_i + \gamma_4(e_{ui} p_u - \lambda_4 q_i) \\ \forall j &\in h(A, i) : \\ y_j &\leftarrow y_j + \gamma_5(e_{ui} p_u \mid N(\theta, i) \mid^{-\frac{1}{2}} \theta_{ji} - \lambda_5 y_j) \end{aligned}$$

where $\gamma_\star$ are constants for the step size.

Measure of similarity between queries θ_{ij}

Since queries are usually very short texts, we adopt a search engine to help understand the similarity between two queries. With the search engine, we first enrich the meaning of the queries with the textual search results, i.e. the documents related to the queries. Then the similarity between two queries is quantified with both the semantic of the documents and the frequencies of their words.

Following the work of (Hofmann 2000), to measure the similarity between two collections of documents, we would like to derive a Fisher kernel function (Jaakkola, Haussler, and others 1999) from the Probabilistic Latent Semantic Analysis (PLSA) model (Hofmann 1999).

Let us consider a collection of documents d_i, and a collection of the words $\{c_n\}$. We define the log-probability of d_i by the probability of all the word occurrences in d_i normalized by length of the document set

$$l(d_i) = \sum_n [\hat{P}(c_n \mid d_i) log \sum_k P(c_n \mid z_k) P(z_k \mid d_i)] \quad (4)$$

where z_k is the latent features, $\hat{P}(c_n \mid d_i) = \frac{COUNT(d_i, c_n)}{\sum_m COUNT(d_i, c_m)}$. Notice that by defining $l(d_i)$ as in Eq.4, $l(d_i)$ is directly correlated to the Kullback-Leibler divergence between the empirical distribution $\hat{P}(c_n \mid d_i)$ and the distribution from PLSA model.

In order to derive the Fisher Kernel, we compute the Fisher information and Fisher scores. By the definition, the Fisher score $u(d_i; \theta)$ is set to be the gradient of $l(d_i)$ with respect to θ. For simplicity, we make the same assumption as in (Hofmann 2000) that the Fisher information matrix approximates to the identity matrix. Above all, the Fisher Kernel of two sets of documents d_i and d_j with respect to the parameter set θ is given by

$$\mathcal{K}(d_i, d_j) = \langle u(d_i; \theta), u(d_j; \theta) \rangle \quad (5)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product operator. Notice that we have two choices for the parameter sets, i.e. $P(z_k)$ and $P(c_n \mid z_k)$. Due to the limits of spaces, we omit the detailed derivation of the gradients and present the results. The similarity measure due to the parameters $P(z_k)$ is given by

$$\mathcal{K}_1(d_i, d_j) = \sum_k \frac{P(z_k \mid d_i) P(z_k \mid d_j)}{P(z_k)} \quad (6)$$

And the similarity measure due to the parameters $P(c_n \mid z_k)$ is given by

$$\mathcal{K}_2(d_i, d_j) = \sum_n [\hat{P}(c_n \mid d_i) \hat{P}(c_n \mid d_j) \sum_k \frac{P(z_k \mid d_i, c_n) P(z_k \mid d_j, c_n)}{P(c_n \mid z_k)}] \quad (7)$$

where $P(z_k \mid d_i)$, $P(z_k)$, $P(z_k \mid d_i, c_n)$ and $P(c_n \mid z_k)$ are obtained from the estimation of the PLSA model in (Hofmann 1999).

The $\mathcal{K}_1$ kernel computes a "semantic" overlap between the two queries via the analysis of two sets of documents, while the $\mathcal{K}_2$ kernel handles the empirical word distributions. We sum the outputs of both measures to produce the similarity θ_{ij}:

$$\theta_{ij} = \rho \mathcal{K}_1(d_i, d_j) + (1 - \rho) \mathcal{K}_2(d_i, d_j) \quad (8)$$

where ρ is the weighting constant to balance the "semantic" overlap and the empirical word distributions when calculating the similarity. And the subscripts i, j is used to index the news items, i.e. $i, j \in I$ and I denotes the index set of the news topics.

External Search Queries as Auxiliary Data

As discussed in a previous section , because of the limited number of view and click behaviours, it would be helpful to use some auxiliary data to help building the links between different pieces of news.

From Bing search, we have plenty of external queries and their search results. In the settings of Bing news topic recommendation as we have discussed, we are presenting news topic recommendations in a format of search result, and a recommendation algorithm is proposed based on the analyzing of search results. Therefore, it is natural to integrate the external search queries and their search results into our proposed model.

As a preliminary attempt to extend our proposed model, we augment the learning of θ. In the previous section, θ is learned to measure the similarity between two queries in the news set I. We introduce the external search query set I', so that θ is learned based on both I and I'. Formally from Equation 8, we obtain θ_{ij}, where $i, j \in \{I, I'\}$. Then this larger θ is used in our previous model.

Experiments

Experimental Settings

In the experiments, we are applying our CCF approach on the Bing news reading dataset. To collect for the dataset, we sampled $30,000$ users. In order to eliminate the bias of position as mentioned in Section *Data Analysis in News Reading at Bing*, we randomly displayed news to these users in the period of data collection. The data collections lasted for 5 days. During the data collection, there are 183 pieces of news displayed on the front page of Bing. We retrieved at least 8 documents for each of the news from Bing search engine and in total there are 1793 documents. To train the model, we adopt the data in the first 3 days. And to test, we use the data in the rest 2 days.

As we have discussed in Section *Data Analysis in News Reading at Bing*, improving the overall users' experiences is equivalent to refining users' satisfaction on the left most position for news display, which is also adopted in the previous news topic recommendation researches (Li et al. 2010). Therefore, in the experiments, we focus on the feedbacks from the left most position.

Performance Comparisons

As discussed previously, to measure the performance, we use the "root-mean-square error" (RMSE) on the left most position of the news displays as the evaluation matric throughout the experiments:

$$RMSE = \sqrt{\sum_{u=1}^{m}\sum_{i=1}^{n} I_{ui}(\mathbf{R}_{ui} - \hat{\mathbf{R}}_{ui})^2 / \mid I \mid}$$

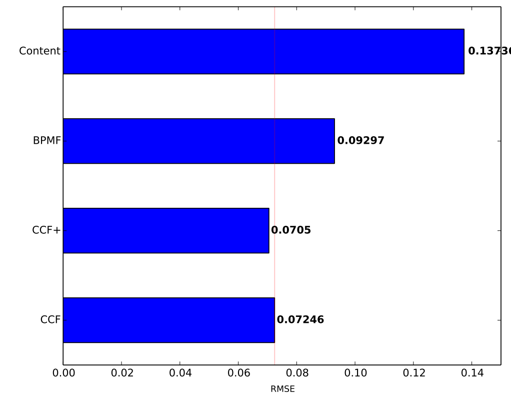

Figure 3: Performance comparison for different recommendation methods.

where $\mathbf{R}_{ui}$ and $\hat{\mathbf{R}}_{ui}$ represent the true and the predicted values for the user-item scores respectively. I_{ui} is the indicator function which is equal to 1 if user u browsed item i, and 0 otherwise. The RMSE is widely adopted as the evaluation matric in the evaluation of recommender systems, such as the Netflix Prize (Bennett and Lanning 2007). In our settings, the RMSE indicates the difference between users' true interests towards the news in the left most position and the predicted ones. The smaller RMSE means better performance of the method.

Other than *CCF*, we compare with the following methods:

- *BPMF* (Salakhutdinov and Mnih 2008). BPMF is a classical Collaborative Filtering approach, using fully Baysian treatment of the Probabilistic Matrix Factorization. It is generally considered to be one of the most effective method when the training data is sparse.
- *Content* As surveyed by (Lops, De Gemmis, and Semeraro 2011), the most effective content-based document recommendation method calculate the document similarity based on both the keyword-based vector space model and the semantic analysis. In our experiments, we use the similarity measure θ, as described in a previous section. This could be considered as an effective Content-based Filtering method.
- *CCF+* We use *CCF+* to denote the extension of the CCF approach, which is discussed in Section *External Search Queries as Auxiliary Data*. These external contents come from the Bing general search. They includes about 1000 general search queries and the resulting documents.

The comparison is shown in Figure 3. Because there are much more negative training samples (exposed but not clicked) than the positive samples in the dataset, the absolute value of RMSE is not very meaningful. As a dummy baseline under the measure of RMSE, we report the RMSE for random recommendation to be 0.20736. We are expecting the relative decreasing of RMSE for a more effective method.

As a combination of both the Content-based Filtering approach and the Collaborative Filtering approach, the *CCF/CCF+* approaches outperform. This empirically demonstrates that our proposed CCF approach is effective in the Bing news topic recommendation settings. Besides, the extended CCF approach, i.e. *CCF+*, leads to more improvements over *BPMF* and *Content* than the *CCF* does. We will further discuss this observation together with another finding in the following sections.

Effectiveness of Contents

We inspect the effectiveness of the contents in helping the Content-based Collaborative Filtering. The results are shown in Figure 4. In figure 4, the x-axis represents number of average resulting documents returned from search. The y-axis represents the RMSE value. For BPMF, it is a pure CF method and does not use search results as inputs. For Content, we use all available search results as inputs. Therefore, for both BPMF and Content methods, which are discussed in Section *Performance Comparisons*, the values in y-axis are constant. However, as a variation of Content method, we use different numbers of documents for each piece of news when learning the similarity between the news. With the increasing number of documents for each piece of news, better results are achieved, as expected. In the extreme case when the number of documents equals zero, the recommendation becomes random.

For the CCF methods, we use different numbers of documents for each piece of news when learning the θ. When the number of documents equals zero, the CCF method reduces to a Collaborative Filtering one, which has similar performance with *BPMF*. Besides, with the increasing number of documents for each piece of news, there is a trend of constant decreasing of RMSE for *CCF*. Due to the system limitations, we kept a limited number of logged documents for each piece of news. But this trend might be served as a good hint for future refinements in the real world applications.

Notice that for the *CCF+*, which uses external contents, we do not report the results of increasing external contents, because the performance gain is not significant in our experiments. There might be some good results when the amount of external contents is increased to the real world scale, i.e. thousands of Terabytes of records, because the general search results might contain the news related documents. However, because this is not the key point of this research and due to lack of the computational power, we would like to save this issue to the future works.

Discussions about Contents in CCF

From Figure 3, we notice that the extended CCF approach, i.e. *CCF+*, leads to more improvements over *BPMF* and *Content* than the *CCF* does. This indicates that the general search queries and resulting documents could be used into the CCF framework, which is an interesting finding. However, from Figure 4 in the experiments of increasing the number of documents for each piece of news, we notice that the trend of performance gain for *CCF* shows when larger number of documents for each piece of news is provided, the performance might be further increased. The two

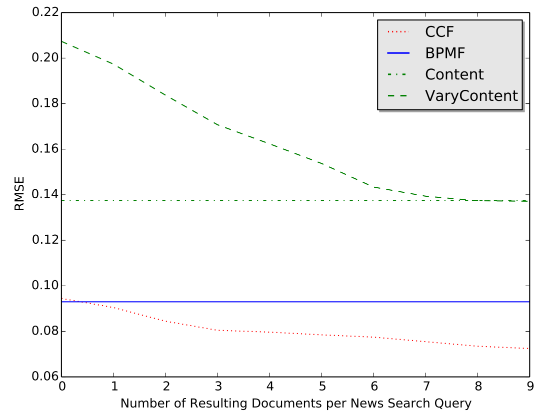

Figure 4: Change the number of news related documents for each piece of news.

observations leads to two possible directions to the future works: either to find good resources to obtain news related documents, or to develop better methods to mine the general search results, which obviously contains much news related documents. Under the framework of CCF, we admit that the performance gain of *CCF+* over *CCF* might be larger, if some methods could be proposed to more effectively transfer the knowledge from the general search queries to the news topic recommendation settings. We hope this preliminary attempts in the CCF framework could inspire the future works.

Another concern is that will some noisy contents lead to bad performance in the CCF approaches? For the CCF approach in this work, we use θ to measure the similarity between two pieces of news, and only the similar piece of news shall be used in the CCF methods. As described in Section *Measure of similarity between queries* θ_{ij}, we define kernels to consider both the semantics and the words distributions in measuring the similarity. This similarity checking strategy would ensure no noisy contents would be used in the learning of the CCF models, although the computational complexity might be an issue.

Conclusions

We proposed the Content-based Collaborative Filtering (CCF) approach for the news topic recommendation in Bing. By utilizing the rich contexts and focusing on the long-tail users, the proposed CCF combines both the advantages of Content-based Filtering approach and the features of Collaborative Filtering approach. This CCF is designed for the settings like Bing news topic recommendation, where a piece of news could be interpreted by rich contexts, such as the querying results. We have demonstrated the performance gains of the CCF approaches over the others in our Bing news displaying dataset. Under this CCF framework, we discussed our insights towards the news topic recommendation for today's web portals.

Acknowledgments

The research has been supported in part by China National 973 project 2014CB340304 and Hong Kong RGC Projects 621013, 620812, and 621211.

References

Basilico, J., and Hofmann, T. 2004. Unifying collaborative and content-based filtering. In *Proceedings of the twenty-first international conference on Machine learning*, 9. ACM.

Bennett, J., and Lanning, S. 2007. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, 35.

Bobadilla, J.; Ortega, F.; Hernando, A.; and Gutiérrez, A. 2013. Recommender systems survey. *Knowledge-Based Systems* 46:109–132.

Das, A. S.; Datar, M.; Garg, A.; and Rajaram, S. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, 271–280. ACM.

Hofmann, T. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 50–57. ACM.

Hofmann, T. 2000. Learning the similarity of documents: An information-geometric approach to document retrieval and categorization.

Jaakkola, T.; Haussler, D.; et al. 1999. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems* 487–493.

Kelly, D., and Teevan, J. 2003. Implicit feedback for inferring user preference: a bibliography. In *ACM SIGIR Forum*, volume 37, 18–28. ACM.

Kompan, M., and Bieliková, M. 2010. Content-based news recommendation. In *E-commerce and web technologies*. Springer. 61–72.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 426–434. ACM.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670. ACM.

Liu, H.; Goyal, A.; Walker, T.; and Bhasin, A. 2014. Improving the discriminative power of inferred content information using segmented virtual profile. In *Proceedings of the 8th ACM Conference on Recommender systems*, 97–104. ACM.

Lops, P.; De Gemmis, M.; and Semeraro, G. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*. Springer. 73–105.

Marlin, B. M. 2003. Modeling user rating profiles for collaborative filtering. In *NIPS*.

Mittermayer, M.-A., and Knolmayer, G. 2006. *Text mining systems for market response to news: A survey*. Institut für Wirtschaftsinformatik der Universität Bern.

Oard, D. W.; Kim, J.; et al. 1998. Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*, 81–83. Wollongong.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 452–461. AUAI Press.

Salakhutdinov, R., and Mnih, A. 2008. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, 880–887. ACM.

San Pedro, J., and Karatzoglou, A. 2014. Question recommendation for collaborative question answering systems with rankslda. In *Proceedings of the 8th ACM Conference on Recommender systems*, 193–200. ACM.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 650–658. ACM.