Personalized News Recommendation Based on Collaborative Filtering

Florent Garcin, Kai Zhou, Boi Faltings

Artificial Intelligence Lab
Ecole Polytechnique Fédérale de Lausanne
Switzerland
firstname.lastname@epfl.ch

Vincent Schickel

Prediggo SA
Switzerland
schickel@prediggo.com

***Abstract*—Because of the abundance of news on the web, news recommendation is an important problem. We compare three approaches for personalized news recommendation: collaborative filtering at the level of news items, content-based system recommending items with similar topics, and a hybrid technique.**

We observe that recommending items according to the topic profile of the current browsing session seems to give poor results. Although news articles change frequently and thus data about their popularity is sparse, collaborative filtering applied to individual articles provides the best results.

I. Introduction

The web provides instant access to a huge variety of online news. It is desirable to have recommender systems that can point a user to the most relevant items and thus avoid extensive search. Recommender systems have been used with good success for products such as books and movies, but have found surprisingly little application in recommending news articles.

News articles must be recommended soon after they are written, hence leaving little time to collect popularity data. Thus, techniques for recommendations based on user behaviour have been considered difficult to apply. Topic models may overcome this by modelling behaviour at the level of topics rather than individual articles [1].

There are often little data available about a user's past behaviour. Since news sites are reached through search engines, we examine a scenario where users are anonymous and only current visit data can be used to make recommendations.

We obtained access logs of two daily newspaper websites: *24 Heures* (24H, www.24heures.ch) and *Tribune de Genève* (TDG, www.tdg.ch), the most-popular newspapers in Canton of Vaud and Geneva, respectively. We evaluated recommendations by comparing them to the news stories that a user read based on earlier news items read during the same visit, and counted every prediction of a future news item that the user actually read.

Contrary to common wisdom [2], [1], we find that content-based and hybrid techniques have surprisingly poor performance. We explain this by the fact that users do not like to read multiple stories about the same topic. We show that collaborative filtering techniques can provide recommendations that are significantly better than such techniques, even with the limited amount of data about the user.

II. Related work

There are two classes of recommender systems [3]. Collaborative filtering [4] recommends items to a user based on users with similar tastes, while content-based techniques [5] create recommendations by analysing the content of the items.

For news recommendation, collaborative filtering is not very common. The Grouplens project was the first to apply collaborative filtering to newsgroups [6], [7], but since then it was not often used except in news aggregation systems such as Google News [2]. In this work, the authors mix collaborative filtering and content-based techniques.

Content-based recommendation is more common for news personalisation. NewsWeeder [8] is probably the first content-based approach for recommendations but in newsgroups. NewsDude [9] or YourNews [10] are two implementations of a content-based system.

It is also possible to combine the two types together in a hybrid system [11]. For example, Liu et al [1] extend the study on Google News [2] by looking at the user click behaviour in order to create accurate user profiles. Li et al [12] introduce a contextual bandit algorithm which recommends news stories based on contextual information about the users and stories.

Most of these works rely on the history of logged-in users. This causes potential privacy issues to the users. Our work departs from this restriction and considers only a one-time session for recommendation, where users do not log in.

III. Recommendation Techniques

We consider three different kinds of recommender systems: collaborative filtering at the level of news items [4], content-based recommendation [5] where we recommend items with similar topics to what was read, and a hybrid [11] where collaborative filtering is applied at the level of topics.

We build a candidate set $\mathcal{C}$ by selecting the most "fresh" news articles. Potential candidates for recommendation are the last $|\mathcal{C}|$ news stories that were accessed at least once. Although this reduces drastically the set of news for recommendation, the main advantage is that it also decreases the complexity of recommender algorithms.

As a baseline, we use the most-popular recommender system which recommends a set of news stories with the highest number of clicks within the candidate set. Although naïve, this approach is actually used very often on newspaper websites.

A. Collaborative Filtering on News Items

Collaborative recommendation compares reading histories in order to extract reading behaviour patterns. It recommends news items that other readers with similar reading histories have read. Because of the sequential nature of news reading, it is natural to model news browsing as a Markov process [13]. Readers are in different states at a given time, and recommendations are generated by looking at the transition probability from one state to another.

Intuitively, a state corresponds to one news item. However, this representation is limited to the previous news story and does not take into account all the news items the reader has seen since the beginning of her visit. Therefore, we adopt a p-gram model of news stories.

We introduce here two parameters, namely the *past* p and *future* f. The *past* indicates how much historical information we consider for generating recommendations. The *future* is how far we look into the future, corresponding to the number of steps we need to reach a given state. For instance, when $p = 3$ and $f = 2$ we have a trigram model and look at 1- or 2-steps transition probabilities.

More formally, let $\mathcal{N}$ be the set of news stories and $n_i \in \mathcal{N}$ a news item. Let S be the set of states, and $s_i \in S$ a state. We define a state s as a sequence of p news items and write $s = \langle n_1, n_2, ..., n_p \rangle$ for a given sequence $n_1 \rightarrow n_2 \rightarrow ... \rightarrow n_p$. We represent a user's visit as a sequence of states. For instance in the case of $p = 3$, an anonymous user who reads news stories in the sequence $n_1 \rightarrow n_2 \rightarrow n_3 \rightarrow n_4$ is represented as a path $\langle n_1, n_2, n_3 \rangle \rightarrow \langle n_2, n_3, n_4 \rangle$.

We implement a item-based collaborative filtering as follows. We denote $TP(s_i, s_j)$ the probability of transition from state s_i to state s_j. It is estimated by counting how many times the transition occurs in the dataset.

$$TP(s_i, s_j) = \frac{count(s_i, s_j)}{\sum_{s \in S} count(s_i, s)} \tag{1}$$

Given the past p and state $s = \langle n_1, n_2, ..., n_p \rangle$, the *reading probability* from the state s to a news story n in exactly one step is defined as

$$RP_1(s, n) = TP(s, \langle n_2, ..., n_p, n \rangle) \tag{2}$$

The reading probability of a news story n given the state s in exactly f steps is computed as

$$RP_f(s, n) = \sum_{s' \in S} TP(s, s') RP_{f-1}(s', n) \tag{3}$$

Hence, the *reading probability* of a news item n in at most f clicks given the sequence s, or the *score* of n given s, is

$$score(n|s) = \sum_{i=1}^{f} \left[RP_i(s, n) \prod_{j=1}^{i-1} (1 - RP_j(s, n)) \right] \tag{4}$$

The system evaluates each candidate news story and recommends the news items with the highest scores.

So far, we model the visits as an ordered sequence of clicks on some news items, and we call it *sequence-of-news* recommender system. Alternatively, we could also consider a "bag" of news instead of an ordered sequence. For instance with a 2-gram model, two readers with the following histories $n_1 \rightarrow n_2$ and $n_2 \rightarrow n_1$ would correspond to the same state $\{n_1, n_2\}$, and no longer to $\langle n_1, n_2 \rangle$ and $\langle n_2, n_1 \rangle$. By doing so, we make the model simpler and more flexible because we will have more data to generate the recommendations for a given state. In other words, the history of a reader does not need to match exactly the history of someone else, but it has to contain the same news items. The definition of reading probability is slightly changed, and is now the transition probability from one bag to another times the popularity of the candidate. We name this method the *bag-of-news* recommender system.

Another possible model considers *co-occurrence* as transition probability. In the training phase, given the past p, future f and $p = f$, all the p-nearest news items are considered as neighbours of the current item. Thus, the neighbour set contains news items after and before the current news but no further than p. We recommend news items with the highest co-occurrence score.

Finally, two issues remain to be addressed. First, we have a cold-start problem when we need to generate recommendations for the first news story. Since we do not have any specific information, we rely on the most-popular recommender system in this case. Second, when the past p becomes large, the sequences become too specific and it is more difficult to generate recommendations because the dataset does not contain enough samples. In this case, we reduce p until we find a valid value.

B. Content-based Recommender System

To avoid the cold-start problem, another approach is to recommend news articles that have similar content as the ones that a user previously read. In general, a recommender system has the following steps: first, it represents the news stories as a vector of features; second, the similarity between vectors are evaluated and finally, it recommends the set with the most similar news items (vectors of features).

We use a probabilistic topic model technique to learn the content. In particular, we choose the Latent Dirichlet Allocation (LDA) over other methods such as Probabilistic Latent Semantic Indexing [14] because the latter suffers from overfitting in practice [15].

The key idea behind LDA is that a journalist writes an article with particular topics in mind, and she draws words with a certain probability from a bag of words of each topic. A news story is then represented as a mixture of various topics.

TABLE I
DATASETS AFTER FILTERING.

	News stories	Visits	Clicks
TDG	10'400	600'256	1'069'131
24H	8'613	249'099	509'978

To apply LDA, we concatenate the title, summary and content of the news item together, then we tokenize the words and remove stopwords. After that, we apply LDA to all the news stories in the dataset, and obtain a topic distribution vector for each news item. Note that the topics might have no meaning because they are neither classified nor named.

Given the topic distribution and the history of a user, it is possible to build a reader's profile based on previously clicked news. The *reader-profiling* recommender system aggregates the topic vectors into one profile vector by averaging them. After that, it computes the similarity between the reader's profile vector and the topic vector of each news candidate. Finally, it recommends the most similar news story to the reader's profile. Again, we use the *past* p which indicates how far in the history we look back.

We investigated 4 different similarity measures: Kullback-Leibler (KL), symmetric KL, Jensen-Shannon (JS) and cosine similarity. The cosine similarity outperformed the other similarities.

C. Hybrid System

The cold-start problem in collaborative filtering can be addressed by applying it at the level of a topic model [1]. We implemented such a system where we cluster news stories based on topic distributions, then build a transition matrix between clusters. Recommendations are made with the transition probabilities and the conditional probability of a specific news story given the current cluster.

We apply a k-means clustering algorithm on the topic vectors to get the clusters. Another solution, named *most probable topic clustering*, is to label each news story with the most probable topic.

Once the clusters are determined, we build a probabilistic model similar to Section III-A. All probabilities (Eq. 1, 3 and 4) are defined with respect to clusters $C \in \mathbb{C}$.

As a news story belongs to only one cluster, the *reading probability* of a news candidate $n' \in C'$ given that the user has previously read news item $n \in C$ in exactly one step is

$$RP_1(C, n') = TP(C, C')Pr(n'|C') \tag{5}$$

where $Pr(n'|C') = Clicks(n')/\sum_{n \in C'} Clicks(n)$ is the conditional probability of reading a news candidate $n' \in C'$ given its cluster is C'. We estimate this probability by using the probability of the news story being clicked in this cluster.

IV. Evaluation

We evaluate prediction quality for the future news a user is going to read. Specifically, we consider sequences $s_i \in S$ of news items $s_i = \langle n_1, n_2, n_3, ..., n_l \rangle$, $n_i \in \mathcal{N}$ read by anonymous users. The sequences and the news items in each sequence are sorted by increasing order of visit time. When an anonymous user starts to read news item n_1, the system generates recommendations. As soon as the user reads another news item n_2, the system updates its model with the past observations n_1 and n_2, and generates a new set of recommendations. Hence the training set and the testing set are split based on the current time: at time t, the training set contains all news items accessed before t, and the testing set has items accessed after t.

For a given sequence $s = \langle n_1, n_2, ..., n_t, ..., n_l \rangle$ and the current news item n_t in this sequence, we define $\mathcal{S}$ as the set of successor news items such that $\mathcal{S} = \{n_i : i > t\}$, and $\mathcal{R}$ as the set of recommended news items. We say that a recommended news item is relevant if it is in the successor set. We always recommend 5 news stories, and we use two metrics to evaluate how good the recommendations are: $Success@5$ ($S@5$) and *Mean Average Precision (MAP)*.

$S@5$ is equal to 1 if the immediate successor of the current items is recommended among the first 5 recommended news stories, 0 otherwise.

We compute the precision at every position k in the ranked sequence of news stories, and take the average at the rank of each relevant news stories. The *Mean Average Precision* is the mean of the average precision for each recommendation set the system generates.

$S@5$ captures how good recommended items are against the immediate successor. However, *MAP* looks at all future news stories and how the recommended news are ordered.

We collected data from the websites of two daily Swiss-French newspapers called *Tribune de Genève* (TDG) and *24 Heures* (24H)[1]. Their websites contain news stories ranging from local news, national and international events, sports to culture and entertainment.

The datasets span from Nov. 2008 until May 2009. They contain all the news stories displayed, and all the visits by anonymous users within the time period. Note that a new visit is created every time a user browses the website, even if she browsed the website before. Table I shows the dataset statistics after filtering out noise.

V. Results

The number of topics does not influence the performance a lot. We decided to fix this to 50 after evaluating experimentally the optimal value in the range from 10 to 200. For the k-means recommender, we choose 30 clusters. We report averages with confidence intervals at 95%. For both datasets, we observed qualitatively very similar behaviours, even though they come from newspapers in different cities and with different readers.

Fig. 1 compares the different approaches against the parameter past p for the 24H dataset. Surprisingly, content-based and hybrid approaches are not as good as pure collaborative filtering for both evaluation metrics. One reason may be that

[1] In 2011, TDG and 24H had a readership of 138'000 and 223'000, and a circulation of 51'487 and 75'796, respectively.

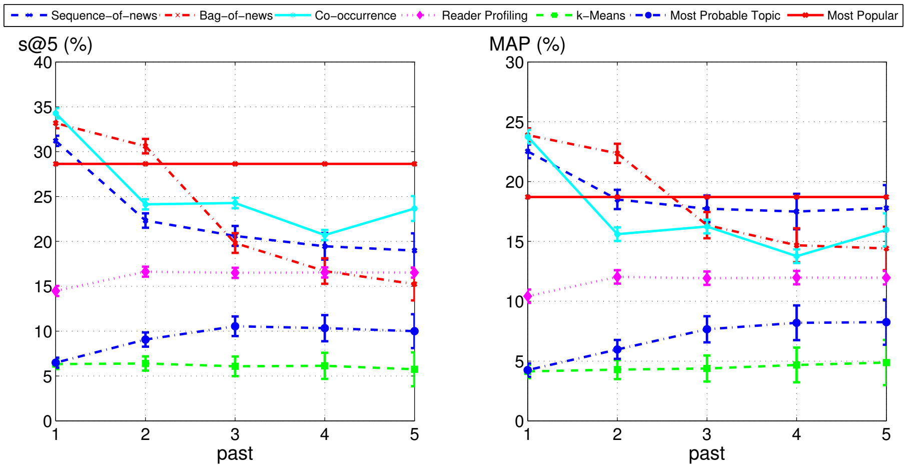

Fig. 1. Evaluation of different recommendation strategies for 24H dataset ($f = 1$, $|\mathcal{C}| = 60$).

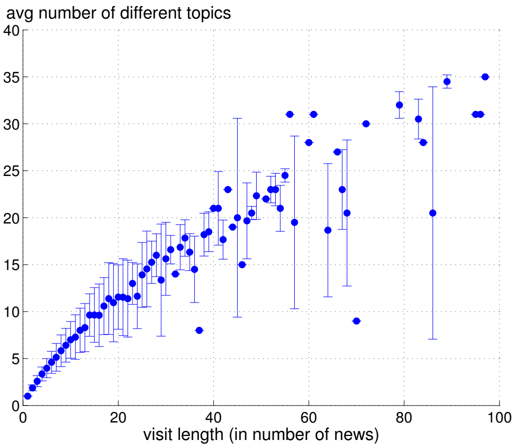

Fig. 2. Average number of different topics per visit, with standard deviation (24H dataset).

the topic alone does not identify which articles an individual reads because it does not capture the differences in quality and orientation of individual news stories. Users do not want to read always about the same topic (Fig. 2). Users who want to learn in-depth about a particular topic tend to stay longer.

Fig. 3 shows how accurate the systems are when we consider only personalized items. To do this, we removed the popular items from $\mathcal{R}$. Content-based and hybrid systems lead again to very poor performance although the items are diverse. There is no tradeoff between overall accuracy and diversity. On the other hand, collaborative filtering systems achieve a relatively good performance with enough diversity compared to the most-popular approach.

The performance of collaborative filtering methods degrades with increasing complexity, which is most probably due to overfitting. However, even the simple collaborative filtering method is much better than the currently common method of recommending the most popular articles, showing that even with little data personalization can bring big benefits. If more data are available, the performance could only improve.

On the other hand, the performance of content-based and hybrid methods improves with increasing complexity, suggesting that no overfitting is present. Thus their performance is not likely to improve when more data is available. We are thus confident in the conclusion that content-based recommendation is not a good idea.

As $|\mathcal{C}|$ becomes bigger (Fig. 4), the set of news candidates contains more ”stale” news. In the case of collaborative filtering systems, the choice of $|\mathcal{C}|$ can significantly increase the performance since stale items are filtered by the recommender itself. However after a particular threshold, a bigger $|\mathcal{C}|$ will neither improve nor harm the performance of the systems. The ”stale” news do not have covisitations with ”fresher” news. In other words, users are more interested in news that are ”fresh”.

The future f does not influence a lot the performance of the collaborative filtering, content-based or hybrid methods. In some cases such as the bag-of-news recommender, the performance slightly decreases as the future increases.

VI. Conclusion

News recommendation is challenging because of the intrinsic property of a news item: when a news story is very recent, there is little data available to generate recommendations.

In this paper, we considered 3 kinds of recommender systems: collaborative filtering at the level of news items, content-based recommendation where we recommend items with similar topics to what was read, and a hybrid where collaborative filtering is applied at the level of topics.

Contrary to common wisdom [2], [1], the content-based and hybrid recommendations have surprisingly poor performance. We explain this by the fact that users do not always want to read about the same topic: a user reads on average 7 different topics for a 10-news visit. The topics do not capture the differences in quality and orientation of individual items, and how this affects the choice of individual articles.

Collaborative filtering on individual news items within a small candidate set works surprisingly well, and better than

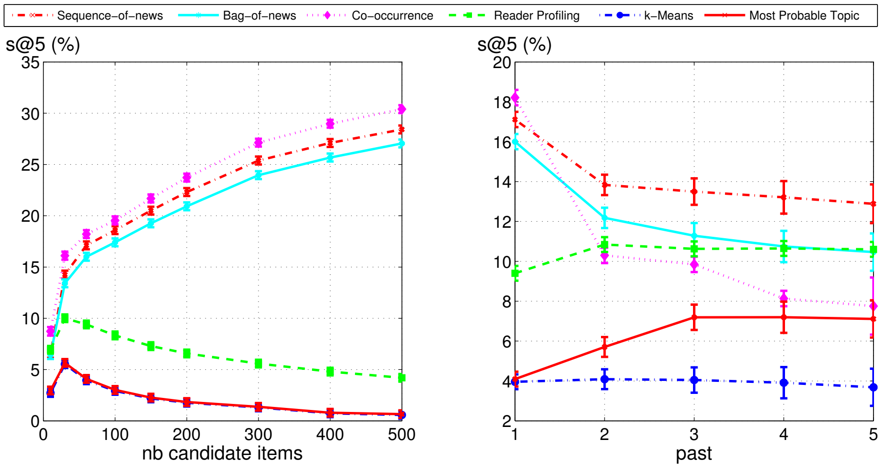

Fig. 3. Success@5 for personalized news items (24H dataset; $p = f = 1$ for left graph; $f = 1$, $|\mathcal{C}| = 60$ for right graph).

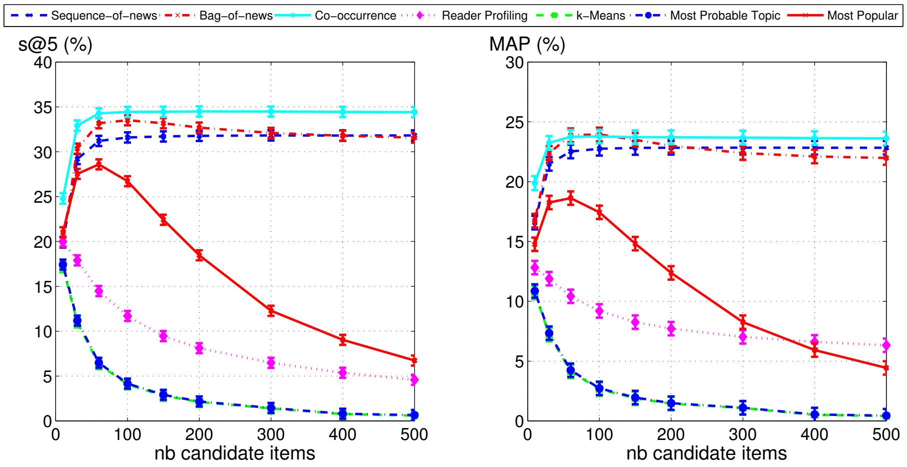

Fig. 4. Effect of number of candidates $|\mathcal{C}|$ for 24H dataset ($p = 1$, $f = 1$).

with topic models despite the data is insufficient. The hybrid approach does not seem to solve the issue that users do not always want to read the same topic.

In conclusion, we demonstrated that personalized recommendations using collaborative filtering can be useful even for individual newspaper sites with limited amounts of data about their users. We believe that news sites should consider these techniques for keeping readers interested in their sites.

References

[1] J. Liu, P. Dolan, and E. Pedersen, “Personalized news recommendation based on click behavior,” in *Proc. of the 15th Int. Conf. on IUI*, 2010, pp. 31–40.

[2] A. Das, M. Datar, A. Garg, and S. Rajaram, “Google news personalization: scalable online collaborative filtering,” in *Proc. of the 16th Int. Conf. on World Wide Web*, 2007, pp. 271–280.

[3] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *Trans. on Know. and Data Eng.*, vol. 17, pp. 734–749, 2005.

[4] X. Su and T. Khoshgoftaar, “A survey of collaborative filtering techniques,” *Adv. in Artif. Intell.*, pp. 2–2, January 2009.

[5] P. Lops, M. Gemmis, and G. Semeraro, “Content-based recommender systems: State of the art and trends,” in *Recommender Systems Handbook*, 2011, pp. 73–105.

[6] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “Grouplens: an open architecture for collaborative filtering of netnews,” in *Proc. of the Conf. on CSCW*, 1994, pp. 175–186.

[7] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, “Grouplens: applying collaborative filtering to usenet news,” *Commun. ACM*, vol. 40, pp. 77–87, March 1997.

[8] K. Lang, “Newsweeder: Learning to filter netnews,” in *Proc. of the 12th Int. Conf. on Machine Learning*, 1995, pp. 331–339.

[9] D. Billsus and M. Pazzani, “A hybrid user model for news story classification,” in *Proc. of the 7th Int. Conf. on User Modeling*, 1999.

[10] J. Ahn, P. Brusilovsky, J. Grady, and D. He, “Open user profiles for adaptive news systems: help or harm?” in *Proc. of the 16th Int. Conf. on WWW*, 2007, pp. 11–20.

[11] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User Modeling and User-Adapted Interaction*, vol. 12, pp. 331–370, November 2002.

[12] L. Li, W. Chu, J. Langford, and R. Schapire, “A contextual-bandit approach to personalized news article recommendation,” in *Proc. of the 19th Int. Conf. on World Wide Web*, 2010, pp. 661–670.

[13] G. Shani, D. Heckerman, and R. Brafman, “An mdp-based recommender system,” *J. Mach. Learn. Res.*, vol. 6, pp. 1265–1295, 2005.

[14] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proc. of the 22nd Int. Conf. on Research and Development in Info. Retrieval*, 1999, pp. 50–57.

[15] D. Blei, A. Ng, and M. Jordan, “Latent dirichlet allocation,” *J. MLR*, vol. 3, pp. 993–1022, 2003.