

News Recommendation in the Wild: CWI's Recommendation Algorithms in the NRS Challenge

Alan Said
alan@cwi.nl

Alejandro Bellogín
alejandro.bellogin@cwi.nl

Arjen de Vries
arjen@acm.org

Centrum Wiskunde & Informatica
Amsterdam, The Netherlands

ABSTRACT

This work presents the recommendation algorithms deployed by the winning team (recomenders.net) in the ACM RecSys 2013 News Recommender Systems Challenge which ran during September 2013. The paper introduces concepts related to the challenge, an overview of the challenges related to news article recommendation and the differences to other recommendation domains, e.g. music, movies, etc. as well as gives a brief overview of the Open Recommendation Protocol (ORP) infrastructure used for the challenge.

Furthermore, the paper presents the recommendation framework and the algorithms implemented by the recomenders.net team in the challenge. Finally, the paper presents the results obtained by the team and discusses the metrics used to crown the winners of the challenge.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval - Information filtering; Relevance feedback; Retrieval models; Search process; Selection process; H.3.4 [Information Technology and Systems Applications]: Decision Support; H.5.1 [Multi-media Information Systems]: Evaluation/methodology

General Terms

Design; Experimentation; Measurement; Human Factors

Keywords

Evaluation; Benchmarking; Live Evaluation; Recommender Systems; News Recommendation; Content-based Recommendation; Recommender Performance

1. INTRODUCTION

Over the last two decades, recommender systems have become a common feature on most websites, whether in the context of social networks (Facebook, Twitter), movies (Netflix, IMDb), music (Last.fm, Spotify) or more recently news

(Google News, Yahoo! News). In this work, we describe a set of recommendation approaches deployed in the news article recommendation domain. Specifically, news recommendation in the scope of the ACM RecSys 2013 News Recommender Systems Challenge [9].

The challenge focused on providing live recommendations for readers of German news media articles by connecting recommendation algorithms to the Plista Open Recommendation Platform¹ (ORP) which has been described in more detail in [4, 6]. ORP consecutively connects the recommendation algorithms to real users, the full details of this are given in Section 2.1.

The concept of news recommendation, although not new, has not been as analyzed as some of the other domains listed above. Reasons for this include the lack of datasets as well as the lack of open systems to deploy algorithms in, making the ORP framework an attractive alternative for the research community to work on the news recommendation problem.

2. RECOMMENDING NEWS ARTICLES

The news recommendation context in the NRS Challenge differs significantly from more traditional recommendation contexts such as movies, movies, etc. Specifically, in the traditional recommendation domains, one can expect a *user profile*, i.e. the history of each user's played tracks or rated movies. This comes as a result of these domains attempting to create personalized recommendations, very often (e.g. Netflix, Movielens, Spotify, etc.) one of the basic system requirements is that the user creates a profile (user name) and consistently uses this profile when adding more ratings or listening to more music. However, when reading a news website, this information is generally not available. The user does not register, and thus creating a consistent user profile becomes problematic. For this reason, common collaborative filtering algorithms [5, 7] will perform poorly. Instead other means of recommendation become necessary, often at the cost of personalization. One of these recommendation models, the item-to-item model, identifies items similar to the candidate item to recommend.

Conceptually, this type of recommendation could be compared to a query being posted to a search engine, or finding similar documents in a library. Generally, the recommendation model becomes more similar to traditional information retrieval than recommendation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

NRS'13, October 13, 2013, Hong Kong, China.

¹<http://orp.plista.com>



Figure 1: The flow of messages sent between a user reading an article, the Plista server, and the participant’s recommendation algorithm.

2.1 Recommendation Infrastructure

The recommendation infrastructure used in the News Recommendation Challenge allowed participants to connect their own recommendation algorithms to Plista’s news delivery framework through an HTTP API. Recommendation requests (from Plista) and recommendations (from the participants) were sent between the contest server and the participant’s client as JSON messages. Fig. 1 shows the flow of messages between the participant’s client running a recommendation algorithm and the user reading a news article.

The recommendation request (1) is sent as the user starts reading an article from one of the providers. Plista’s servers forward the request to one of the participants (2a), while other participants are sent the impression (the user’s id and information on the news article being read) without a recommendation request (2b). This step ensures all participants have access to information on all user-news article interactions. Whenever a participant receives a recommendation request (3), his or her system must provide an answer (a list of recommended articles) within 200 ms, otherwise the recommendation will be served by Plista itself. Messages (5) and (6) are only sent when the reader clicks on the recommended news article. Fig. 2 shows an example of a news recommendation being delivered to a reader.

Recommended items must be from the same provider as where the recommendation request was issued. All articles are tagged as either “recommendable” or not. Whether or not an item is recommendable is decided by Plista and is communicated to the participants in the impression messages, i.e., message (2) in Fig. 1, and in update messages (sent by Plista when an item is created or updated in one of the providers). News articles typically remain recommendable while they are “fresh”, and become outdated after a few days.

The news providers served within the challenge are listed in Table 1. A further analysis on specific characteristics of click behavior and other features is available in [4, 6].

2.2 Implementation

The recommendation framework used for the News Recommender System Challenge is a modified version of the JAVA ORP Client² provided by Plista. The client was implemented as a standalone Maven project³. Apart from the client classes, the project also includes a default Recent Rec-

²<https://github.com/plista/orp-sdk-java>

³<https://github.com/recommenders/plistaclient>



Figure 2: An example of a recommendation being presented to the reader (in the widget on the bottom right) on Tagesspiegel. Note that there are multiple types of widgets, this is an example of a news recommendation widget containing only one recommendation, other widgets can contain more recommendations. [change image](#)

ommender (see below). Other recommendation algorithms can be added by specifying the recommendation class in the `plista.properties` file. The recommendation algorithms were implemented in a separate Maven project⁴ using the client as a dependency. The deployment instructions are outlined in the project’s README file⁵.

The recommendation infrastructure was divided in to these two projects in order to separate the Client/Server API implementation from the recommendation algorithms. The projects are described below, the recommender algorithms developed for the project are described in Section 2.3.

Plista Client. – The Plista Client (*plistaclient*) project (based on the JAVA ORP Client) manages the messages sent between the Plista server and the recommenders. Additionally, it contains an implementation of the Recent Recommender which is the default recommender should no others be given.

Plista Recommenders. – The Plista Recommenders (*plistaarecs*) project contains additional recommenders and utility classes used by the recommenders. The recommenders follow a common interface in order to be easily exchangeable and stackable (to create basic ensemble recommenders).

2.3 Recommendation Models

The recommenders deployed by the recommenders.net team in the News Recommender Systems Challenge were the following:

⁴<https://github.com/recommenders/plistaarecs>

⁵<https://github.com/recommenders/plistaarecs/blob/master/README.md>

Provider	Type	URL
CFO World	Business	http://www.cfoworld.de
CIO	IT News	http://www.cio.de
CNET.de	IT	http://www.cnet.de
Computerwoche	IT News	http://www.computerwoche.de
Gulli	IT & Games	http://www.gulli.com
Kölner Stadt-Anzeiger	News	http://www.ksta.de
Motor Talk	Automotive	http://www.motor-talk.de
Technologie- und Business-Nachrichten	IT	http://www.silicon.de
Sport 1	Sports	http://www.sport1.de
Tagesspiegel	News	http://www.tagesspiegel.de
Tecchannel	IT	http://www.tecchannel.de
Wohnen und Garten	Home & Garden	http://www.wohnen-und-garten.de
ZDNet	IT	http://www.zdnet.de

Table 1: The 13 news article providers, the types of news they deliver, and their URLs. Note that all providers deliver news articles in German.

Recent Recommender. – The recent recommender is based only on the recency of the articles. For each news provider, the most articles most recently with are recommended to the users, without any means of personalization or relatedness to the currently displayed article. Although the main criterion for this recommender is the article freshness, by updating its model for every interaction, implicitly it is also capturing the items’ popularity, a signal linked to the overall preferences of users in a community, which is generally perceived as a well-performing (although not personalized) recommender [2, 8]

Lucene Recommender. – The Lucene Recommender is a text retrieval system built on top of Apache Lucene⁶. The Lucene recommender contains an index of articles sent to the recommender through API messages. The articles are indexed on their title and preamble. On each recommendation request, the title and preamble of the article currently being read is issued as a query to the Lucene recommender. The list of recommendations contains the most similar articles in the index, provided that the items should be recommendable and the creation time of the articles should be within the previous 3 days.

Since all providers in the News Recommender Systems Challenge are from German providers (i.e. in German), the Lucene recommender uses a German Analyzer⁷ for stemming and stop word removal.

Category-based Recommender. – Inspired by the Lucene recommender, this content-based algorithm uses the article’s category – as provided by Plista – instead of the content. Items are either suggested from the same category (CR) or from a different category (OCR) in order to add increase diversity into the list of recommendations. Ranking of items within a category is based on two weighting factors: recency and popularity.

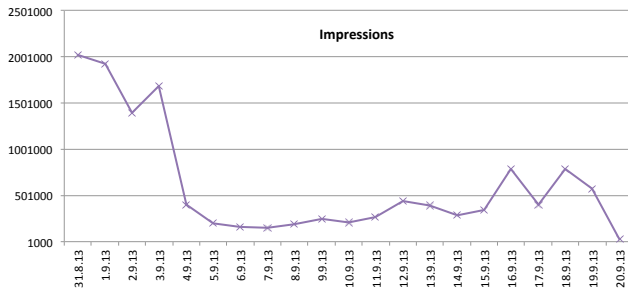
⁶Available at <http://lucene.apache.org/>

⁷https://lucene.apache.org/core/4_3_0/analyzers-common/org/apache/lucene/analysis/de/GermanAnalyzer.html

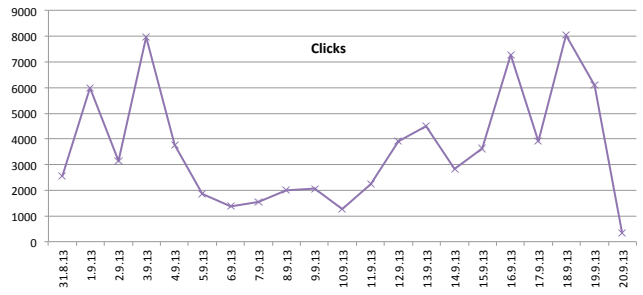
User Filter. – A user filter, even though it shares the basic structure of a recommender cannot be deployed as the main recommendation algorithm. Instead, this component filters out the articles previously observed by the current user, under the typical assumption in the literature that users have less interest in items already consumed. In the case of article news recommendation, however, it is not entirely clear if this assumption holds or not, since users may want to read the same article more than once. Additionally, they may expect the same articles to be recommended for a particular news item [3], however, within the ORP framework, this may be impossible to achieve, since different teams (i.e., different algorithms) may serve the recommendations for the same article and user at different moments in time.

Combined Recommenders. – The combined recommenders create a *stack* or cascade of two or more of the above recommenders [1]. Stacking in this context means that should the primary recommender not be able to recommend the requested number of items, a backup recommender (one step lower in the stack) is issued and finds additional articles to recommend. This stacking is particularly useful in the case of the Lucene recommender as it may not find related articles if the index is not sufficiently large, or if related articles are old (published more than 3 days ago from the time of the recommendation request). The combinations deployed in the challenge were:

- *Lucene Recommender with Recent Recommender (LRwRR)* – a Lucene Recommender with a Recent Recommender backup should the Lucene Recommender not be able to find related news articles.
- *Lucene Recommender with User Filter (LRwUF)* – a Lucene Recommender using a User Filter to exclude the news articles had already read.
- *Category-based Recommender with User Filter (PRCWUF)* – a Category-based recommender using a User Filter to exclude previously observed news articles, breaking ties using popularity and recency information.

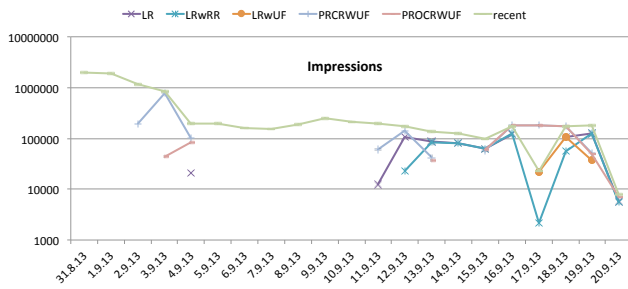


(a) Impressions per day

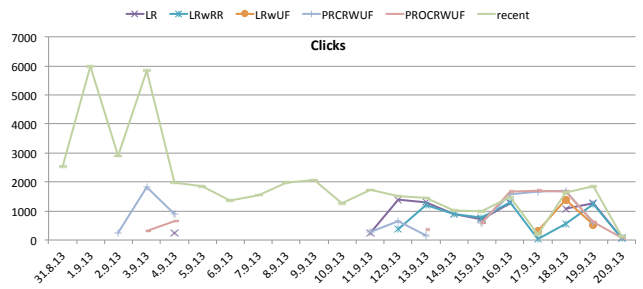


(b) Clicks per day

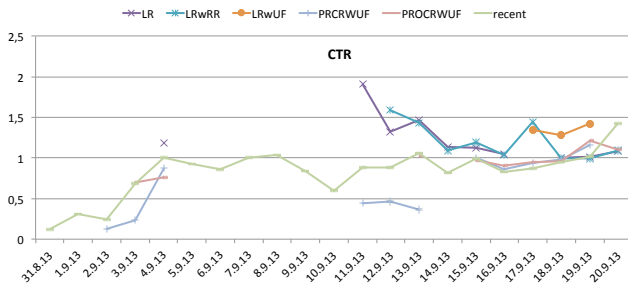
Figure 3: The combined number of impressions and clicks for all recommenders.net recommenders per day during the teams participation in the NRS Challenge. Note the high amount of impressions/clicks during the first days of the challenge even though only one recommendation algorithm was deployed at the time (see Fig. 4 for a comparison).



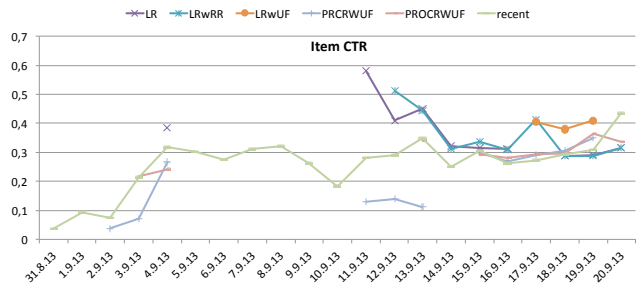
(a) The daily impression counts for the recommenders.net algorithms throughout the challenge.



(b) The daily click counts for the recommenders.net algorithms throughout the challenge.



(c) The daily CTR for the recommenders.net algorithms throughout the challenge.



(d) The daily *item* CTR for the recommenders.net algorithms throughout the challenge.

Figure 4: Temporal evolution of recommenders.net algorithms according to different metrics. No values indicate either that the algorithm had not been deployed yet or was not used (paused) during the time period.

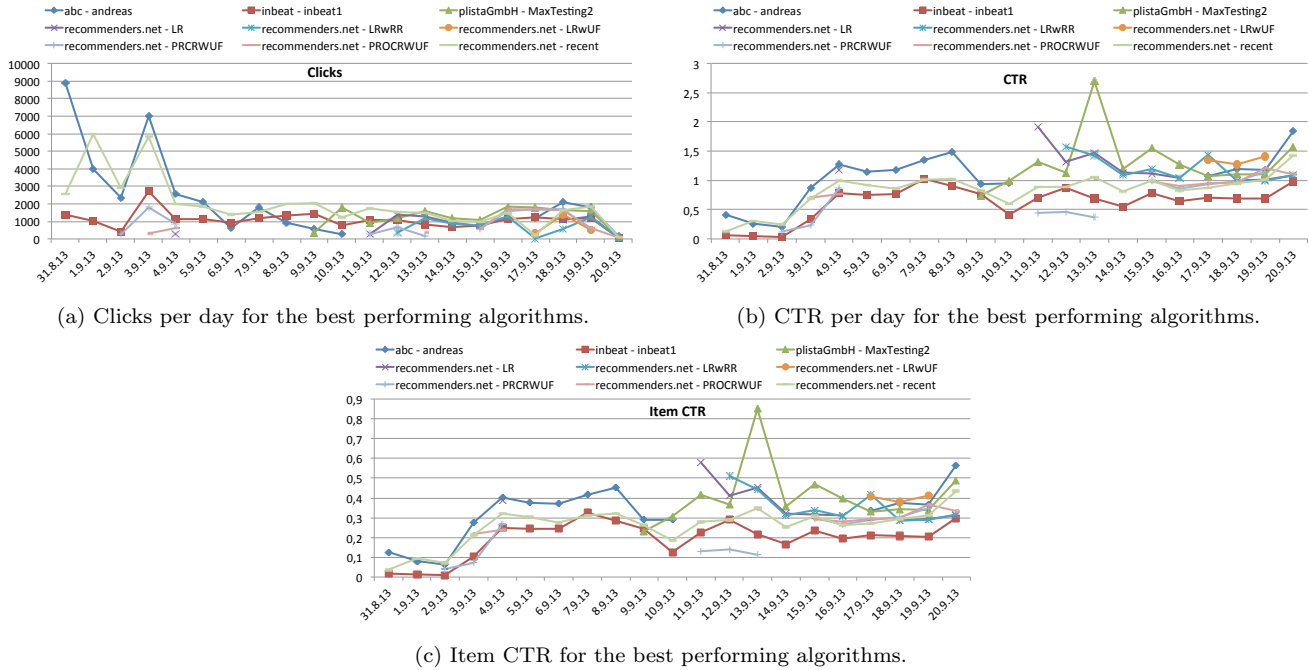


Figure 5: Clicks, CTR, and Item CTR for the recommenders.net algorithms and the other best performing ones in term of the final number of clicks (abc-andreas), impressions (inbeat-inbeat1), and average CTR (plista GmbH-Max Testing 2) excluding recommenders.net recommenders. Note that the first and third recommenders listed are part of the organisers, and thus, did not qualify as participants of the challenge.

- *Other Category-based Recommender with User Filter (PROCRWUF)* – a Category-based recommender using other categories different from the current article and a User Filter to exclude previously observed news articles, breaking ties using popularity and recency information.

2.4 Performance

In the NRS Challenge, the metric chosen to express the quality of a recommendation algorithm was the accumulated number of clicked recommendations per algorithm throughout the duration of the challenge. Although the metric represents an absolute quality over the entire time span of the challenge, it has some limitations. First, new recommenders developed in the scope of the challenge have a clear disadvantage with respect to other, previously deployed methods. Second, this measure does not take into account the number of recommendation requests per recommender, which may change due to technical issues or temporal constraints, as noted before. We think that a normalized version of this metric (per day or week) would be more appropriate to provide a fair comparison between recommenders, while at the same time it would not be favorable to methods successful only for short periods of time.

Specifically, the result of the early advantage is shown in Fig. 3 and Fig. 5, where we can also observe the number of impressions (and subsequently the number of clicks) was significantly higher during the first days of the challenge than after the initial phase. This is reflected in Fig. 3 between 31.8.13 and 4.9.13 where the only deployed algorithm

received nearly ten times as many daily impressions compared to the following days (note that the graphs show the accumulated number of impressions/CTR per day for all recommenders.net algorithm, which initially only consisted of the Recent Recommender).

In the figures presented in Fig. 4 we can observe the evolution of the different types of feedback provided by the NRS challenge organizers: number of impressions, number of clicks, CTR (Click Through Rate per recommendation widget⁸ even though the recommendation was successful), and Item CTR (Click Through Rate per number of recommender items⁹). Here we can observe that the Recent recommender is consistently obtaining more clicks (except for one occasion when it was taken offline). When looking at the CTR and Item CTR we find that the recommenders based on Lucene (LR, LRwRR, and LRwUF) consistently achieve higher values than the Recent recommender. This result suggests that should the recommenders have ran over a longer period of time, the number of clicks for these recommenders would have probably exceeded that of Recent.

Another observation to be made based on the available data is that the benefit of using the filter is not entirely clear. This is a result of the fact that the algorithms using the filter did

⁸Each recommendation widget can contain several recommended items, thus one clicked recommendation in a widget of four recommended items will result in a CTR of 25%

⁹In this case, the CTR is calculated by dividing the number of clicked recommendation by the number of recommendation widgets, i.e. even if a widget contains four items, one click per widget results in a CTR of 100%

not run for long periods of time, and thus suffered from cold start-related problems (probably due to a lower number of impressions received in the first hours). In particular, since the Category-based recommenders were not run without the filter, its effect cannot be properly analyzed for this method. The Lucene recommender with the filter on the other hand shows promising results, but further evaluations needs to be made before drawing any conclusions.

3. DISCUSSION

The News Recommendation Challenge provided a unique opportunity for researchers to perform large scale A/B testing in a production system. However, due to the metric chosen by the organizers, the algorithm crowned as the winner clearly did not represent the state of the art of recommender systems. It should however be stated that from a production-oriented perspective, a recommendation algorithm that is able to run over a long period of time, consistently delivering recommendations without failure can nevertheless be regarded as a successful one – even if its recommendation accuracy is not on par with untested (in a production environment) state of the art research methods.

We have however shown that the remaining algorithm used by the recommenders.net team did indeed outperform other participating teams (and organizers) not only in terms of raw clicks, but also in CTR and item CTR (see Fig. 5).

4. CONCLUSIONS AND FUTURE WORK

In this paper we have presented the recommenders.net team’s recommendation algorithms in the News Recommender Systems Challenge. We have given an overview of the architecture, algorithms and evaluation metrics used to crown the Challenge’s winning algorithm (Recent Recommender).

As future work, we intend to extend the presented algorithms by include personalized features based on information available on the user (e.g. taste across different domains). Additionally, we plan in including demographic features supported by the the ORP API like age, gender, income, etc. into to the recommendation algorithm to identify news articles of interest for each user.

Finally, we are currently developing a parallel evaluation based on the data logs collected in the NRS challenge to observe if the implemented recommender algorithms could have been improved by the user filters implemented during a later stage in the challenge.

5. ACKNOWLEDGMENTS

Parts of this work were carried out during the tenure of an ERCIM “Alain Bensoussan” Fellowship Programme. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n°246016.

6. REFERENCES

- [1] R. D. Burke. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.*, 12(4):331–370, 2002.
- [2] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys ’10, pages 39–46, New York, NY, USA, 2010. ACM.
- [3] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, Jan. 2004.
- [4] B. Kille, F. Hopfgartner, T. Brodt, and T. Heintz. The plista dataset. In *Proceedings of the Workshop on News Recommender Systems*, NRS ’13, New York, NY, USA, 2013. ACM.
- [5] P. Resnick and H. R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, Mar. 1997.
- [6] A. Said, J. Lin, A. Bellogín, and A. de Vries. A month in the life of a production news recommender system. In *Proceedings of the CIKM Workshop on Living Labs for Information Retrieval Evaluation*, LivingLab. ACM, 2013.
- [7] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, WWW ’01, pages 285–295, New York, NY, USA, 2001. ACM.
- [8] H. Steck. Item popularity and recommendation accuracy. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys ’11, pages 125–132, New York, NY, USA, 2011. ACM.
- [9] M. Tavakolifard, J. A. Gulla, K. C. Almeroth, F. Hopfgartner, B. Kille, T. Plumbaum, T. Brodt, and T. Heintz. Workshop and challenge on news recommender systems. In *Proceedings of the seventh ACM conference on Recommender systems*, RecSys ’13, New York, NY, USA, 2013. ACM.