Cold Start Problem: a Lightweight Approach at ECML/PKDD 2011 - Discovery Challenge

Leo Iaquinta and Giovanni Semeraro

University of Bari "Aldo Moro", v. Orabona 4, 70125 Bari, Italy
{iaquinta, semeraro}@di.uniba.it

Abstract. The paper presents our participation [5] at the ECML/PKDD 2011 - Discovery challenge for the task on the cold start problem. The challenge dataset was gathered from VideoLectures.Net web site that exploits a Recommender System (RS) to guide users during the access to its large multimedia repository of video lectures. Cold start concerns performance issues when new items and new users should be handled by a RS and it is commonly associated with pure collaborative filtering-based RSs. The proposed approach exploits the challenge data to predict the frequencies of pairs of cold items and old items and then the highest values are used to provide recommendations.

1 Background and Motivation

Recommender systems usually suggest items of interest to users by exploiting explicit and implicit feedbacks and preferences, usage patterns, and user or item attributes. Past behaviour is assumed to be useful to make reliable predictions, thus past data is used in the training of RSs to achieve accurate prediction models. A design challenge comes from the dynamism of real-world systems because new items and new users whose behaviour is unknown are continuously added into the system. As a consequence, recommendations may be negatively affected by the well-known *cold start* problem.

Cold start is commonly associated with pure collaborative filtering-based RSs. Particularly, item-based collaborative filtering techniques assume that items are similar when they are similarly rated and therefore the recommendations concern items with the highest correlations according to the usage evidence. A straight drawback is that new items cannot be recommended because there is not an adequate usage evidence.

Prediction involving *cold* items requires different approaches by comparing the performance for the predictions about *hot* items. This may be desirable due to other considerations such as novelty and serendipity. Thus evaluating the system accuracy on cold items it may be wise to consider that there is a trade-off with the entire system accuracy [7].

The first of the two tasks of the ECML/PKDD 2011 - Discovery Challenge[1] was focused on the cold start problem. The used dataset was gathered from VideoLectures.Net web site. Indeed, VideoLectures.Net

[1] http://www.ecmlpkdd2011.org/challenge.php

exploits a RS to guide users during the access to its large multimedia repository of video lectures. The main entities of the dataset are the lectures. They are described by a set of attributes and of relationships. The attributes are of various kind: for instance, *type* can have one value in a predefined set (lecture, keynote, tutorial, invited talk and so on); *views* attribute has a numeric value; *rec_date* and *pub_date* have a date value; *name* and *description* are unstructured text, usually in the same language of the lecture. The relationships link the lectures with 519 context events, 8,092 authors, and 348 categories. Each of these entities has its own attributes and relationships to describe taxonomies of events and categories. The lectures are divided into 6,983 for the training and 1,122 for the testing as cold items.
In addition, the dataset contains records about pairs of lectures viewed together (not necessarily consecutively) with at least two distinct cookie-identified browsers. This kind of data has a collaborative flavour and it is actually the only information about the past behaviour. The user identification is missing, thus any user personalization is eliminated. User queries and feedbacks are also missing.

2 Proposed Approach

To overcome the cold start problem in the approaches based on collaborative filtering, a common solution is to hybridize them with techniques that do not suffer from the same problem [1]. Thus, a content-based approach is used to bridge the gap between existing items and new ones: item attributes are used to infer similarities between items.
The proposed solution is obtained mainly by three steps: the data preprocessing, the model learning, and the recommendation.

Data pre-processing step starts with obtaining an in-memory object-oriented representation of provided data.
The main output of this step is a set of 20 numeric values describing the similarities between lectures of each pair in the training set. The used features involve language, description, recording and publication ages, conference, authors and their affiliations, and categories. More details are reported in [5].

Model learning step allows to obtain a prediction model for the frequency of a pair of lectures. The available data and the lightweight goal determined the selection of a linear model for the learning problem. Used features for different learned models are reported in [5].
The learned weights of a model are stored in a configuration file, with the option to add a boost factor for each weight to easily explore the feature influences beside the learned model. Fig. 1a and Fig. 1b report the values of the evaluation metric (Mean Average R-precision - MARp) for the recommendations using the model with all the available features when a boost factor is changed. Fig. 2 reports the evaluation metric values for the submitted solutions when the boost factors for the learned weight are changed: the submitted solutions always outperform the provided random baseline (MARp: 0.01949).

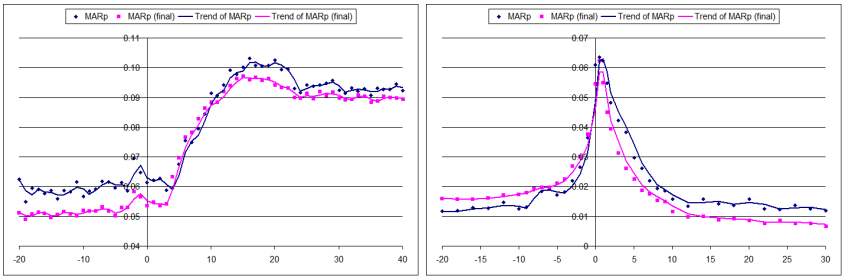

Fig. 1. Boost factor effects for "categoryBest" and "deltaRecAge"

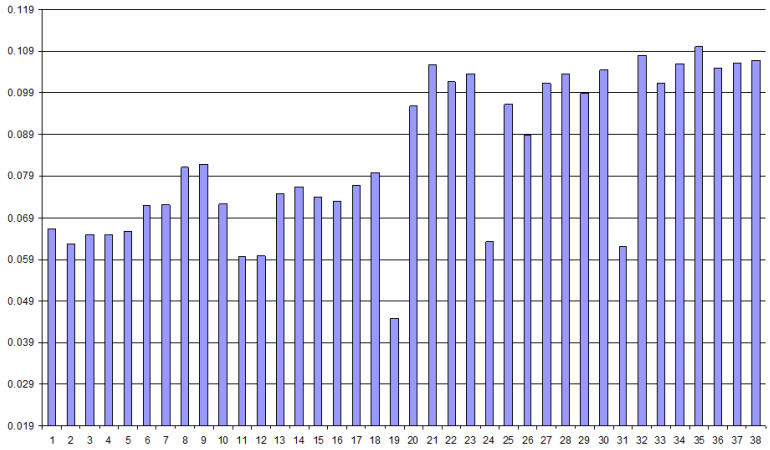

Fig. 2. Mean Average R-precision of submitted solutions

Recommendation step uses the in-memory representation of the pre-processing step and the learned weights to predict the pair frequency of an old item against each selected cold item. The highest values are used to provide recommendations.

2.1 Scale Problem

With the growth of the dataset, many recommendation algorithms are either slowed down or require additional resources such as computation power or memory. As such, it is often the case that algorithms trade other properties, such as accuracy or coverage, for providing rapid results for huge datasets [2]. The trade-off can be achieved by changing some parameters, such as the complexity of the model, or the sample size.
RSs are expected in many cases to provide recommendation on-line, thus it is also important to measure how fast does the system provides recommendation [3, 6]. Common measurement are the number of recommendations that the system can provide per second (the throughput of

the system) and the required time for making a recommendation (the latency or response time).
The developed components allow to complete the recommendation task for the 5,704 lectures in almost 85 seconds on a notebook with an Intel Core 2 at 2.0 GHz as CPU and 2GB of RAM, i.e., each new recommendation about 30 cold items over the selected 1,122 ones is provided in almost 15 milliseconds. Reasonably, a production server allows to reduce further the response time for new recommendations and a cache specifically devised for the recommendations allows to increase the throughput.

3 Conclusions

We have described the steps to achieve the submitted solution that outperforms the random baseline at the ECML/PKDD 2011 - Discovery challenge. The content-based hybrid approach allows to deal the cold start problem. In addition it chances to provide also serendipitous recommendations alongside classical ones [4]. Indeed the content-based item similarity can be used to spot potential serendipitous items as further trade-off with the entire system accuracy.
Finally, the scalability performance is considered as a primary requirement and a lightweight solution is pursued. The preliminary performance for the notebook execution is quite promising and some future directions for improving latency and throughput are sketched.

References

1. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction 12, 331–370 (2002)
2. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: Proc. of the 16th int. conf. on World Wide Web (WWW '07). pp. 271–280. ACM (2007)
3. Herlocker, J., Konstan, J.A., Riedl, J.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information Retrieval 5, 287–310 (2002)
4. Iaquinta, L., de Gemmis, M., Lops, P., Semeraro, G., Filannino, M., Molino, P.: Introducing serendipity in a content-based recommender system. In: Xhafa, F., Herrera, F., Abraham, A., Köppen, M., Bénitez, J.M. (eds.) Proc. of the 8th int. conf. on Hybrid Intelligent Systems (HIS-2008). pp. 168–173. IEEE Computer Society (2008)
5. Iaquinta, L., Semeraro, G.: Lightweight approach to the cold start problem in the video lecture recommendation. In: Šmuc, T., Antonov-Fantulin, N., Morzy, M. (eds.) Proc. of the ECML/PKDD Discovery Challenge Workshop. CEUR, vol. 770, pp. 83–94 (2011)
6. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: Proc. of the 10th int. conf. on World Wide Web (WWW '01). pp. 285–295. ACM (2001)
7. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 257–297. Springer (2011)