Estimate features relevance for groups of users

Stefano Cereda[1], Leonardo Cella[1], and Paolo Cremonesi[1]

Politecnico di Milano

Abstract. In item cold-start, collaborative filtering techniques cannot be used directly since newly added items have no interactions with users. Hence, content-based filtering is usually the only viable option left.
In this paper we propose a feature-based machine learning model that addresses the item cold-start problem by jointly exploiting item content features, past user preferences and interactions of similar users. The proposed solution learns a relevance of each content feature referring to a community of similar users. In our experiments, the proposed approach outperforms classical content-based filtering on an enriched version of the Netflix dataset.

1 Introduction and Related Works

Traditional Content Based recommender systems (CBF) need to represent users and items profiles in order to recommend similar items to those previously liked by users. Their main advantage is the capability of recommending previously unseen items, thus they solve the item-cold start issue. On the contrary, Collaborative Filtering (CF) algorithms usually reach better performance on predictions, especially with many interactions between users and items [1]. Their downside consists of the inability of recommending items with no previous interactions. Even if CBF approaches are able to solve the item cold-start problem, they are affected by at least two relevant limitations: recommended items tend to be too similar to previously rated items (over-specialization problem) and recommendations do not depend on preferences of similar users.

Some attempts to improve recommendation quality of CBFs consists of: Filtering methods and Embedded approaches. The former main drawback is that they do not take into account the ratings of users, therefore ignoring if the feature-based similarity between items is aligned with the user perception of similarity ([2], [3]). Embedded approaches perform feature weighting during the learning process and use its objective function to guide searching for relevant features. Instances of this methodology are: SSLIM [5] , UFSM [4] and Factorization Machines. The main drawback of embedded methods is the coupling between the collaborative and content components of the model. When used on datasets with unstructured user-generated features (e.g., tags) the noise from the features propagate to the collaborative part, affecting the overall prediction quality.

As a first solution to this problem, we have developed a machine learning algorithm whose aim is to compute global[1] feature weights based on a pure item

[1] in the sense that the relevance scores were shared by all the different users.

collaborative filtering approach. Its main objective was to embed in item features also information regarding user interests. In this research we propose an extension to this approach, the main contribution brought by this work is a general, straightforward wrapper to make content-based methods rate-aware and based on communities of similar users. Our experiments are conducted on the Netflix dataset in a version enriched with IMDB attributes. The experiments shown that the proposed solution outperforms classical pure content-based approaches.

2 Clustered Feature Weighting

Our objective is to recommend items from a set I to users in a set U. Items are described by the d-dimensional set of features F. User interactions are collected with the $\mathbf{R}^{|U|\times|I|}$ feedback matrix. Item features are described by the item features matrix $\mathbf{A}^{|I|\times|F|}$, $a_{ij} = 1$ iff item i has feature j. In general, user-cluster based recommender systems rely on a cluster-dependent similarity matrix $S_p^{|I|\times|I|}$, where p denotes the considered subset of users.
The predicted rate of user u, that belongs to the group p_u, for item i is computed as follows:

$$\hat{r}_{ui} = \frac{\sum_{j \in N_k^{p_u}(i)} r_{uj} s_{ij}^{p_u}}{\sum_{j \in N_k^{p_u}(i)} s_{ij}^{p_u}} \tag{1}$$

where: $s_{ij}^{p_u}$ is a local item-item similarity derived from the user subset p_u to which the target user u belongs and $N_k^{p_u}(i)$ is the set of k nearest neighbors of item i according to the similarity model of cluster p_u. Starting from this model, we would recommend the items whose predicted ratings are the largest. Feature weighting aims to derive a feature vector $\mathbf{w}^{p_u} \in \mathbb{R}^{|F|}$ such that each entry $w_l^{p_u} \in \mathbf{w}^{p_u}$ reflects the l^{th} feature relevance for the p_u subset of users. We define the weighted similarity $s_{ij}^{p_u}$ between items i and j for the cluster p_u as:

$$s_{ij}^{p_u} = \sum_{f \in F} w_f^{p_u} a_{if} a_{jf} = \langle \mathbf{w}^{p_u}, \mathbf{a}_i \odot \mathbf{a}_j \rangle \tag{2}$$

where $\mathbf{a}_i, \mathbf{a}_j \in \{0,1\}^{|F|}$ are the feature vectors of items i and j respectively and $\odot$ is the element-wise product. We propose to compute the feature weights by solving the following LSQ problem for each cluster of users p_u:

$$\underset{\mathbf{w}*^{p_u}}{\operatorname{argmin}} \sum_{i \in I} \sum_{j \in I \setminus \{i\}} ||s_{ij}^{cCF} - s_{ij}^{p_u}||^2 \tag{3}$$

More specifically, in our experiments we have adopted LSLIMr0 [6] as local similarity matrix S^{cCF} and CLUTO [7][2] to derive the user subsets p_u.
Since our goal is to learn a set of feature weights so that CBF similarities mimic as close as possible CF ones, there is no need to add a regularization term,

[2] this choice is based on the methodology followed in [6].

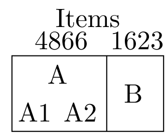
thus greatly simplifying the optimization. Experimental results confirmed this hypothesis.

When a new item is added to the catalog, we use $\mathbf{w}^{*p_u}$ to compute its weighted similarity w.r.t. the previously existing items. Then, it can be recommended to users belonging to subset p_u by using Equation 1. We call the proposed approach CLFW (Clustered Least-square Features Weighting).

3 Experimental Evaluation

Dataset. For our experiments, we used a version of the Netflix dataset enriched with structured and unstructured attributes extracted from IMDB. This dataset has 186K users, 6.5k movies and 6.7M ratings in 1-5 scale. The rating data is enriched with 16803 binary attributes representing various kinds of meta-information on movies such as director, actor, genres and user-generated tags[3]. To investigate the new-item scenario, we performed a 70/30 random hold-out split over items as shown in Figure 1. The sub matrix A has then been divided by moving 30% of positive (> 3) ratings into A2 and everything else in A1. A1 has then been used to compute LSLIMr0 and therefore to fit CLFW. When evaluating the warm-start scenario we used A1 as user profiles and A2 as ground truth, whereas for the cold-item we used the positive ratings of B as ground truth and A1 as user profiles.

Fig. 1. Dataset partitioning.



Baselines. As in the previous work we have used simple unweighted cosine similarity (Cos) and TF-IDF-weighted cosine similarity (CosIDF) as CBF baselines to evaluate the performance of CLSFW in both scenarios.

Performance Analysis In Table 1, we report the RMSE computed over predicted rates for different neighborhood sizes k in the new-item scenario. The warm-start scenario is instead represented by Table 2.

Table 1. RMSE evaluation for the new-item scenario.

k	CLFW	regCLFW	Cos	CosIDF
50	**1.321**	1.714	2.317	2.612
100	1.426	**1.409**	1.923	2.160
200	1.188	**1.182**	1.538	1.756

Table 2. RMSE evaluation for the warm-start scenario.

k	LSLIMr0	CLFW	regCLFW	Cos	CosIDF
50	**1.372**	1.838	1.962	2.634	2.901
100	**1.208**	1.522	1.619	2.209	2.477
200	**1.129**	1.291	1.362	1.821	2.058

We can state that in both scenarios CLFW consistently outperforms both the baselines on RMSE at any value of k. Moreover, in the warm-start scenario,

[3] the set of content features was significantly augmented with respect to our previous unclustered work.

it is nearly as good as LSLIMr0. We want to also highlight that CLFW differs from the other CBF baselines solely in the feature weighting scheme. Therefore, the improvement in performance must be due to a better feature weighting discovered by our approach. By comparing the CLFW column with the regCLFW one [4], we can observe that the regularization does not bring a performance improvement. This is reasonable and totally in agreement with our prediction. In fact, the data from which we are learning do not contain noise and, further more, the number of weights that we learn does not allow to overestimate the model complexity.

4 Conclusions and Future Work

With this research we investigated the possibility of deriving a user based feature weighting. We have presented ongoing results of an extended approach that solves the item cold-start issue by defining personalized features relevance. The ongoing development is focused in the usage of different personalization methodologies and extension to other datasets. Moreover, we are interested in combining this clustered approach with the, already developed, global one.

References

[1] Istvn Pilszy, Domonkos Tikk: Recommending new movies: even a few ratings are more valuable than metadata RecSys 09 Proceedings of the third ACM conference on Recommender systems , 93–100 (2009)

[2] Lops Pasquale , De Gemmis Marco , Semeraro Giovanni: Content-based recommender systems: State of the art and trends Recommender systems handbook , 73–105 (2011)

[3] Panagiotis Symeonidis, Alexandros Nanopoulos, Yannis Manolopoulos: Feature-Weighted User Model for Recommender Systems UM 07 Proceedings of the 11th international conference on User Modeling , Springer, 10.1007/978-3-540-73078-1-13 97 – 106 (2007)

[4] Elbadrawy Asmaa, Karypis, George: User-Specific Feature-Based Similarity Models for Top-n Recommendation of New Items, ACM Trans. Intell. Syst. Technol., 6, 33:1–33:20, (May 2015) 10.1145/2700495, ACM

[5] Ning Xia , Karypis George: Sparse Linear Methods with Side Information for Top-N Recommendations, Proceedings of the 21st International Conference on World Wide Web , WWW ’12 Companion, ACM 581 – 582, (2012)

[6] Christakopoulou Evangelia, Karypis George: Local Item-Item Models for Top-N Recommendation, 10th ACM Conference on Recommender Systems , RecSys16 67 – 74 , (2016)

[7] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

[4] which contains the results of our algorithm when the feature weights are computed adding an l_2 regularization term to Equation 3