Towards Time-Aware Semantic enriched Recommender Systems for movies

Marko Harasic, Pierre Ahrendt,Alexandru Todor, Adrian Paschke

FU Berlin

Abstract. With the *World Wide Web* moving from passive to active, the role of *recommender systems* as an aid to make decisions play a very prominent role. This enables its users to find new *items* of high personal interest, which they were previously unaware of. While traditional approaches have shown the generation of high quality recommendations, the additional use of background knowledge to describe the *items* and their *preferences* on a more granular level is still lacking. Furthermore, these approaches do not take into consideration the contextual information, wherein the dimension *'time'* plays a significant role. In this paper, we propose a new approach for recommending movies, which semantically enriches the process of generating recommendations by using a taxonomy derived out of different data sources from the LOD-Cloud. Furthermore, the paper also addresses the interplay between the rating behavior of the users and the dimension *'time'*.

1 Introduction

The amount of information in the world wide web grows continuously, causing users to be overwhelmed by the sheer volume of data. Users need a mechanism which aid them in their decisions when choosing the most useful *item*. *Recommender systems* collect their actions and then infer on their preferences. The system then generates an internal user representation, the user profile. With a search space reduction based his preferences, *items* of low personal interest will gradually be removed from the displayed *items*, improving his search results and simplifying the navigation.

Adomavicus et. al. [1] established a commonly used classification of *recommender* according to their approach. They distinguished them into *content based filtering*, into *collaborative filtering* and into *hybrid recommender*. *Content based filtering (CBF)* analyse items in advance, their attributes are extracted and a representation of the *item* is generated [6]. Afterwards, those *items* are recommended, which are most similar to the highest rated items of the user. In opposite to the item similarity of *CBF*, users with most similar preferences compared to the actual user form the neighbourhood in *Collaborative filtering (CF)*. Items are then drawn from these users [9]. *Hybrid recommenders* combine the two types of *recommenders* into a new system [2]. One approach is"collaboration via content" [7]. User profiles don't consist of ratings on *items*, they contain the *item* attributes. Especially when combined with background knowledge e.g. given by a taxonomy of attributes, *hybrid recommenders* outperform the other systems [11]. A special type is the *semantic recommender*, which uses background knowledge derived from taxonomies described in semantic web languages [8].

Recommendations based on traditional algorithms don't consider any contextual information like date, place, companion and mood. Considering the different contextual aspects, the dimension *'time'* can be identified as the most important one [5]. Obviously, as the users preferences change, the *'time'* information allows to track the evolution of his habits. Ding and Li proposed an exponential decay rate on the ratings [3]. Older ratings are considered as less significant, giving more recent ratings a higher importance. These newer ratings should reflect the preferences of an user in a higher degree.

2 Approach

The proposed algorithm follows the "collaboration via content" approach. User profiles contain the preferred attributes, derived from the item representations. They are further extended by using a attribute-taxonomy, which assigns the attributes to appropriate super-class relations, hence allowing to regard connections between different but related attributes. An exponential dampening factor is applied, to reflect the generality of attributes nearer to the root. Following the assumption, that old ratings have a lesser influence on the preferences than newer ones, they are weighted less by applying a decay rate.

The approach is formally described as follows: Let U be the set of users, I the set of *items* and R the systems rating scale as an totally ordered set of values. Then $ut : U \times I \rightarrow R$ is the utility function, that calculates the usefulness of a single *item* i to the user u. In order to create a representation $A(i)$ of an *item* $i \in I$, its direct attributes $D(i) := \{a_j | a_j \in \mathcal{T}\}$ are determined and weighted by the function $w(i, a_j)$, which specifies the significance of an attribute and follows the $TF - IDF$ metric. Beside the direct attributes a_j, the representation contains their indirect attributes $sa_d(a_j)$. This hierarchy of attributes forms the attribute-taxonomy $\mathcal{T}$. Based on its structure, the path $pa(a_j) = (sa_0(a_j) \cdots sa_d(a_j) \cdots, ra)$ for the attribute $sa_0(a_j) \equiv a_j$ to the root ra is determined. Each attribute is weighted according to its distance d from a_j by the height-function $h(sa(a_j)) = d^\gamma$, whereby the hierarchy influence parameter $\gamma > 0$ adjusts the influence of indirect attributes. To reflect the different weighting schema of direct and indirect attributes, the weighting function $w(i, a_j)$ takes possible multiple occurrences of the indirect attributes into account. The representation $A(i)$ of an element i is then defined as:

$$A(i) := \{(a_j, w(i, a_j))\} \cup \bigcup_{a_j \in D(i)} \{(sa_j, w(i, sa_j)) | sa_j \in pa(a_j)\} \quad (1)$$

$$\text{with } w(i, a) = \begin{cases} \frac{1}{|A(i)|} \cdot \log \frac{n(a)}{|I|} & \text{if } a \text{ is direct attribute} \\ \sum\limits_{a_j \in D(i) \wedge a \in pa(a_j)} h(a) \cdot w(i, a_j) & \text{otherwise} \end{cases} \quad (2)$$

$$\text{with } n(a_j) = \text{ is the number of } \textit{items}\text{, containing the attribute } a_j \quad (3)$$

Based on the *item* representations $A(i)$ of the users rated *items* $I(u)$, his profile $p(u) := \{(a_j, pr(u, a_j))|a_j \in A(i) \wedge i \in I(u)\}$ can be constructed. It contains the *items* attributes a_j and their preference weights $pr(u, a_j, t)$, which show the degree of interest on it at the point in *time* t. Profiles are generated in an iterative way. Starting with the oldest rated *item* $A(i)_0$, it is filled with attributes and their weights. Only those *items* are taken into account, which are rated higher than the mean of the systems rating scale. In order to reflect the preference changes over *time*, the *time* decay factor $0 \leq \alpha \leq 1$ is applied to the contained attributes when a new *item* $A(i)_t$) is rated at a later point at *time* t.

$$pr(u, a_j, t) = \begin{cases} (1-\alpha) \cdot pr(u, a_j, t-1) + \alpha \cdot w(i, j) & \text{if } a_j \in A(i)_t \\ (1-\alpha) \cdot pr(u, a_j, t-1) & \text{otherwise} \end{cases} \quad (4)$$

By applying the Pearson correlation between the user profiles, the similarity $sim(u, u')$ of two users u and u' can be calculated. Hereby are the preference weights $pr(u, a)$ of the common attributes $ca(u, u') := \{a|a \in p(u) \cap p(u')\}$ the used variables for the correlation. The prediction $ut(u, i)$ for a single *item* i is defined as the weighted sum of the k most similar users u', which have rated i, their similarities $sim(u, u')$ to u and their ratings $r_{u,i}$ on i. By using $ut(u, i)$, the recommendation set $RS(u, N)$ for a user u can be generated. It contains the N unseen *items*, which has the highest predicted rating.

$$sim(u, u') = \frac{\sum\limits_{a \in ca(u,u')} (pr(u, a) - \overline{pr(u)}) \cdot (pr(u', a) - \overline{pr(u')})}{\sqrt{\sum\limits_{a \in ca(u,u')} (pr(u, a) - \overline{pr(u)})^2} \cdot \sqrt{\sum\limits_{a \in ca(u,u')} (pr(u', a) - \overline{pr(u')})^2}} \quad (5)$$

$$ut(u, i) = \frac{\sum\limits_{u' \in N(u,k)} sim(u, u') \cdot r_{u',i}}{\sum\limits_{u' \in N(u,k)} \|sim(u, u')\|} \quad (6)$$

$$RS(u, N) := \{i_i | i_i \in I \backslash I(u) \wedge ut(u, i_j) \geq ut(u, i_{j+1}\} \quad (7)$$

3 Data

The *MovieLens* project provides a dataset, consisting of 3.500 users, who rated 6.000 movies with 1.000.000 ratings and was enriched with *URIs*, which identify the movies in DBpedia. A top-down approach starting with the DBPedia *URI* for *Category:Film* and going down the tree via *skos:broader of* until it reached the categories assigned to the movies was applied. The created taxonomy $\mathcal{T}$ consisted of 3.804 direct attributes a of the movies and furthermore 691 indirect attributes sa while having a height h of 12.

Other knowledge bases such as Freebase extract information form multiple data sources. One interesting aspect of Freebase is the way it categorizes films into genres by the property *film.film_genre*. The film.genre taxonomy contains a total of 700 genres which are related to each other via *child_genre relations*. In order to use the freebase film genres in our approach we, first had to extract all the genres and their relations, and then proceeded to infer a genre taxonomy where the genres are related to each other via subclass relations. This taxonomy contains 700 direct attributes a assigned to the movies and 238 indirect attributes sa which are contained inside the taxonomy $\mathcal{T}$ with height h of 4.

The DBpedia taxonomy contains more distinct attributes due to the higher number of film related categories in comparison to the genres in Freebase. Furthermore the Freebase taxonomy is rather shallow in comparison to the DBpedia taxonomy. The number of distinct indirect attributes represents the nodes that are not present in the direct *item* description but are generated due to taking into account the superclasses of those categories/genres. For example for movie that has cyberpunk as a genre we would also take sci-fi into account as a genre due to it being the superclass of cyberpunk.

4 Evaluation

To evaluate the system, a 5-fold cross validation was performed with each set containing 1.200 users. Each of the 5 evaluation runs use 4 different training sets U_t and the remaining set U_e is the test set. The system was first evaluated to find the optimal values for k, α and γ, using the $F_1@N$ metric [4]. To determine the accuracy for the system, the mean value of $F_1(u)@N$ for all users from the test set $u \in U_e$ is calculated by comparing the *items* of $RS(u,N)$ with the relevant ones R_u (i.e rated higher than the average) for the user u. rel_{i_u} is the binary relevance value of *item* i for user u.

$$P_u@N = \frac{1}{N}\sum_{i=1}^{N} rel_{i,u} \qquad R_u@N = \frac{1}{|R_u|}\sum_{i=1}^{N} rel_{i,u} \qquad F_1(u)@N = 2 \cdot \frac{P_u@N \cdot P_u@N}{P_u@N + P_u@N}$$

As baseline algorithm, the widely used item-to-*item* collaborative filtering algorithm using the adjusted cosine similarity [10] was implemented. While the dbpedia *recommender* used a taxonomy for enriching the profiles, the other systems only use the attributes taken from dbpedia respectively freebase without a taxonomy. Each of the following figures show the accuracy of the system in regard of the evaluated parameter as well as the accuracy of the baseline algorithm. All results are for recommendation sets with the size $N = 5$

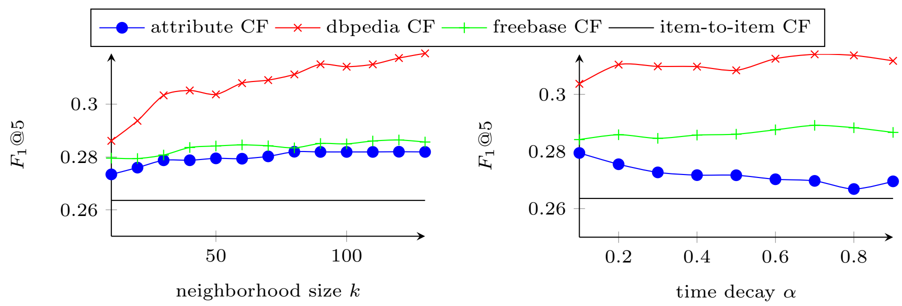

Fig. 1. Influence of the neighbourhood size k on the accuracy of the recommendations

Fig. 2. Influence of the time decay α on the accuracy of the recommendations

Different neighbourhood sizes k have a significant influence on the accuracy. Figure 1 shows the influence of k. Thereby achieved the different approaches their highest accuracy at $k \geq 120$. Smaller values for k result into a degradation of the accuracy. If the neighbourhood is to small, the few contained users have a strong impact on the predictions. Larger neighbourhoods don't have an influence on the accuracy. For the taxonomic approach was the value of k higher, because more users with shared general preferences can be considered, hence achieving a higher accuracy.

Figure 2 shows the impact of the time decay value α on the accuracy. Following the assumption, the time degradation of the preferences have an positive effect on the acurracy, achieving the maximum accuracy at $\alpha = 0.7$ for both taxonomic approaches. Higher values for α lowers the accuracy. All changes in taste are only reflected by a single model, and a mere decay of older preferences loose to much information. At a certain point in time in the past, the user rated an *item* differently than his current preferences would indicate. Lower values for α would not cover the shift of preferences over *time* for the user.

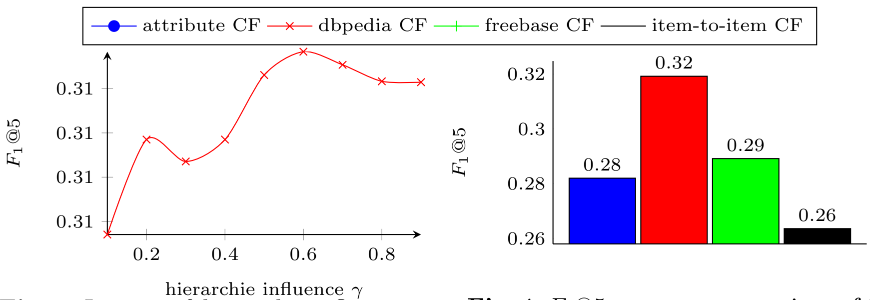

Fig. 3. Impact of hierarchy influence γ on the accuracy of the recommendations

Fig. 4. F_1@5 accuracy comparison of the different implemented algorithms

According to the height h of the used taxonomy, γ has to be adjusted accordingly. Figure 3 shows its impact on the accuracy of the system. While to small and to large values lowers the accuracy, as optimal value could $\gamma = 0.6$ be determined. When γ is chosen to low, the additional information gain by the taxonomy has a negligible effect and the system behaves similar to the concept only recommender. The taxonomy enables to find users, which has related preferences in movie genres, e.g. in dystophic sci-fi and in cyberpunk. But if the value for γ is set to large, the concepts nearer the root gain a higher influence resulting in a loss of precision in the preference representation of the users. Thereby users are treated equally, even if they are only loosely connected.

Figure 4 shows the accuracy according of the used F_1@5 metric. Each algorithm used the parameters for k, *alpha* and γ, where it achieved its highest accuracy. Since both tononomix approaches behaved similar according to the time influence, its value was set to $\alpha = 0.7$ and for the attribute only *recommender* $\alpha = 0.1$. For the taxonomy using system was $\gamma = 0.6$, and $k = 120$.

5 Conclusion and future work

The paper proposed a new approach for semantically enriching the process of recommending *items* by using a taxonomy derived out of the LOD-cloud. It outperforms the baseline algorithm to a significant level. In addition to this, there was a positive influence of the taxonomy on the accuracy of recommendation. As the DBpedia-*recommender* uses the same concepts as the attribute-*recommender* and follows the same paradigm, its accuracy is significantly increased by the use of the taxonomy. In a nutshell, the proposed approach seems to be well suited to work with the structure, given by the DBpedia-category taxonomy.

Following to our assumption, the degradation of each users' preference over his usage period has an positive impact on the accuracy. But some preferences tend to exist in present, even if they first were captured at the beginning. Therefore, the degradation of preferences has to be considered individually for each user and each of his preference. A deeper analysis on the influence of the aspect *'time'* on the proposed approach falls out of scope of this paper and is addressed independently in the forthcoming papers.

References

1. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 17(6):734–749, 2005.
2. Robin D. Burke. Hybrid web recommender systems. In *The Adaptive Web*, pages 377–408, 2007.
3. Yi Ding and Xue Li. Time weight collaborative filtering. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 485–492, New York, NY, USA, 2005. ACM.
4. Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22:5–53, January 2004.
5. Marius Kaminskas and Francesco Ricci. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, 6(2-3):89–119, 2012.
6. Ken Lang. Newsweeder: Learning to filter netnews. In *ICML*, pages 331–339, 1995.
7. Michael J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev.*, 13(5-6):393–408, 1999.
8. E. Peis, J. M. Morales del Castillo, and J. A. Delgado-López. Semantic recommender systems. analysis of the state of the topic. *Hipertext.net*, 6:(online), 2008.
9. Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW*, pages 175–186, 1994.
10. Badrul Sarwar, George Karypis, Joseph Konstan, and John Reidl. 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.
11. Cai-Nicolas Ziegler, Georg Lausen, and Lars Schmidt-Thieme. Taxonomy-driven computation of product recommendations. In *CIKM*, pages 406–415, 2004.