

A Hybrid Explanations Framework for Collaborative Filtering Recommender Systems

Shay Ben-Elazar Noam Koenigstein
Microsoft R&D, Israel
{ shaybe, noamko }@microsoft.com

ABSTRACT

Augmenting personalized recommendations with explanations is believed to improve users' trust, loyalty, satisfaction, and recommender's persuasiveness. We present a flexible explanations framework for collaborative filtering recommender systems. Our algorithm utilizes item tags to automatically generate personalized explanations in a natural language format. Given a specific user and a recommended item, the algorithm utilizes the user's personal information as well as global information (e.g., item similarities, metadata) in order to rank item tags based on their "explanatory power". The top tags are chosen to construct a personalized explanation sentence which helps shed light on the underlying recommender. Our system has been well received by both focus groups as well as in expert evaluations and is scheduled to be evaluated in an online experiment.

1. BACKGROUND AND INTRODUCTION

In the context of recommender systems, an explanation is a short textual description accompanying each recommended item to explain the recommender's choice for the user. The importance of such explanations for real world applications is increasingly recognized. Good explanations are believed to improve user trust, loyalty, satisfaction, and persuasiveness [6]. Experimental evidence has shown that explanations can improve users' acceptance [1].

The problem of explaining recommendations is not new. Several algorithms have been presented in the past and extensive surveys can be found in [3, 6]. Herlocker et al., [1] distinguishes between "white box" and "black box" explanation algorithms. For example, in neighborhood models [5] explanations can naturally follow from the recommendation algorithms: "We recommend *Shrek* because you watched *Toy Story*". This is a "white box" approach in which the explanations are based directly on the recommendation algorithm. However, explanations based on similarities are often simplistic and not very engaging. Furthermore, in matrix factorization based systems such as the Xbox recommender

system [2, 4], recommendations are based on a global optimization process and there is no clear method to associate an explanation item with the recommended item. We therefore turn to "black box" approaches which justify recommendations in ways which are related to the underlying recommender but in an indirect manner.

In this work we present a "black box" explanations framework based on tags (e.g., as in [7]) in order to compute personalized explanations in the form of "We recommend *Shrek* based on your history of watching humorous computer animated films featuring imaginary storyline fitting a family outing".

2. ALGORITHMIC OVERVIEW

Given a recommended item and its tags, we select a small subset of tags that serve best as explanations. These tags are then utilized to automatically generate the explanation sentence. Hence, explanations are computed in two stages: First, we compute scores for the candidate tags to quantify their ability to serve as explanations ("explanatory power"). Then, the tags are ranked and the top tags are chosen to constitute the explanation sentence by applying a predefined rule-set on tag categories.

The first stage integrates several modules which constitute scores for either the amount of personalization, quality of the explanation, or quality of the available tags in the context of the recommendation. Our framework allows plugging different modules and tuning their relative contribution with corresponding module weights. Each module is represented as a vector or a matrix and can be of one of the following types: a) Tag modules - vectors \mathbf{f}_k ; b) Item modules - vectors \mathbf{r}_i ; c) Items to tags relationship modules - matrices \mathbf{P}_j ; Our framework allows for multiple modules per each type above and we denote by k, i, j the indexes of the modules per type.

The scoring algorithm applies an ensemble approach which treats each module as an estimator and computes the weighted average. Let m, n be the number of tags and items, accordingly. Then,

$$\mathbf{s}^{m \times 1} = \left(\sum_i w_i \mathbf{r}_i^{1 \times n} \right) \times \left(\sum_j w_j \mathbf{P}_j^{n \times m} \right) \cdot \left(\sum_k w_k \mathbf{f}_k^{m \times 1} \right), \quad (1)$$

and \mathbf{s} is the resulting scores vector¹. Namely, $\mathbf{s}(l)$ is the score quantifying the "explanation power" of tag l (see figure 1). The weights w_i, w_j, w_k are used to tune the importance of

¹Note the interchanging use of both the dot product and cross product.

each of the modules. For example, by increasing w_k we can put more emphasis on the module \mathbf{f}_k .

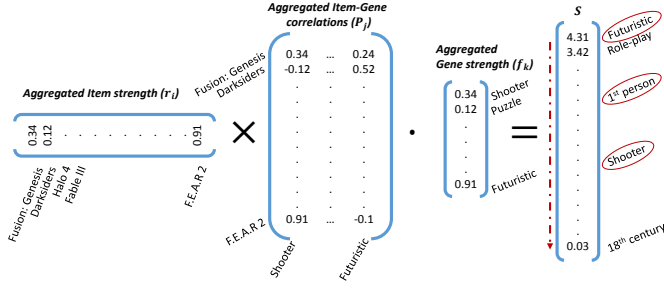


Figure 1: Aggregating modules for tag scoring. Top tags from different categories are chosen to constitute the explanation sentence.

Each item module $\mathbf{r}_i^{1 \times n}$ weights the items based on their relevance to the recommended item. For example, items similar to the recommended item are more important than items which are very different from the recommended item. A another example for an item module is a module that gives higher weight to items that the user likes more – e.g., Xbox games which are played more frequently should have a higher weight. Information from the different item modules is aggregated into an aggregated tag term $\sum_i w_i \mathbf{r}_i^{1 \times n}$.

The items to tags relationship modules $\mathbf{P}_j^{n \times m}$ are used to encode the affinity of items to tags. The aggregated items module $\sum_i w_i \mathbf{r}_i^{1 \times n}$ is projected onto the tags by multiplying by the aggregated relationship matrix $\sum_j w_j \mathbf{P}_j^{n \times m}$. This results in a tag weights vector which encodes both information from the item modules and the items to tags relationship modules. Some examples of items to tags modules are: 1) a binary matrix which indicates tags association to items; 2) weighted items to tags strengths which put more emphasis on rare tags.

The tag weights are adjusted by a point-wise multiplication with the aggregated tag modules $\sum_k w_k \mathbf{f}_k^{m \times 1}$. The tags modules score tags based on their ability to serve as a good explanation. For example, weighting tags based on categories e.g., movie director tags may have more “explanatory power” than plot tags. Additional examples of tag modules are: 1) prior multiplier for specific tags which serve as better / worst fit for explanations; 2) TF-IDF weights based on item descriptions; 3) a binary or weighted vector signifying the relevance to the recommended item.

In order to construct the explanation sentence we define a sentence structure which is composed of 3-4 “slots”. Each slot should be matched with a tag according to predefined tag categories. We rank the tags according to \mathbf{s} and select the highest ranking tags from the appropriate categories to fill in the slots. Note that the lowest ranking tags, those with negative scores but high absolute values, can be used as a negation e.g., “non-dramatic movies”. Finally, a rule-set is applied to place the appropriate connective words between the slots according to the tag categories. This results in a natural language personalized explanation sentence to be presented to the user.

3. RESULTS

We augment each recommended item with an automatically generated personalized explanation sentence (see figure 2). The explanations are based on tags that describe the recommended item and showed dominance in the user’s historical usage patterns. This results in a more meaningful and engaging explanation sentence. Our final system has been well received by both focus groups as well as in expert evaluations and is scheduled to be evaluated in an online experiment.



Figure 2: The resulting explanations showcased in the Xbox One recommender system.

4. REFERENCES

- [1] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, CSCW '00*, 2000.
- [2] Noam Koenigstein and Ulrich Paquet. Xbox movies recommendations: Variational bayes matrix factorization with embedded feature selection. In *Proc. 7th ACM Conference on Recommender Systems*, 2013.
- [3] David McSherry. Explanation in recommender systems. *Artif. Intell. Rev.*
- [4] Ulrich Paquet and Noam Koenigstein. One-class collaborative filtering with random graphs. In *Proceedings of the 22nd international conference on World Wide Web, WWW '13*, pages 999–1008, 2013.
- [5] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW*, 2001.
- [6] Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop, ICDEW '07*, 2007.
- [7] Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: Explaining recommendations using tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces, IUI '09*, pages 47–56, 2009.