Modeling and Predicting User Actions in Recommender Systems

Tural Gurbanov
Free University of Bozen-Bolzano
Bozen-Bolzano, Italy
tgurbanov@unibz.it

Francesco Ricci
Free University of Bozen-Bolzano
Bozen-Bolzano, Italy
fricci@unibz.it

Meinhard Ploner
ProSiebenSat.1 Media SE
Unterföhring, Germany
meinhard.ploner@prosiebensat1digital.de

ABSTRACT

Many collaborative filtering recommender systems collect and use users' explicitly entered preferences in the form of ratings for items. However, in many real world scenarios, this form of feedback can be difficult to obtain or unavailable (e.g., news portals). In this case recommendations must be built by leveraging more abundant implicit feedback data, which only indirectly signal users' preferences or opinions. A record in such datasets is a result of an *action* performed by a user on an item (e.g., the item was clicked or viewed). State-of-the-art implicit feedback recommender systems predict whether the user will act on a target item and interpret this prediction as a discovered preference for the item. These models are trained by observations of user actions of one single type. For instance, they predict that a user will watch a video using a dataset of observed video watch actions. In this paper we conjecture that multiple types of user actions may be jointly exploited to predict one target type of actions. We present a general prediction model (MMF - Multiple action types Matrix Factorization) that implements this conjecture and we illustrate some practical examples. The empirical evaluation of MMF, which was conducted on a large real world dataset, shows that using multiple actions is beneficial and it can outperform a state-of-the-art implicit feedback model that uses only the target action data.

Keywords

Collaborative filtering; implicit feedback; matrix factorization; user actions

1. INTRODUCTION

Collaborative filtering (CF) is a popular recommendation technique where user ratings/likes are analyzed to predict missing ratings/likes. Many CF systems collect and use users' explicitly entered preferences: ratings for items. However, in many real world scenarios ratings can be difficult to obtain or unavailable (e.g., news portals). Hence, recommenders have been also built by leveraging more abundant implicit feedback data, such as the log of item/page views or item purchases [9], which only indirectly signal users' preferences or opinions [10].

We say that an item is labeled (positively) by a user if the user performed an "action" on the item (e.g., the item was viewed or purchased). We note that it might be erroneous to conclude that since the user acted on an item, then the user liked it (implicit feedback). Besides, analysing user actions, it is even harder to reliably identify which items a user does not like. In fact, usually there is no explicitly negative feedback/action, and the items not acted by the user cannot be directly assigned to the "negative" label. Furthermore, data is usually extremely sparse and unbalanced [15]: only a small part of the items is labeled with actions.

Two general approaches to treat implicit feedback data in CF have been proposed: weighting [3, 12] and sampling [12]. The first one treats all the unlabeled items as negatively labeled examples and the (label) prediction model is based on weighted low-rank approximation [8]. The second uses a sampling strategy to draw some unlabeled items and treats them as negative examples. The source data is thus split into several parts where each part has the same positive examples and different negative and unlabeled examples. The final prediction model is built by employing all these data parts. Both approaches use a non-negative matrix of weights, which are called confidence weights, that indicate the confidence in the negativeness or positiveness of the assigned label. We stress that these systems can only predict whether an item is labeled positively, i.e., a user will act on it. When these systems recommend the positively predicted items it is implicitly assumed that the user likes the items on which she has acted.

In this paper we consider, simultaneously, multiple types of user actions, e.g., user clicks, user views, and user bookmarks. We introduce a method that reinforces the confidence weights, either for the positive or negative labels of an action (e.g., view), by leveraging observations of another type of action (e.g., click). The proposed method takes advantage of the existence of a range of action types to improve the confidence in certain observations for a given target action. For example, if a user opened the description of a

UMAP '16, July 13-17, 2016, Halifax, NS, Canada

DOI: http://dx.doi.org/10.1145/2930238.2930284

movie, read it, watched a movie trailer but did not watch the movie, then the method increases the confidence that the user will not watch this movie. This information, as we will show in this short article, may help to develop a more accurate prediction model of the video views.

We note that this research initiated when we observed the existence of a correlation between the presence of an action of type A on an item with the absence of an action of another type B. We conjectured that this type of correlations may be generalized and leveraged to overcome the data sparsity problem and to better predict users' actions.

We conducted an experimental evaluation of the proposed method using a dataset provided by the mass media company P7S1[1]. We designed a novel, multiple actions, prediction model (MMF), that extends Hu's et al. [3] weighted CF model for implicit feedback (IMF). MMF incorporates multiple action types in order to better predict a target user action. The experiment results show that the proposed model outperforms IMF.

The rest of the paper is organized as follows. We introduce state-of-the-art user actions prediction techniques in Section 2. We then describe our general model and provide concrete examples of it in Section 3. Section 4 describes the data and Section 5 contains the results of our experiments. Finally, in Section 6 we conclude the paper and indicate some future work.

2. STATE OF THE ART

We start this section by introducing a state-of-the-art model (IMF) for predicting user actions that follows the approach proposed in [3]. Later we will generalize this model by considering observations for a range of action types.

The input data for the action prediction model is a $|U| \times |I|$ non-negative matrix, whose entries a_{ui} count the number of observed actions of user u on item i, e.g., the number of clicks on an item. If no action is observed then a_{ui} is zero.

The model predicts whether the user will act on a target item, hence an indicator function p, is introduced:

$$p_{ui} = \begin{cases} 1 & a_{ui} > 0 \\ 0 & a_{ui} = 0 \end{cases}$$

The model generates predictions $\hat{p}_{ui}$ for the items i where $p_{ui} = 0$. In fact, the absence of observed actions of the user u on item i does not mean that the user will not act on the item in the future. Hu et al. [3] interprets this prediction $\hat{p}_{ui}$ as a (predicted) preference for the item. For instance, in case the action is a click, a predicted click is interpreted as a predicted preference. We will not follow this interpretation and we will simply assume that the model predicts user actions.

IMF introduces also a confidence function:

$$c_{ui} = c(a_{ui})$$

which indicates the confidence in the value assigned to p_{ui} on the base of the observation a_{ui}. In fact, if $a_{ui} > 0$, hence $p_{ui} = 1$, the model should strongly rely on this information, as it is based on real observations of user u actions on the item i. Conversely, we should not force the model to rely on information that u never acted on i ($a_{ui} = 0$): the user may act on the item in the future, that is what we want to predict.

[1]http://www.prosiebensat1.com/

Furthermore, it might be the case that the user was never exposed to the item and therefore did not act on it. Hence, the confidence that $p_{ui} = 0$, for items where $a_{ui} = 0$ (i.e., when we miss observations) must be much smaller than the confidence that $p_{ui} = 1$ for items where $a_{ui} > 0$. Otherwise, if we strongly believe that $p_{ui} = 0$ when $a_{ui} = 0$, then there will be no point in building a prediction model $\hat{p}_{ui}$ for the items where $a_{ui} = 0$.

In IMF the target action prediction $\hat{p}_{ui}$ is computed using matrix factorization (MF). Each user u and item i is associated with an f-dimensional factors vector $x_u \in \mathbb{R}^f$ and $y_i \in \mathbb{R}^f$ respectively. The predicted value is computed by the inner product of these two vectors: $\hat{p}_{ui} = x_u^T y_i$. The factor vectors are computed by minimizing the following cost function:

$$\min_{x*,y*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2) \quad (1)$$

The constant λ weights the regularizer term to avoid model's overfitting [13]. The value of λ is data-dependent and determined by cross-validation.

In addition to IMF, other approaches have incorporated user- or item-related information to the confidence values to improve the prediction system results [5, 14, 2]. There are also models that predict a user's immediate next action by observing sequential patterns of actions [7, 16]. Moreover, existing models can be employed as components in more complex boosting [6] or hybrid [1, 11] techniques.

3. GENERAL MODEL AND EXAMPLES

In this paper we conjecture that the prediction accuracy of a model can be improved by leveraging the observation of a range of action types, i.e., not only the actions that are to be predicted. For instance, the information whether a user bookmarked a page may be useful to predict if the user will read a page.

We now describe our general action prediction model called MMF (Multiple action types Matrix Factorization). Suppose we have a dataset of $d+1$ types of users' actions. The variable a_{ui}^j counts or measures the observations of an action of type j performed by a user u on an item i. We assume that there is a target action $j = 0$ (predicted), and non-target actions are instead those for $j \in \{1, \ldots, d\}$ (predictive).

The indication function of the target action p_{ui}^0 and the confidence c_{ui}^0 of the target action are now functions of all the observations for all the action types:

$$p_{ui}^0 = p(a_{ui}^0, a_{ui}^1, \ldots, a_{ui}^d) \qquad c_{ui}^0 = c(a_{ui}^0, a_{ui}^1, \ldots, a_{ui}^d)$$

As in IMF, in order to build the prediction model for the target action by using MF techniques one should minimize the cost-function in Equation 1. The difference, with respect to IMF, is due to the usage of different confidence and indicator matrices: c_{ui}^0 and p_{ui}^0. Moreover, the optimization process must consider all the possible u, i pairs, and the huge number of terms prevents us from applying the most direct optimization techniques, such as stochastic gradient descent. Thus, an alternating least squares (ALS) optimization procedure was employed.

Below we illustrate two applications of MMF related to P7S1's video streaming service 7TV[2] that provides access

[2]http://www.7tv.de/

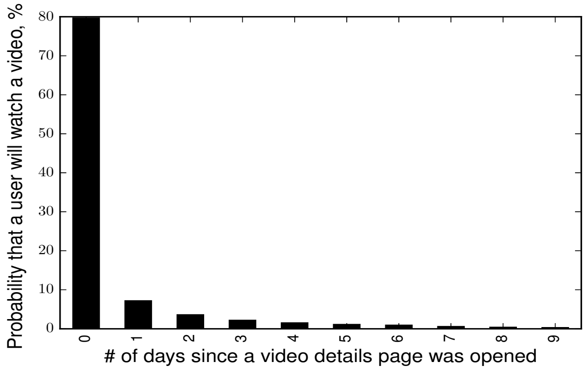

Figure 1: The distribution of probability that a user will start to watch a video after Δ days since the video details page was opened

to videos and TV-shows (a set of videos). We consider three types of actions: a user opened a video details page, a user stopped to watch a video and a user marked a TV-show as "watch later".

3.1 Predicting video views with open details actions

In the first example MMF predicts that a user will watch a video by also leveraging the observations of two additional action types: the user opened the video details page and the user stopped to watch a video.

Let $v_{ui} \in [0, 1]$ denote the percentage of video i viewed by user u. The service obtains this data when a user stops to watch a video. A positive example of the user target action (a user started to watch a video) is observed when $v_{ui} > 0$. So, in case only this target action type is used, the indicator function p can be defined as:

$$p_{ui} = \begin{cases} 1 & v_{ui} > 0 \\ 0 & v_{ui} = 0 \end{cases}$$

In general, as v_{ui} grows, we have stronger confidence that $p_{ui} = 1$. At the same time, for items where we missed observations the confidence that $p_{ui} = 0$ should be much smaller. A feasible choice for c_{ui} would be:

$$c_{ui} = 1 + \alpha v_{ui}$$

The parameter $\alpha \geq 0$ is a meta parameter that must be optimized. The value of α is data-dependent and determined by cross-validation. Increasing α places (proportionally) more weight on the non-zero observations while decreasing α places more weight on non-observed items.

Analyzing the data we noted that if a user opened a video details page but did not watch the video then there is a high probability that the user will not watch the video in the future. The probability is proportional to the time passed since the video details page was opened (Fig. 1).

This observation can be used to increase the confidence in the negative examples ($p_{ui} = 0$) of the target action. The more days are passed since the video details page was opened (the video still has not been watched), the higher must be the confidence that p_{ui} will stay equal to 0.

Let us denote with o_{ui} the variable indicating that a user has visited ($o_{ui} = 1$) or not ($o_{ui} = 0$) a video details page. Moreover, let Δ_{ui} be the number of days since the visit action, $o_{ui} = 1$, was observed, and $b^{-\Delta_{ui}}$ is the function that approximates the probability distribution displayed in Fig. 1. Then the confidence value is assumed to be as follow:

$$c_{ui} = \begin{cases} 1 + \alpha v_{ui} & v_{ui} > 0 \\ 1 + \beta(1 - b^{-\Delta_{ui}}) & v_{ui} = 0 \textit{ and } o_{ui} = 1 \\ 1 & v_{ui} = 0 \textit{ and } o_{ui} = 0 \end{cases}$$

Here α, β and b are data-dependent meta parameters that should be tuned. By cross-validation we found that $\alpha = 80$, $\beta = 8$ and $b = 10$ produce good results in our experiments. The parameter b should be greater than 1.

3.2 Predicting TV-show views with watch later actions

In the second example MMF predicts that a user will watch a TV-show. Even in this scenario we use observations of two action types: a user watches a video, and a user marks a TV-show as "watch later". We denote with $s_{ui} \in [0, 1]$ the percentage of a TV-show i viewed by a user u. The positive examples of the user target action (a user started to watch a TV-show) are observed when $s_{ui} > 0$.

In contrast to the previous example, the "watch later" action type can be here used to reinforce the confidence in positive examples of the target action. In fact, in 86% of cases a user watched a TV-show after she marked it as "watch later". Let $l_{ui} \in \{0, 1\}$ indicate the absence/presence of item i in the set of TV-shows that u would like to watch later. According to our data analysis, if a user adds an item to the "watch later" set, then there is a high probability that a user will watch such an item. Hence, the indicator function p can be defined as:

$$p_{ui} = \begin{cases} 1 & s_{ui} > 0 \textit{ or } l_{ui} = 1 \\ 0 & s_{ui} = 0 \textit{ and } l_{ui} = 0 \end{cases}$$

At the same time, we conjecture that the confidence that a user will watch a TV-show should be increased when a user marks it as "watch later". So, the confidence value can be defined as:

$$c_{ui} = \begin{cases} 1 + \alpha s_{ui} + \gamma l_{ui} & s_{ui} > 0 \textit{ or } l_{ui} > 0 \\ 1 & s_{ui} = 0 \textit{ and } l_{ui} = 0 \end{cases}$$

As in the previous example, α and γ are meta parameters that must be determined by cross-validation. In our experiments, setting $\alpha = 40$ and $\gamma = 5$ was found to produce optimal results.

4. DATASETS

The data[3] provided by mass media company P7S1 contains information about the interaction of users with the on-line video streaming service 7TV for one month. A user interacts with the service through a set of devices. For this reason we consider a device as a user, i.e., users are anonymised. Table 1 shows some statistics about the devices (users), videos and TV-shows (items), observations of different action types.

A "Video details opened action" record indicates that a user opened a video details page. A "Video view stopped

[3]http://www.inf.unibz.it/~gurbanov/vod_data.html

Table 1: Statistics of the used datasets

Collection name	Size
Devices	128k
Videos	23k
TV-shows	347
Video view stopped action	0.8M
Video details opened action	1.6M
Watch later action	1.1k

Table 2: "Video view stopped action" record

User	Video	Watched percent	Timestamp
4E6E8D...	380934	0.84	1438387203

action" record contains the maximal percentage of a video viewed by a user. A "Watch later action" record refers to a TV-show that a user marked to watch it later. All action records contain the time when the action was performed. For the "Video view stopped action" type the *timestamp* field stores the last time a user watched the video (Table 2).

"Videos", "TV-shows" and "Devices" in Table 1 count only items that are found in the actions collections. The sparsity of the indicator and confidence values matrices are approximately 99.9%.

5. EVALUATION

In our experimental evaluation we have compared MMF with IMF. We used the actions from the first three weeks as training sets, while the more recent actions was used for validation and testing purposes. We indicate with T the test set of target actions, and the observation of an action in T is denoted with a^t_{ui}. In our datasets users often perform a target action multiple times. Because it is more interesting for a user to be recommended with items that she has not acted recently, or that she is not aware of [3], we removed from the test set T all the actions a^t_{ui} belonging to the target type that were already presented in the training set.

The models generate for each user a list of items sorted in descending order by their predicted value $\hat{p}_{ui}$. Since the test set contains only positive observations of the target action, precision based metrics are not appropriate to measure the quality of the compared models, hence, we used a recall measure: Mean Percentage Ranking (MPR) [4, 3]. MPR evaluates a user's satisfaction with an ordered list of items.

$$MPR(T) = \frac{\sum_{u,i \in T} a^t_{ui} rank_{ui}}{\sum_{u,i \in T} a^t_{ui}}$$

We denote by $rank_{ui}$ the percentile ranking of the item i within the ordered list of all the items predicted to receive the target action by user u. $rank_{ui} = 0\%$ indicates that i is predicted on top of the items that will be acted by u, while $rank_{ui} = 100\%$ indicates that i is predicted at the bottom of the ranked items for u. Lower values of MPR are better, as they indicate that users actually acted ($a^t_{ui} > 0$) on the clips that are ranked on top. Notice that a random ranking will have MPR equal to 50%. Thus, $MPR < 50\%$ indicates an algorithm better than random.

We also calculated Recall at 20 (R@20). We denote with S^{20}_u the top 20 items in the ranked list of the predicted items that will be acted by a user u, and with A^T_u the items on which the user u actually acted and present in T. If U_T denotes the users in T then R@20 is defined as follow:

$$R@20(T) = \frac{1}{|U_T|} \sum_{u \in U_T} \frac{|S^{20}_u \cap A^T_u|}{|A^T_u|}$$

Table 3: Comparison of IMF and MMF (f=20)

	Metric	IMF	MMF	Improvement
Video views prediction	MPR	12.9	12.6	2.3%
	R@20	17.1	17.3	1.1%
TV-show views prediction	MPR	27.3	24.4	10.6%
	R@20	38.8	44.8	15.5%

We evaluated the two example applications presented above for different number of latent factors $f \in \{10, 20, 30, 40\}$. We anticipate that for all these choices MMF outperforms IMF. Hence, we present the results only for the optimal number of latent factors, $f = 20$, which was found by cross validation on the training data.

In the first example the models predict that a user will watch a video, while in the second one the models predict that a user will watch a TV-show. IMF is trained by using only the target action data, while MMF uses also the "watch later" action type. As can be seen in Table 3, MMF outperforms IMF in terms of MPR and R@20.

We must observe that in the second example the "watch later" actions let us add positive examples of the indicator function, thereby increasing prediction recall. Conversely, in the first example by considering the "open video details" actions we can only tune the confidence of negative examples of the indicator function. That explains why the improvement brought by MMF is larger in the second case.

6. CONCLUSIONS

In this work we conjectured that multiple types of user actions, e.g., user clicks, user views, and user bookmarks, can be jointly exploited to predict one target type of actions. Hence, we have presented a general model (MMF) that predicts a target user action by leveraging information about actions of multiple types and we illustrated two practical applications of this model. The empirical evaluation of MMF, which was conducted on a large real world dataset, showed that using multiple actions is beneficial and can outperform a state-of-the-art implicit feedback model that uses only the target action data. This conclusion was not obvious since not all available action types may be relevant for the prediction of a target action. Moreover, the presence of latent relations between action types might have a negative impact on the prediction results.

The incorporation of multiple action types into the prediction requires the definition of specific indicator and confidence functions and the optimization of multiple meta parameters. Currently these functions have been identified heuristically, after an exploratory analysis of the data. As future work, we want to set up data mining solutions that can automatically discover the hidden relationships between actions of different types and compute the indicator and confidence functions. The ultimate goal of our research is the application of the proposed model as a component of a movie recommender system, hence we will also derive a preference model from the learned actions prediction model.

7. REFERENCES

[1] K. Choi, D. Yoo, G. Kim, and Y. Suh. A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electron. Commer. Rec. Appl.*, 11(4):309–317, July 2012.

[2] Y. Fang and L. Si. Matrix co-factorization for recommendation with rich side information and implicit feedback. In *Proceedings of the 2Nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, pages 65–69, New York, NY, USA, 2011. ACM.

[3] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 263–272, Washington, DC, USA, 2008. IEEE Computer Society.

[4] C. C. Johnson. Logistic matrix factorization for implicit feedback data. *Advances in Neural Information Processing Systems*, 27, 2014.

[5] Y. Li, J. Hu, C. Zhai, and Y. Chen. Improving one-class collaborative filtering by incorporating rich user information. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 959–968, New York, NY, USA, 2010. ACM.

[6] Y. Liu, P. Zhao, A. Sun, and C. Miao. A boosting algorithm for item recommendation with implicit feedback. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1792–1798. AAAI Press, 2015.

[7] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Using sequential and non-sequential patterns in predictive web usage mining tasks. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 669–672, 2002.

[8] N. S. Nati and T. Jaakkola. Weighted low-rank approximations. In *In 20th International Conference on Machine Learning*, pages 720–727. AAAI Press, 2003.

[9] E. R. Núñez Valdéz, J. M. Cueva Lovelle, O. Sanjuán Martínez, V. García-Díaz, P. Ordoñez de Pablos, and C. E. Montenegro Marín. Implicit feedback techniques on recommender systems applied to electronic books. *Comput. Hum. Behav.*, 28(4):1186–1193, July 2012.

[10] D. Oard and J. Kim. Implicit feedback for recommender systems. In *in Proceedings of the AAAI Workshop on Recommender Systems*, pages 81–83, 1998.

[11] V. C. Ostuni, T. Di Noia, E. Di Sciascio, and R. Mirizzi. Top-n recommendations from implicit feedback leveraging linked open data. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 85–92, New York, NY, USA, 2013. ACM.

[12] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 502–511, Washington, DC, USA, 2008. IEEE Computer Society.

[13] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. *Proceedings of KDD Cup and Workshop*, pages 39–42, 2007.

[14] L. Peska and P. Vojtás. Evaluating the importance of various implicit factors in e-commerce. In X. Amatriain, P. Castells, A. P. de Vries, C. Posse, and H. Steck, editors, *Proceedings of the Workshop on Recommendation Utility Evaluation: Beyond RMSE, RUE 2012, Dublin, Ireland, September 9, 2012*, volume 910 of *CEUR Workshop Proceedings*, pages 51–55. CEUR-WS.org, 2012.

[15] F. Ricci, L. Rokach, and B. Shapira. *Recommender Systems Handbook.* Springer US, New York, NY, USA, 2nd edition, 2015.

[16] D. Rosaci. Cilios: Connectionist inductive learning and inter-ontology similarities for recommending information agents. *Information Systems*, 32(6):793 – 825, 2007.