Considering Words in Tag Recommendation
Master Project

Daniel K. I. Weidele*

University of Constance
Dept. Computer and Information Science
daniel.weidele@uni-konstanz.de

Abstract. When it comes to tag recommendation fundamental approaches often rely on bare user per item tagging data. However, newer approaches highlight that it may be reasonable to consider a certain degree of context such as content of items within factorization models. As part of my master program this study aims on an improvement of pairwise interaction tensor factorization (PITF) by adding a word indicator to its model (PITF-WI).

Keywords: Tag Recommendation Pairwise Interaction Tensor Factorization Word Indicator PITF PITF-WI Model

1 Introduction

Recommender systems shall support the user in decision making while interacting with large information space. Typically they propose a rating $\hat{y}_{u,i}$ of an item by the user, e.g. based on the user's social environment, the user's rating history, item similarities or context (ref. [1], [5]). Given a specific user u, item i this means, that these models aim to estimate the real rating $y_{u,i}$ as best as possible:

$$y_{u,i} - \hat{y}_{u,i} \approx 0$$

With *tag recommendation* (ref. [2], [6]) there exists a more specialized task in the field of recommender systems. For example web 2.0 applications more and more make use of tagging as it constitutes another component in personalization of content, which is valuable in terms of advertising and accessibility to the user. Moreover tag recommendation may provide little insight into item semantics by tag interpretation as the item is not only traded by a numerical rating. For tag recommendation the basic task is to predict the top N tags t a user u would give to the item i:

$$top(u, i, N) := \arg\max_{t \in T}^{N} \hat{y}_{u,i,t}$$

with N being the number of tags in the returned list.

* supervised by Prof. Dr. Steffen Rendle

Obviously this also implies a ranking in the returned list:

$$\forall t_i, t_j \in T_N : \hat{y}_{u,i,t_i} \leq \hat{y}_{u,i,t_j} \Leftrightarrow t_i \trianglelefteq t_j$$

In the following sections this study will outline two existing approaches, present a new hybrid approach and evaluate them against fundamental baselines in tag recommendation.

2 Related Work

Typical datasets for tag recommendation contain categorical information about which user u has posted which tags t to items i. Therefore the historical input dataset can be formulated as the triple subset of the three-dimensional tensor

$$S \subseteq U \times I \times T$$

In this section there is presented a factorization-based approach solely relied on this kind of input data, as well as a similarity-based approach that also takes into account information about the content (words) of items.

2.1 Pairwise Interaction Tensor Factorization (PITF [7])

The first approach to be outlined is a factorization-based method which is additionally aware of implicit tagging data by drawing pairwise interactions of tags given a specific user and item. PITF [7] is a special form of Tucker Decomposition (see fig. 1) with the core set to the diagonal tensor and explicit modeling of two-way interactions between users and tags or items and tags [1]. Moreover the approach is adoptable by factorization machines as presented in [4].

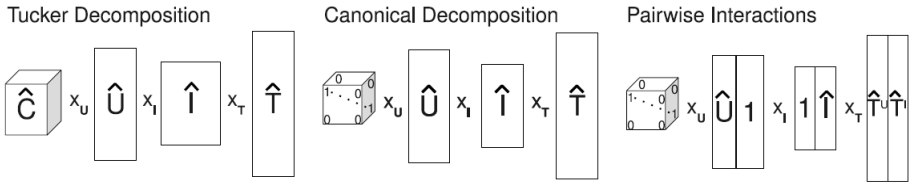

Fig. 1. From Tucker Decomposition to PITF (from [7]).

[1] Note that the user ↔ item interaction induced by general PITF gets off for ranking with BPR as optimization criterion.

Training Data For machine learning it is of interest to take negative tag posting into account, as well. Therefore the authors improve the trivial approach of declaring all triples of $(U \times I \times T) \backslash S$ as the negative class by modeling the training data as D_S with quadruples of the form (u, i, t_A, t_B) with $t_A, t_B \in T$, for which the following constraint holds:

$$D_S := (u, i, t_A, t_B) : (u, i, t_A) \in S \wedge (u, i, t_B) \notin S)$$

The example (fig. 2) outlines the drawing of such quadruples, e.g. for the red column there is being set up each tag from 1 to 5 in x- and y-axis. Then there is marked the pairwise interaction between two tags for the given post of user u on item i by setting the cell value to '+' ('-') if the tag in x-axis (y-axis) occurred more often. For quadruples with no or equal occurrence of both tags this interaction is said to be *missing* and can be marked by '?'. By comparing its columns the resulting $T \times T$ matrix implies that for item i the user u prefers tag t_1 to tag t_2 and to t_3, as well as t_4 to t_2 and t_3:

$$t_1 >_{u,i} t_2 \qquad t_1 >_{u,i} t_3 \qquad t_4 >_{u,i} t_2 \qquad t_4 >_{u,i} t_3$$

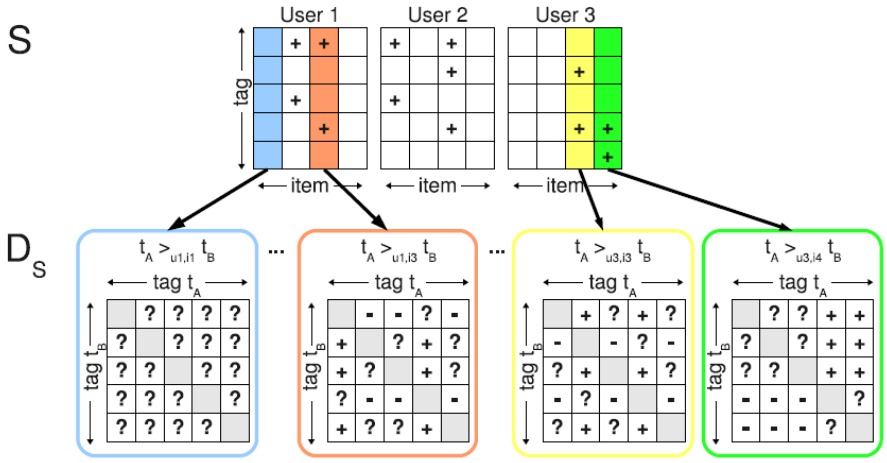

Fig. 2. Visualization of D_S (from [7]).

Bayesian Personalized Ranking For their factorization model the authors derive a generic Bayes-based optimization criterion for personalized ranking:

$$\text{BPR-Opt} := \sum_{(u,i,t_A,t_B) \in D_S} \ln \sigma(\hat{y}_{u,i,t_A,t_B}(\Theta)) - \lambda_\Theta ||\Theta||_F^2$$

with Θ as the model parameters, λ_Θ as the regularization constant and σ as a simple logistic curve defined as

$$\sigma(x) := \frac{1}{1+e^{-x}}$$

Furthermore the authors present a generic learning algorithm (see alg. 1) for a model parameter Θ that randomly picks quadtruples of D_S and performs stochastic gradient descent on it.

Algorithm 1 LearnBPR [7]

Input: D_S, Θ
Output: $\hat{\Theta}$
  initialize Θ
  repeat
    draw(u, i, t_A, t_B) uniformly from D_S
    $\Theta \leftarrow \Theta + \alpha \frac{\Delta}{\Delta\Theta}(\ln \sigma(\hat{y}_{u,i,t_A,t_B}) - \lambda_\Theta ||\Theta||_F^2)$
  until convergence
  return $\hat{\Theta}$

Finally the above gradient for updating the model parameter Θ can be estimated by

$$(1 - \sigma(\hat{y}_{u,i,t_A,t_B})) \cdot \frac{\Delta}{\Delta\Theta} \hat{y}_{u,i,t_A,t_B} - \lambda_\Theta \Theta$$

PITF Model Parameters As PITF explicitly factorizes into *user* $\leftrightarrow$ *tag* and *item* $\leftrightarrow$ *tag* interaction the model equation has been formalized as

$$\hat{y}_{u,i,t} = \sum_f \hat{u}_{u,f} \cdot \hat{t}^U_{t,f} + \sum_f \hat{i}_{i,f} \cdot \hat{t}^I_{t,f} \tag{1}$$

with the model parameters

$$\hat{U} \in \mathbb{R}^{|U| \times k} \qquad \hat{I} \in \mathbb{R}^{|I| \times k} \qquad \hat{T}^U, \hat{T}^I \in \mathbb{R}^{|T| \times k}$$

2.2 Probabilistic Modeling for Personalized Tag Prediction (PMPTP [8])

With PMPTP [8] Yin et al. propose an approach that takes into account item content in form of words, as well. Moreover their probabilistic model is built to explicitly reflect a user's environment by using similarities to other users. PMPTP can informally be described for one user u as a similarity-weighted result of the environmental probabilities combined with the user-centered probabilities of occurrences of tags $t \in T$ and words $w \in W_i$ for an item i:

$$\hat{y}_{u,i,t} = \sum_{W_i} \log(\sum_{u_k \neq u} p_{u_k,u} \cdot P(w|t,u_k) + \alpha \cdot P(w|t,u)) + \log(\sum_{u_k \neq u} p_{u_k,u} \cdot P(t|u_k) + \alpha \cdot P(t|u)) \quad (2)$$

with similarity of users u_i, u_j estimated by

$$p_{u_i,p_j} := \frac{\text{sim}(u_i,u_j)}{\sum_{u_k} \text{sim}(u_i,u_k)}$$

and

$$\text{sim}(u_i,u_j) := \frac{V_{u_i} \cdot V_{u_j}}{|V_{u_i}| \times |V_{u_j}|}$$

the cosine similarity between tag distributions V_u of users.

The environmental similarities for a user almost sum up to 1, s.t. the *ego-centric* effect α can be denoted by

$$\alpha := 1 - \sum_{u_k \neq u} p_{u_k,u}$$

For the rest of equation 2 there is the probability $P(t|u)$ received by counting how often user u makes use of tag t among all his tags:

$$P(t|u) := \frac{n_{u,t}}{\sum_{t_i \in T} n_{u,t_i}}$$

Finally there is $P(w|t,u)$ denoting the probability of a word to occur on fixed tag t and user u:

$$P(w|t,u) := \frac{X_{u,t,w}}{\sum_{i \in I} X_{u,i,t}} \text{ with } X = \begin{cases} 1 \text{ , if observed} \\ 0 \text{ , else} \end{cases}$$

To avoid zero probabilities the authors propose to make use of additive smoothing, which has been applied in this study as well.

3 Motivation

To get an overview about which ingredients hold more or less information on modeling a tag recommender in this section there are investigated 4 different trivial baselines which lead to the motivation of this study.

3.1 Formalization of Baselines

MP-T Baseline The *Most Popular Tag* recommendation model is characterized by predicting always the same tags to each user on any item, i.e. the recommended tags are the top used tags among all users on any items:

$$\hat{y}_{u,i,t} := \sum_{u_k \in U} \sum_{i_k \in I} n_{u_k,i_k,t}$$

MP-TU Baseline With MP-T not being sensitive to personalization the *Most Popular Tag per User* model overcomes this property by simply recommending the top tags that have been applied by the user to predict for in the past:

$$\hat{y}_{u,i,t} := \sum_{i_k \in I} n_{u,i_k,t}$$

MP-TI Baseline Instead of personalization per user the *Most Popular Tag per Item* recommendation model personalizes on the item dimension by predicting the top tags for an item given among all users:

$$\hat{y}_{u,i,t} := \sum_{u_k \in U} n_{u_k,i,t}$$

MP-TW Baseline Lastly PMPTP (see 2.2) implicitly suggests to solely investigate *Most Popular Tags per Word* as well. Therefore $\hat{y}_{u,i,t}$ is modeled as the sum of probabilities of the tag to appear for the item's words (independently of user or item) and normalize by the number of words in the item:

$$\hat{y}_{u,i,t} := \frac{\sum_{w \in W_i} \frac{n_{t,w}}{n_w}}{|W_i|}$$

3.2 Comparison of Baselines

The evaluation of the baselines has been applied on post-core dataset of ECML PKDD 2009 Discovery Challenge [3]. Figure 3 shows the result in the form of F-measure, precision and recall of the 4 different approaches. While personaliztaion in general seems to be a good idea there is much less information among the user dimension as tagging behavior seems to relate more to the item than to

the user itself. With MP-TW outperforming all other baselines, e.g. due to overcoming sparseness by not regarding an item as a fixed categorical value but as a patchwork of content, which may be partially known from other items' content, it is reasoned to consider words for tag recommendation.

The following section will outline the introduction of a new model parameter to PITF in order to support consideration of words.

4 Approach

The comparison of baselines already indicates that considering words in tag recommendation may significantly improve prediction quality. Therefore the PITF-model presented in section 2.1 shall be extended by a word indicator to further investigate the practical impact of it.

Algorithm 2 Optimizing the PITF-WI model with LearnBPR (based on [7])

Input: $D_S, \hat{U}, \hat{I}, \hat{W}, \hat{T}^U, \hat{T}^I, \hat{T}^W$
Output: $\hat{U}, \hat{I}, \hat{W}, \hat{T}^U, \hat{T}^I, \hat{T}^W$
  draw $\hat{U}, \hat{I}, \hat{W}, \hat{T}^U, \hat{T}^I, \hat{T}^W$ from $N(\mu, \sigma^2)$
  repeat
    draw(u, i, t_A, t_B) uniformly from D_S
    $\hat{y}_{u,i,t_A,t_B} \leftarrow \hat{y}_{u,i,t_A} - \hat{y}_{u,i,t_B}$
    $\delta \leftarrow (1 - \sigma(\hat{y}_{u,i,t_A,t_B}))$
    for $f \in 1, ..., k$ **do**
      $\hat{u}_{u,f} \leftarrow \hat{u}_{u,f} + \alpha(\delta \cdot (\hat{t}^U_{t_A,f} - \hat{t}^U t_B, f) - \lambda \cdot \hat{u}_{u,f})$
      $\hat{i}_{i,f} \leftarrow \hat{i}_{i,f} + \alpha(\delta \cdot (\hat{t}^I_{t_A,f} - \hat{t}^I t_B, f) - \lambda \cdot \hat{i}_{i,f})$
      $\hat{t}^U_{t_A,f} \leftarrow \hat{t}^U_{t_A,f} + \alpha(\delta \cdot \hat{u}_{u,f} - \lambda \cdot \hat{t}^U_{t_A,f})$
      $\hat{t}^U_{t_B,f} \leftarrow \hat{t}^U_{t_B,f} + \alpha(-\delta \cdot \hat{u}_{u,f} - \lambda \cdot \hat{t}^U_{t_B,f})$
      $\hat{t}^I_{t_A,f} \leftarrow \hat{t}^I_{t_A,f} + \alpha(\delta \cdot \hat{i}_{i,f} - \lambda \cdot \hat{t}^I_{t_A,f})$
      $\hat{t}^I_{t_B,f} \leftarrow \hat{t}^I_{t_B,f} + \alpha(-\delta \cdot \hat{i}_{i,f} - \lambda \cdot \hat{t}^I_{t_B,f})$
      $s_{\hat{w}_{w,f}} \leftarrow 0$
      for $w \in W_i$ **do**
        $s_{\hat{w}_{w,f}} \leftarrow s_{\hat{w}_{w,f}} + \hat{w}_{w,f}$
        $\hat{w}_{w,f} \leftarrow \hat{w}_{w,f} + \alpha(\delta \cdot \frac{1}{|W_i|} \cdot (\hat{t}^W_{t_A,f} - \hat{t}^W t_B, f) - \lambda_W \cdot \hat{w}_{w,f})$
      end for
      $\hat{t}^W_{t_A,f} \leftarrow \hat{t}^W_{t_A,f} + \alpha(\delta \cdot \frac{1}{|W_i|} \cdot s_{\hat{w}_{w,f}} - \lambda_W \cdot \hat{t}^W_{t_A,f})$
      $\hat{t}^W_{t_B,f} \leftarrow \hat{t}^W_{t_B,f} + \alpha(-\delta \cdot \frac{1}{|W_i|} \cdot s_{\hat{w}_{w,f}} - \lambda_W \cdot \hat{t}^W_{t_B,f})$
    end for
  until convergence
  return $\hat{U}, \hat{I}, \hat{W}, \hat{T}^U, \hat{T}^I, \hat{T}^W$

4.1 PITF-WI

The formalization of the PITF (see equation 1) already outlined the basic idea to explicitly model user and tag or item and tag interactions. It can be simi-

larly extended to support word and tag interaction, which is pointed out by the following equation

$$\hat{y}_{u,i,t} = \sum_f \hat{u}_{u,f} \cdot \hat{t}^U_{t,f} + \sum_f \hat{i}_{i,f} \cdot \hat{t}^I_{t,f} + \sum_f \sum_{w \in W_i} \frac{\hat{w}_{w,f} \cdot \hat{t}^W_{t,f}}{|W_i|} \tag{3}$$

with additional model parameters

$$\hat{W} \in \mathbb{R}^{|W| \times k} \qquad \hat{T}^W \in \mathbb{R}^{|T| \times k}$$

To train the additional model parameters the following gradients can be applied

$$\frac{\Delta \hat{y}_{u,i,t}}{\Delta \hat{w}_{w,f}} = \sum_{w \in W_i} \frac{\hat{t}^W_{t,f}}{|W_i|} \qquad \frac{\Delta \hat{y}_{u,i,t}}{\Delta \hat{t}^W_{t,f}} = \sum_{w \in W_i} \frac{\hat{w}_{w,f}}{|W_i|}$$

which results in the algorithm PITF-WI (see alg. 2) based on *LearnBPR* (ref alg. 1) formulated in analogy to [7].

4.2 Runtime Analysis

In comparison to PITF the runtime for learning the PITF-WI model increases by factor $|W_i|$ to $O(k \cdot |W_i|)$. However, for prediction of $\hat{y}_{u,i,t}$ PITF-WI can be enhanced by reformulating the word indicator model from

$$\frac{1}{|W_i|} \sum_f \sum_{w \in W_i} \hat{w}_{w,f} \cdot \hat{t}^W_{t,f} \qquad \text{to} \qquad \frac{1}{|W_i|} \sum_f \hat{t}^W_{t,f} \cdot \sum_{w \in W_i} \hat{w}_{w,f}$$

by defactorizing the sum of products to receive $\sum_{w \in W_i} \hat{w}_{w,f}$ independently of t thus being able to precalculate it in several ways depending on the application.

5 Evaluation

For evaluation again the post-core dataset of ECML PKDD 2009 Discovery Challenge [3] has served for application of PITF, PMPTP and PITF-WI. Figure 4 shows the outcoming F-measure, precision and recall also compared to MP-TW for convenience. With suitable $\lambda_W = 0.005$ PITF-WI marginally improves PITF for top 2 tag prediction, which is not worth the expediture of factor 8 in runtime (about 5 days until convergence with $k = 64$ and $\alpha = 0.01$). For the post-core dataset with its property to have each user, item and tag appearing at least twice the word indicator obviously does not necessarily come into play, as there is not too much sparseness among the items. PITF already seems to gain enough information to have its model trained and perform well on the dataset. In comparison PMPTP does not strike and seems to be very much driven by MP-TU. However, one must note that the authors outline further tweaks which can improve the performance of the approach.

To evaluate on more sparseness some of the models have also been applied to the cleaned dump dataset of ECML PKDD 2009 Discovery Challenge (see fig. 5). Due to its long runtime there could not be identified the best regularization for PITF-WI on this dataset, yet. However, even with potential overfitting ($\lambda_W = 0.0$, but also not yet converged) PITF-WI (yellow) improves the F-measure of PITF (turquoise) by around 5%.

6 Conclusion

We have seen four baselines in tag recommendation that motivated the introduction of a word indicator to the presented PITF model. The resulting model PITF-WI could keep up and partially overtake in quality depending on the sparseness of the dataset. Moreover there could be shown that deploying the word indicator on PITF should be carefully considered as the training time may dramatically increase. It has also been pointed out that plain baselines like MP-TW are good indicators to understand modeling, and frequently they even pose a hurdle to be passed first. Finally PITF-WI can be interpreted as a step towards robustness as it combines the benefits of PITF with the ability to bridge over sparseness within one model.

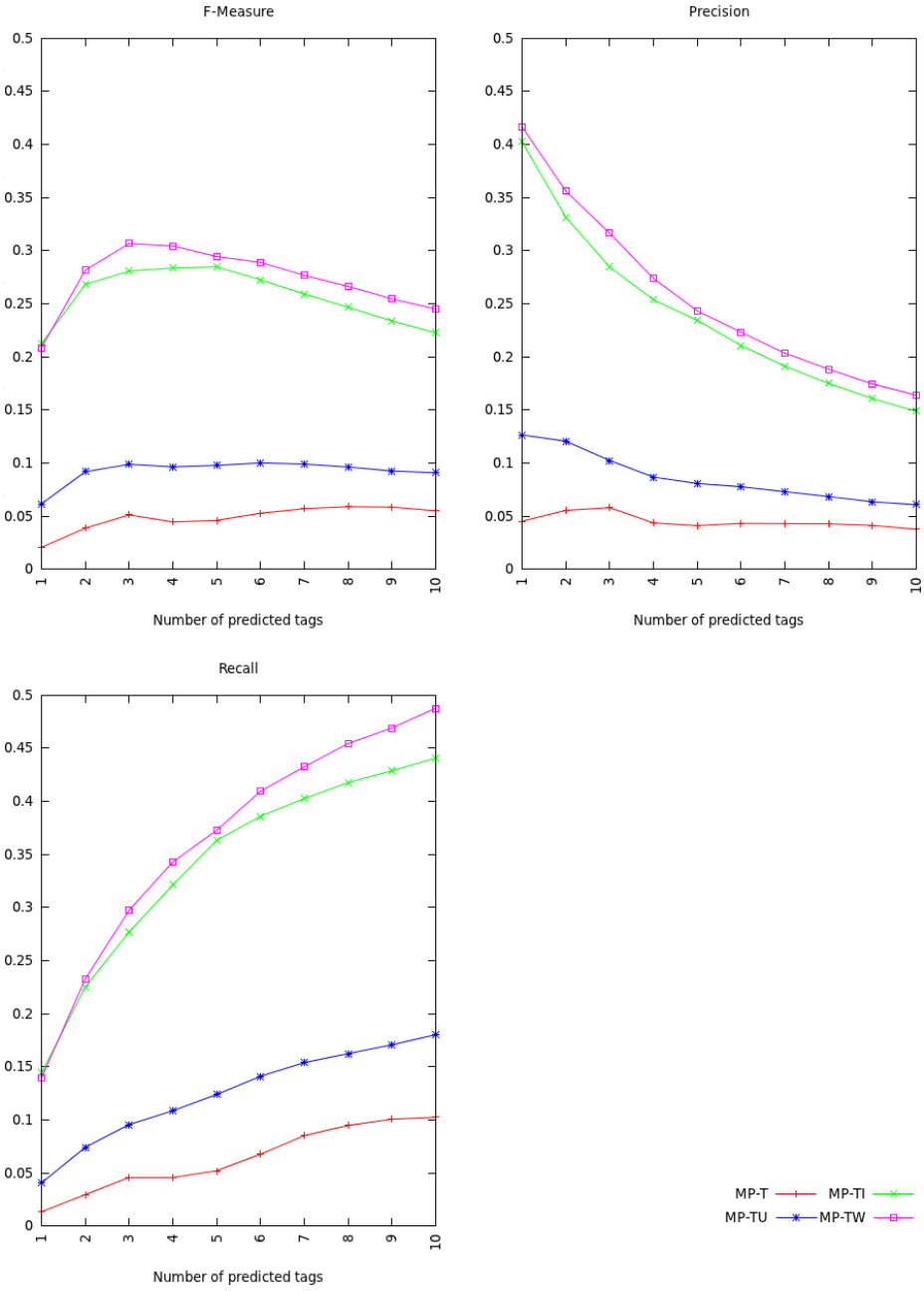

Fig. 3. Comparison of the baselines MP-T, MP-TI, MP-TU and MP-TW on post-core dataset of ECML PKDD 2009 Discovery Challenge [3].

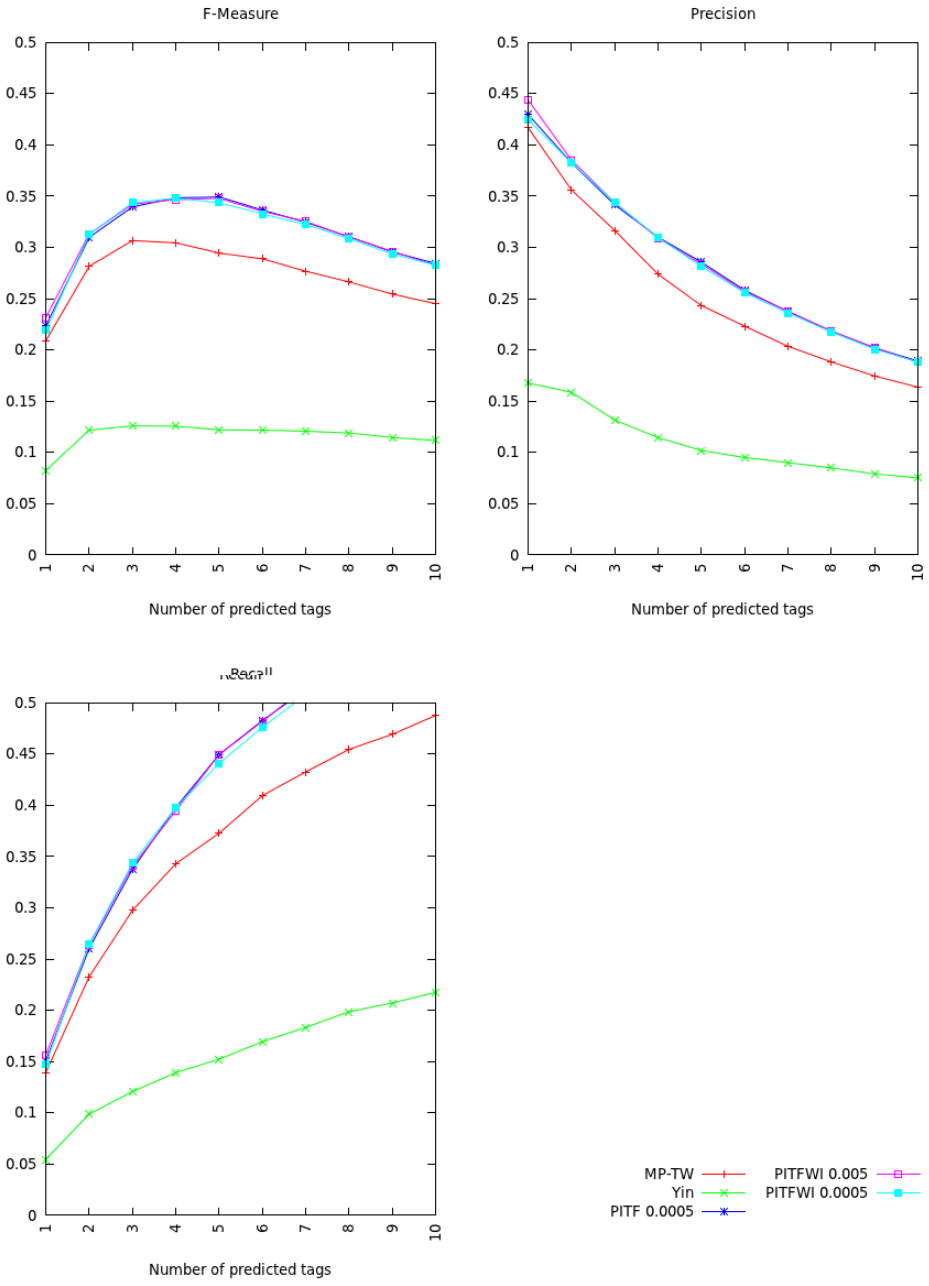

Fig. 4. Evaluation of the presented PITF-WI model in comparison to other approaches on post-core dataset of ECML PKDD 2009 Discovery Challenge [3].

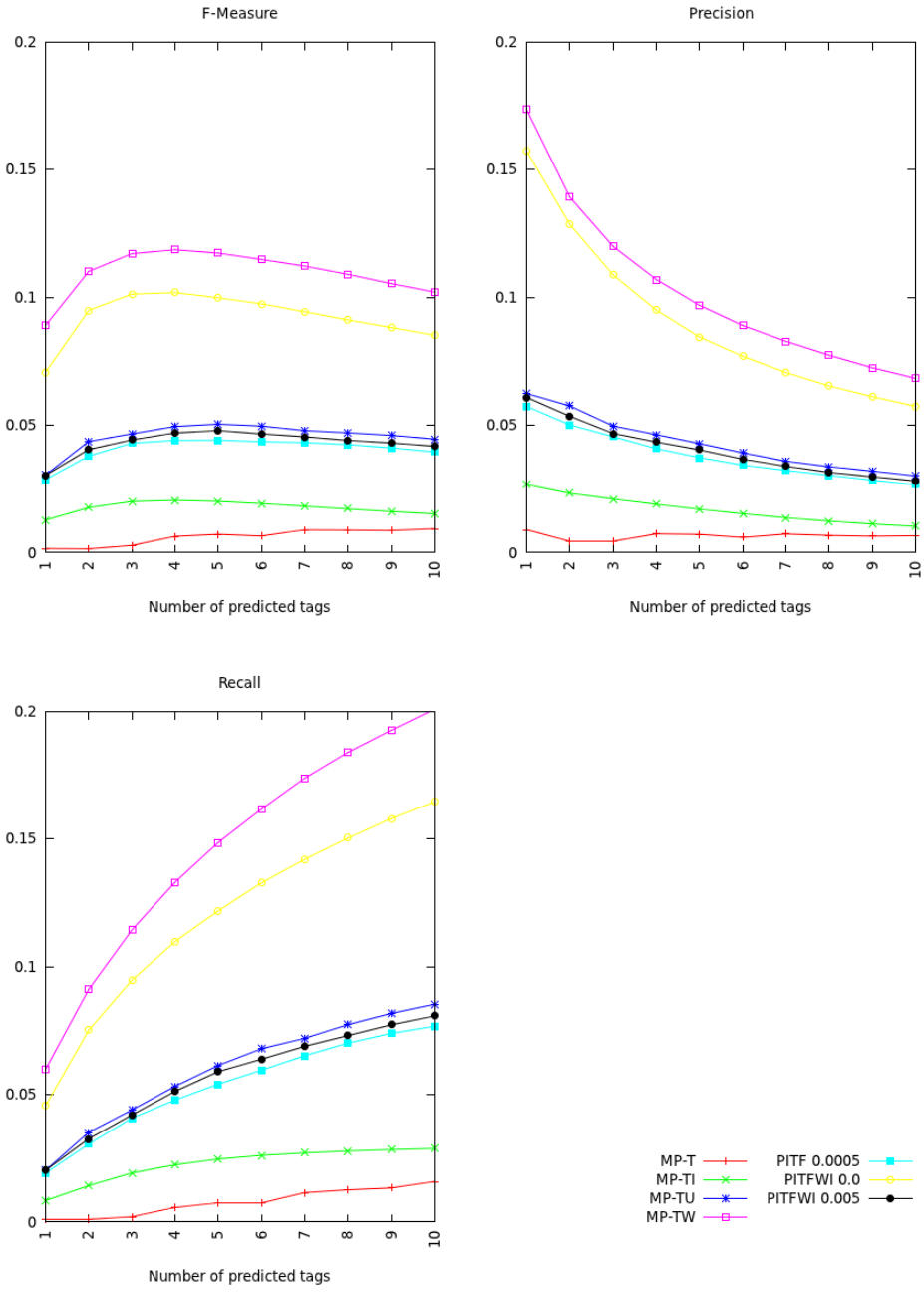

Fig. 5. Evaluation of presented models in comparison to other approaches on cleaned dump dataset of ECML PKDD 2009 Discovery Challenge [3].

References

1. Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 79–86, New York, NY, USA, 2010. ACM.
2. Caimei Lu, Xiaohua Hu, Jung-ran Park, and Jia Huang. Post-based collaborative filtering for personalized tag recommendation. In *Proceedings of the 2011 iConference*, iConference '11, pages 561–568, New York, NY, USA, 2011. ACM.
3. European Conference on Machine Learning, Priciples, and Practice of Knowledge Discovery in Databases. Datasets of discovery challenge 2009. http://www.kde.cs.uni-kassel.de/ws/dc09/dataset, April 2009.
4. Steffen Rendle. Factorization machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 995–1000, Washington, DC, USA, 2010. IEEE Computer Society.
5. Steffen Rendle. *Context-Aware Ranking with Factorization Models*, volume 330 of *Studies in Computational Intelligence*. Springer, 2011.
6. Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 727–736, New York, NY, USA, 2009. ACM.
7. Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 81–90, New York, NY, USA, 2010. ACM.
8. Dawei Yin, Zhenzhen Xue, Liangjie Hong, and Brian D. Davison. A probabilistic model for personalized tag prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 959–968, New York, NY, USA, 2010. ACM.