Analysis of PITF and BPR-OPT

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Richard J Park

Spring, 2020.

Technical Project Team Members

Hongning Wang

Madhav Marathe

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Hongning Wang, Department of Computer Science

Analysis of PITF and BPR-OPT

Analysis of the runtime complexity speedup and importance of optimization in Recommender Systems

Richard Park
University of Virginia
rp5zp@virginia.edu

ABSTRACT

The work of Pairwise Interaction Tensor Factorization (PITF) for personalized tag recommendation has won the ECML/PKDD Discovery challenge in 2009 for graph-based tag recommendation, Rendle et al. This capstone project aims to analyze the work of the PITF model, specifically, the learning criterion of Bayesian Personalized Ranking Optimization (BPR-OPT) and the tradeoff between runtime and prediction quality from the Tucker Decomposition (TD) model for personalized tag recommendations. In doing so, I will also be analyzing the importance of the learning criterion and baseline testing to show the improvements in the factorization dimension from a cubic runtime to a proposed linear runtime in both prediction and learning (PITF-BPR). By analyzing the runtime of a factorization model to achieve better prediction quality, I will be combing the works done through both courses, Algorithms and Information Retrieval, to provide insights about the work and the importance for optimization in recent neural recommender approaches.

1 Introduction

There is an abundance of information on the internet available to browse and search. The goal of a tag recommendation is to offer a list of tags for users to annotate an item, allowing easier access to browsing and searching. The task of tag recommendations has been tackled with solutions like non-personalized tag recommendations and personalized tag recommendations, however, non-personalized methods have been significantly outperformed by recent personalized tag recommenders [2]. The main difference between personalized and non-personalized recommendations is the utilization of a user's past tagging data in order to make a recommendation. Personalization of tag recommenders as a concept seems it could outperform any non-personalized method, but in reality, there is a need for a larger amount of user behavior data in order to create these personalized recommender systems, and there comes a difficulty with measuring baselines for these methods.

Since the Netflix Prize competition, matrix factorization has been the state-of-the-art model for collaborative filtering. Progress in this field in recent years has shown the potential for neural collaborative filtering, but studies have shown that with proper optimization, matrix factorization methods still outperform most of these state-of-the-art neural models regarding item recommendation problems. Throughout this analysis, I will be analyzing the PITF model for tag recommendations, but will also draw a connection to item recommendations tasks since learning algorithms and optimization are generalized for these tasks.

There are still some non-personalized methods that have shown to outperform other methods using proper hyperparameter optimization and given appropriate datasets. In Dacrema et al., the authors call for better baseline method testing and proper hyperparameter tuning practices by showing how naïve and simple implementations can outperform state-of-the-art collaborative filtering methods including Neural Collaborative Filtering [3]. In the information retrieval community, there have been many calls for stronger baseline testing in order to measure the progress of the research in this field. Rendle et al. highlights the difficulty in measuring baselines and the need for standardized baselines by showing how simple matrix factorization methods can outperform many newly proposed researches in 2019. Through the analysis of the PITF model that is learned with BPR-OPT, I will analyze the progress made in tag recommendations to show that results in prediction quality not only depend on the model, but heavily rely on the optimization and learning functions in order to make quality recommendations.

There are 3 main contributions from the work of PITF model that I will analyze and summarize:

1. The use of Bayesian Personalized Ranking optimization criterion for tag recommendation, using a learning algorithm based on stochastic gradient descent with bootstrap sampling. BPR-OPT

was used to optimize the baseline algorithms of the TD model as well.

2. The runtime of the PITF model is linear in prediction runtime, I will also show the relationship to the generic TD model and Canonical Decomposition model (CD) as proposed by Rendle et al.
3. BPR-PITF model outperforms RTF-TD in runtime by dropping from cubic $O(k^3)$ to linear $O(k)$, where k is the factorization dimension. On top of runtime, the BPR-PIFT outperforms the baseline RTF-TD method on larger datasets, and performs very similarly on the Bibsonomy dataset.

2 Related Works

2.1 Non-Personalized Tag Recommender

As mentioned in the introduction, non-personalized solutions have been used in recommendation problems. For item recommendations, they have shown to be useful in some cases, but in the scenario for tag recommendations, it has been empirically shown that personalized methods like Folkrank and RTF outperform the theoretical upper bounds of any non-personalized methods.

2.2 Personalized Tag Recommender

Personalized recommendations have been the main focus for tag recommendations as the adoption of FolkRank produced high quality results that outperformed many previous models and collaborative filtering models. FolkRank is a variation of the original PageRank ranking algorithm.

Factorization methods were introduced to tag recommendations beginning with the Tucker Decomposition model. Rendle et al. introduces an optimized learning approach for TD models that uses the area under the ROC-curve as a ranking statistic to optimize model parameters. This approach will later be mentioned as the RTF (Ranking with Tensor Factorization) optimization criterion for tag recommendations. Similarly, to this AUC model, the optimization of BPR for tag recommendations also optimizes over pairs of ranking constraints. Although, in contrast to this method, BPR optimizes for pair classification.

2.3 Tensor Factorization Models

Factorization models are widely used in the field of recommender systems, most notably beginning with the Tucker decomposition model on which tag recommenders like tensor dimensionality reduction were based on. These models are based on a Higher-Order-Singular-Value-Decomposition, (HOSVD), which corresponds to a TD model that optimizes for square-loss where values not observed are learned as 0s. The HOSVD method has been shown to be outperformed by other optimization criteria to achieve better recommendations. For the proposed PIFT model, we will also look at the Canonical Decomposition (CD) that is considered a special case of TD using parallel factor analysis, and will be used to assess the runtime speed up of PIFT.

2.4 Pairwise Interaction Model

The PIFT model was able to score first place in the ECML PKDD Discovery challenge in 2009. The purpose of this paper is to analyze the approaches for optimization used in this model, and apply it to present day research in this field. Differently from the paper written in the challenge, this paper will show the relation to TD, RTF, HOSVD, as well as the CD model. The authors specifically show how their approach compares to state-of-the-art methods on other datasets

2.5 BPR-OPT

The importance of an optimization method is highlighted by Rendle et al. [2]. BPR-OPT is a general optimization criterion for personalized ranking, using the maximum a posteriori estimation based on a Bayesian analysis of the ranking problem. The authors show that BPR optimized methods outperform other methods like SVD Matrix Factorization (SVD-MF) and Weighted-Regularized Matrix Factorization (WR-MF). The focus on BPR optimization for matrix factorization methods shows that even though methods may share the same exact model (SVD-MF, WR-MF, BPR-MF), the optimization technique yields much differing prediction quality. The results from [2] indicate that BPR optimization is the appropriate choice for the task of personalized ranking.

3 System Design

3.1 Personalized Tag Recommendation

Personalized tag recommendations suggest relevant list of tags for users to annotate web resources with. Annotating an item consist of a tag that describes the item, for example, a music website in which users are suggested tags to annotate a song that describes the song using keywords. As mentioned in the introduction, personalized tags are useful for these problems since the past historical data of the system is very useful for future recommendations. Items that are tagged similarly in the past can be used by recommender systems to suggest future tags for similar items.

The formalization for personalized tag recommendation uses the notation as follows: *U* is the set of all users, *I* is the set of all items and *T* is the set of all tags. The past historical data of tagging events is $S \subseteq U \times I \times T$. The set of all users, items, and tags make up the triples in *S*, representing the positive tagging data from the past. This ternary relation can be represented as a three-dimensional tensor that shows only the positive observations and no explicit observations of negative tagging events.

For tag recommendation problems, we are interested in looking at the recommended list of tags for a given user-item pair (*u,i*). The (*u,i*) pair is known as a post. Recommendations for posts are formulated as a ranking problem, therefore it is formulated as predicting a total order over tags that must satisfy totality, anti-symmetry, and transitivity [1].

All models in the paper predict a scoring function that is used to satisfy the ranking conditions of totality, anti-symmetry, and transitivity. Identical scores from two unique tags and the same user-item combination results in random ranking of one tag before the other in order to ensure totality.

3.2 Data Analysis

The data collected from past tagging events, *S*, only observes positive events. This become a major problem in data analysis in current machine learning approaches since this would mean that all negative events (tags that a user does not like) are not recorded. Many approaches solve this issue by taking the set of all tags not in *S* to be considered as negative events.

There are drawbacks to this method listed in other papers, therefore the approach used by the PIFT model proposes to infer pairwise ranking constraints, D_S, from S. The training data D_S for pairwise constraints is defined as:

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

Within a post (*u,i*), a tag (Tag A) is proposed to be preferred over another tag (Tag B) iff Tag A has been observed and Tag B has not been observed. The advantage of this approach is that rankings that should be predicted in the future are treated as missing values in which rankings cannot be inferred.

3.3 BPR for Tag Recommendation

The Bayesian Personalized Ranking was first introduced in the problem setting of item recommendation. The analysis of this section is a derivation of this BPR-OPT and LearnBPR algorithm, that are used to optimize the factorization models. These models are generic and are not restricted to factorization models like PITF.

The optimization criterion for finding the best ranking for a given post is formalized in the original paper. The goal is to maximize the probability of the model parameters. If we treat each event of posts as independent events, then this assumption leads to the maximum a posteriori estimator for the model parameters [1]. This probability can be estimated from the observed data, assuming pairwise independence.

Next, the estimator is derived by plugging in the scoring function model mentioned in section 3.1 for the tags. The authors also assume that model parameters are drawn from a Normal distribution across theta, centered at 0, with the logistic function being the model specific variance vector. By filling this into the MAP estimator, we get the BPR-OPT criterion:

$$\begin{aligned} \text{BPR-OPT} &:= \ln \prod_{(u,i,t_A,t_B) \in D_S} \sigma(\hat{y}_{u,i,t_A,t_B})\, p(\Theta) \\ &= \sum_{(u,i,t_A,t_B) \in D_S} \ln \sigma(\hat{y}_{u,i,t_A,t_B}) - \lambda_\Theta ||\Theta||_F^2 \end{aligned}$$

The BPR learning algorithm is derived by optimizing the BPR-OPT model parameters. The authors state that a computation of normal gradient descent is not feasible for this model since the dataset for the ECML/PKDD challenge consisted of 3,299,006,344 quadruples in the evaluation section. Therefore, optimizing BPR-OPT would be time consuming and inefficient using a full gradient descent approach. Since this is the case, the BPR algorithm instead draws random quadruples from the dataset. The observation that multiple quadruples overlap in their three dimensions is what motivated this idea, since it would be beneficial for many other related cases to sample a random case and performing stochastic gradient descent. The formalization for gradients for optimization is listed in the original paper. In summary, BPR optimizes the tag recommender model with bootstrapping based stochastic gradient descent. The learning rate and regularization constants are also listed.

4 Factorization Models

As mentioned, Factorization models became an incredibly successful model for recommender systems beginning with the performance at the Netflix Challenge. In the task of item prediction, factorization models have been empirically shown to outperform baseline methods like KNN collaborative filtering. For tag recommendations, these models generate high quality predictions that outperform Folkrank. The main difference in two-dimensional matrix factorization and in tag recommenders is the many different ways of factorizing the data. The focus of this paper is on 3 models of factorization methods in tag

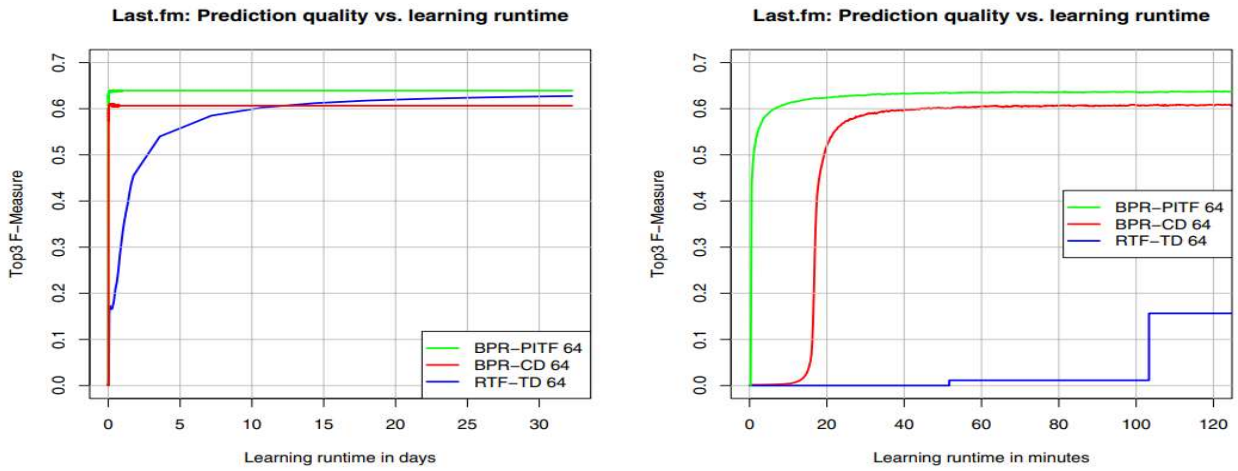

Figure 1. The top 3 list F-Measure score vs learning runtime in days/ minutes on the Last.fm dataset. Last.fm is a relatively larger dataset .

recommendations: TD, CD, and PITF. Each model is learned with BPR and uses the same scoring function to rank the posts, which allows tags to be sorted with respect to their score.

4.1 Tucker Decomposition (TD)

Tucker Decomposition is a factorization model that factorizes a high-order cube into one core tensor and a factor matrix for each dimension. The model parameters and TD model with BPR-OPT formalization is shown in [1]. The limitation of this model is the cubic runtime complexity for predicting one triple. There is a nested sum of degree 3, therefore, we can say that the runtime complexity for predicting a triple (*u, i, t)* is $O(k^3)$ (table 2).

4.2 Canonical Decomposition (CD)

The CD model is a case derived from the TD model that is achieved by setting the core tensor equal to the diagonal tensor. The gradients for this model are also listed in the original paper. By replacing the core tensor of the TD model, the CD model sees a runtime speed up from cubic to linear, O(k), as the model equation (table 2) no longer contains nested sums.

4.3 Pairwise Interaction Tensor Factorization (PITF)

The approach of PITF models the interaction between users, tags and items by factorizing the two-way interaction between these sets. PITF looks at the interactions of user-tag, item-tag, and user-item. The interaction of user-item can be cancelled out since both the BPR-OPT criterion and ranking ignores the score of any user-item interaction. This results in the final model equation that is a summation of two dimensionalities, user-tag and item-tag (table 2). Most notably, the runtime for predicting a triple is O(k), therefore, since the PITF model is a special case of the mentioned CD approach, this has a final runtime complexity of O(2k).

4.4 TD, CD, and PITF Runtime Analysis

The proposed PITF model and its relation to the other factorization models is described (table 2). CD is a derivation of TD, and PITF is a derivation of CD. At first, it seems as if reducing the expressiveness of the factorization model worsens the prediction quality as a tradeoff for improved runtime. This is the case since the replacement of the core tensor in CD to a diagonal tensor treats some unused features as 0. In figure 1, the CD method is significantly faster to achieve a higher prediction accuracy, but over time, TD catches up to outperform this model. Thus, with the improvements in runtime, there is a simultaneous need to improve prediction quality using pairwise interactions. The following section will discuss the tradeoff and investigate the runtimes and prediction quality for each model.

5 Results

There are 3 datasets used for evaluation, the Bibsonomy, Last.fm, and the ECML/PKDD dataset 2009. The datasets have a property of p-core. The p-core of a dataset is the largest subset such that every user, item, and tag must occur in at least p posts. The Bibsonomy data is 5-core, Last.fm is 10-core, and the ECML dataset is 2-core.

dataset	Users $\|U\|$	Items $\|I\|$	Tags $\|T\|$	Triples $\|S\|$	Posts $\|P_S\|$
BibSonomy	116	361	412	10,148	2,522
Last.fm	2,917	1,853	2,045	219,702	75,565
ECML/PKDD Discovery Challenge 09	1,185	22,389	13,276	248,494	63,628

Table 1. Characteristics of each dataset. Last.fm contains the most posts (user- item pairs) and is also 10-core.

5.1 Evaluation Methods

In evaluating the bibsonomy and last.fm dataset, one post per user is randomly removed from the training set and replaced in the test set. The reason for this random removal was motivated by the evaluation that some of the data contained users that only had 2 posts.

Once the train-test split has been made, each recommender model is trained on the test set and prediction quality is measured based on the testing set. The evaluation metrics of F-measure in Top-N lists is used. Based on the test set; Precision, Recall, and F-measure are recorded. Experiments are repeated 10 times, each iteration sampling a new train/test set split. The average of all runs is taken and the f-measure is reported as the f-measure over the average recall and average precision.

The hyperparameters for the models are searched with respect to the first training split. RTF-TD, BPR-PITF, and BPR-CD are all recorded with their respective runtime in learning with a C++ implementation. The runtime experiments were carried out using a compute cluster of 200 cores in total. To ensure consistency, each compute node has identical hardware and software, as well as the usage of 1 core per run (no parallelization within the compute nodes). Since previous experiments had already measured the proposed methods with non-personalized tag recommenders, outperforming all theoretical upper bounds, the comparisons will be made with HOSVD, FolkRank, and Adapted Pagerank.

5.2 Learning Runtime

The convergence of BPR-PIFT, BPR-CD, and RTF-TD are presented for a given time span of 30 days (figure 1). The results from experimentation takes the top-3 list f-measure score for each model and graphs it along the learning runtime in days to observe the relationship between prediction quality and learning runtime. For a model of k=64, RTF-TD needed about 12 days to fully converge and achieve prediction quality comparable to BPR-CD. RTF-TD is still unable to perform as well as BPR-PITF even after 30 days.

BPR-PITF and BPR-CD converge significantly quicker than the TD model as expected. The interesting part is how quickly the BPR optimized models converge. Another graph displayed showing the convergence of these models over a 2-hour time period. BPR-PIFT converges after 20 minutes, while BPR-CD takes about 40 min to converge on this dataset. Another interesting observation from this experiment is the beginning of the BPR-CD model, in which Rendle et al. states that it is possible to see the need for updates in the first couple minutes of iteration in CD because of the three-way interaction searching within the CD structure. This is different than the proposed two-way interaction of PITF since the third interaction is already given by the two pairwise interactions.

Furthermore, another notable analysis is that the compute nodes were not parallelized during this process, but this could change the results for the learning of both BPR-CD and BPR-PITF since both of these models can be easily parallelized due the nature of the quadruples that are randomly drawn usually don't have shared parameters. RTF-TD on the other hand cannot be parallelized since all entries share the core tensor.

5.2 Prediction Quality

After evaluating the learning runtime and observing the expected results, we can now analyze the prediction quality and observe the tradeoff between runtime and prediction. BPR-PITF is compared to many other competing methods. To summarize the results, it shows as expected, that factorization models result in the highest prediction quality. The only exception to this is the FolkRank algorithm that performed competitively on small datasets and small list sizes (figure 2).

BPR-PITF is able to outperform BPR-CD. This comparison is important since these models both have a linear runtime with respect to k. BPR-CD is more generalized than PITF, but through observations of the experiment, it seems that the CD model is unable to identify the pairwise structure of PITF in order to achieve regularization at the same time.

When comparing the BPR-PITF method to RTF-TD, the prediction quality of the TD model is actually able to outperform BPR-PITF in small datasets like the Bibsonomy set. When using larger datasets, BPR-PITF outperforms RTF-TD (figure 2) on all list sizes. With these results, we can conclude that BPR-PITF has a higher prediction quality

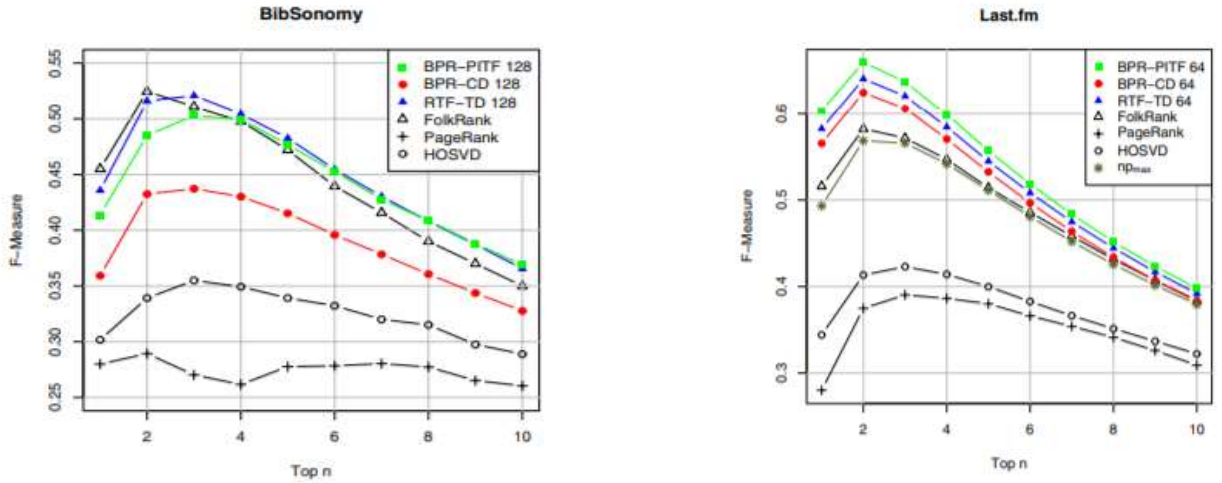

Figure 2. Results of F-measure in top-n lists for BibSonomy dataset and Last.fm. Experiment used 128 and 64 factorization dimensions, respectively.

in larger datasets that does not come at the cost of learning runtime. The learning speedup does not affect the prediction quality, therefore for larger datasets, it is expected to have BPR-PIFT outperform RTF-TD largely in runtime and comparably in prediction quality.

6 Conclusion and Analysis

The proposed factorization model for tag recommendations models the pairwise interactions between the set of users, items, and tags. The analysis of runtime speedup and the relationship to the TD and CD model shows the improvement from a cubic runtime ($O(k^3)$, where k is the factorization dimensions of the feature matrices) to a linear runtime O(k). This difference was also shown by observing the convergence of learning rate and prediction quality graphs (figure 1). This was a significant speed up compared to the cubic TD model that did not come at the cost of prediction quality. Unlike the speedup proposed by the CD model, we empirically saw the prediction quality outperform the TD model. Even though TD is able to outperform in small datasets, the cost for learning is not worth the lesser prediction quality in large scale. Especially with the vast amount of data available, there is a need for a runtime speed up at large scale that does not affect prediction quality. The proposed model also furthered the research of BPR-OPT to show the importance of optimization criterion, as well as the usage of the BPRLearn algorithm. Through this capstone research, I was able to combine the works done in courses of Algorithm and Information Retrieval to analyze the work of PIFT, BPR-OPT, and analyze the prediction runtime/prediction quality tradeoff for the proposed method.

The connections from item recommendation system to tag recommendation systems is present throughout this paper, therefore I further this analysis by drawing connections to a recent 2019 paper on the progress of recent neural approaches to top-n recommendation problems.

6.1 Progress in Recent Works Analysis

Rendle et al. touches on the difficult of evaluating baselines for recommender systems in a separate paper. Similarly, Dacrema et al. [3] makes a call for better research practices in recommender systems to improve the "state-of-the-art" proposed research methods. This is interesting to note since this PIFT method was a proposed state-of-the-art method for factorization models. Dacrema et al. ran experiments to justify his claim and found that most reproducible work was able to be outperformed by naïve solutions with proper fine tuning, even some non-personalized solutions that Rendle et al. empirically showed were inferior. For the case of tag recommendations, Rendle emphasizes the need for proper optimization, which is the reason for the usage and focus on BPR-OPT and the BPR learning algorithm. As a conclusion, there exists an inevitable difficulty in measuring baselines for recommender systems since most comparisons are made on empirical data and experimentation with accuracy metrics. The need for a standardized community baseline of metrics is required to improve progress in this field and highlight the importance of optimization.

7 Future Work

As part of future work, I would like to draw more connections to older works in factorization models to newer works in neural models in order to show the importance of optimization. This can include reimplementation to verify the optimizations and produce my own experiments to calculate error within these algorithms. Dacrema et al. mentions in their future works about using matrix factorization baselines to address the problems of progress in neural approaches. As research grows, it is important to continuously move baselines at the same speed as state-of-the-art methods, since any method can look good and perform well compared to a naïve, unoptimized approach.

Factorization Model	**Prediction Runtime in Big O-notation for predicting triple (*u,i,t*)**	**Prediction Quality Notes**	**Other Notes**
RTF-TD	$O(k^3)$, where k is factorization dimensions. Model Equation: $\hat{y}^{\text{TD}}_{u,i,t} := \sum_{\tilde{u}} \sum_{\tilde{i}} \sum_{\tilde{t}} \hat{c}_{\tilde{u},\tilde{i},\tilde{t}} \cdot \hat{u}_{u,\tilde{u}} \cdot \hat{i}_{i,\tilde{i}} \cdot \hat{t}_{t,\tilde{t}}$ Cubic, therefore slow learning even for small dimensions	Outperforms (including BPR-PIFT) all others in **small datasets** (BibSonomy) and small top-n (1,2,3) list sizes.	Converges slowly, but able to achieve higher prediction quality than BPR-CD.
BPR-CD	O(k) Model Equation: $\hat{y}^{\text{CD}}_{u,i,t} := \sum_{f}^{k} \hat{u}_{u,f} \cdot \hat{i}_{i,f} \cdot \hat{t}_{t,f}$ Linear, much better runtime complexity since no nested sums.	Prediction quality converges significantly faster than RTF-TD (figure 1)	Converges quickly, but a faster prediction runtime is traded for a lower prediction quality. No nested sums like TD model, but if dimensionality of the feature matrices differ, then some features end up being not used and set to 0.
BPR-PITF	O(k) Model Equation: $\hat{y}_{u,i,t} = \sum_{f} \hat{u}_{u,f} \cdot \hat{t}^{U}_{t,f} + \sum_{f} \hat{i}_{i,f} \cdot \hat{t}^{I}_{t,f}$ Linear, models two-way interactions between users, tags, and items. Learns faster than both TD and CD models	Outperforms RTF-TD on **larger datasets**, on all list sizes.	Shows prediction quality does not come at cost with runtime improvements

Table 2. Summary table for Runtime/Prediction quality on Factorization Models

REFERENCES

[1] Rendle, S., & Schmidt-Thieme, L. (2010). Pairwise interaction tensor factorization for personalized tag recommendation. Proceedings of the Third ACM International Conference on Web Search and Data Mining - WSDM '10. the third ACM international conference. https://doi.org/10.1145/1718487.1718498

[2] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009), 2009

[3] Dacrema, M. F., Cremonesi, P., & Jannach, D. (2019, September 10). Are we really making much progress? A worrying analysis of recent neural recommendation approaches. Proceedings of the 13th ACM Conference on Recommender Systems. RecSys '19: Thirteenth ACM Conference on Recommender Systems. https://doi.org/10.1145/3298689.3347058