# Proposal of Course Project

Erheng Zhong and Si Shen

March 22, 2011

## 1 Introduction

People are always fascinated by beauty of music during the long history of humanities. Various music genres and styles have emerged, reflecting the diversity of personalities in numerous cultures and age groups. It is hence not surprising that human tastes in music are remarkably diverse. Yahoo! Music has collected billions of user ratings for music pieces, which will reveal hidden information if properly analyzed. Possible knowledge includes how songs are grouped, which hidden patterns link various albums, which artists complement each other, and above all, which songs users would like to listen to. Based on these motivations, KDD-CUP 2011 has set up two challenges: 1. predicting the scores that users would give to a particular item; 2. distinguishing beloved songs from others for each user.

To tackle the problems, we plan to explore methods stemming from *Matrix Factorization* (MF). One of the advantages of these methods is that auxiliary information of both users and items can be incorporated into the prediction model. Furthermore, having its origin in stochastic optimization algorithm, the approach can be operated efficiently. In general, we propose to achieve this goal in five steps:

1. To compute some statistics of the dataset to gain some insights (03/15/2011∼03/31/2011)
2. To apply basic MF methods to the rating matrix to obtain baseline results (04/01/2011∼04/07/2011)
3. To explore collaborative MF methods to utilize meta information of songs (04/08/2011∼04/22/2011)
4. To build ensemble machine using boosting and model average (04/23/2011∼05/07/2011)
5. To include temporal information by using *Tensor Factorization* (05/07/2011∼05/14/2011l)

## 2 Plan and Methodology

We briefly state some key points of our plan in this section. We first describe the basic models used in Track1 and Track2 and then propose some extensions:

ensemble, temporal or dynamic prediction, etc.

## 2.1 Track 1

We apply MF model to solve problem of link prediction. Let $R$ denote the rating matrix, and $Y$ the matrix which pre-sets the meta information of items. We aim to factorize these two matrices into three latent matrices: $U$, $V$ and $W$:

$$\min_{U,V,W} \ell(R,U,V) + \lambda\ell(Y,V,W) + \alpha\mathcal{R}(U,V,W) \tag{1}$$

where $\ell$ is the approximation loss, $\mathcal{R}$ is the regularization terms for the latent matrices and $\lambda$ and $\alpha$ are the regularization parameters. Different loss functions and regularization terms can be explored and studied. To be more precise, we have different strategies on each terms in Eq.(1). For $\ell(R,U,V)$, we will consider the implicit feedbacks from users and the implicit relationship between users in addition to the explicit ratings. For $\ell(Y,V,W)$, weights will be introduced to different features for the meta information is diversified. For the restriction part $\mathcal{R}(U,V,W)$, different items would receive varying regularization terms due to the hierarchical structure, which also serves as the motivation for us to decompose the matrices locally under different contexts.

## 2.2 Track 2

This track bears two properties: implicit feedback (beloved vs. be-hated or -1 vs. 1) and top-K recommendation. We model the former as a collaborative classification problem while proposing a collaborative ranking model for the latter. One natural scheme is to follow the same framework of track 1 but with different terms. For example, we can define $\ell(R,U,V)$ as

$$\ell(R,U,V) = \sum_{r_{ij}\in R} |1 - r_{ij}(U_i V_j^T)| \tag{2}$$

where $r_{ij} \in \{-1, 1\}$. After decomposition, we can also predict user preference on each item and pick the top-K as the results. Another example is to model the preference difference between two items directly:

$$\ell(R,U,V) = \sum_{r_{ij}\in R} \sum_{r_{ik}\in R} |1 - (r_{ij} - r_{ik})(U_i V_j^T - U_i V_k^T)| \tag{3}$$

## 2.3 Ensemble Method

We plan to build ensemble machines in the following two ways. 1. Linearly combine models that belong to the same category using boosting or model average. For example, we can apply AdaBoost-like techniques to build a series of basic MF models and combine them. 2. Build different methods first and then combine their prediction results. Let $F = \{f^1, \cdots, f^n\}$ denote the built models. We use them as *features*, the target ratings as true labels and hence a new dataset is generated. For instance, for a rating $r_{ij}$, we can build a instance as

$$\{f_{ij}^1, \cdots, f_{ij}^n; r_{ij}\} \tag{4}$$

where $f_{ij}^k$ is the $k$th model's prediction on the rating $r_{ij}$. After that, this dataset can be used to build a classifier and then generate a sophisticated ensemble result.

### 2.4 Temporal

We can add one more dimension on the rating matrix to generate a 3D tensor, which includes the temporal information. Thus, one instance is represented as $< u_i, v_j, t_k, r_{ijk} >$ meaning that user $u_i$ rated item $v_j$ as $r_{ijk}$ at timestamp $t_k$. Consequently, we explore different tensor factorization methods to decompose this new tensor and obtain predictions.

## 3 Background

### 3.1 Competition Description

The contest releases over 300 million ratings by over 1 million anonymized users. The ratings are given to different types of items - songs, albums, artists, and genres - all tied together within a known taxonomy. There are two tracks: Track1 aims at predicting scores that users would give to different items and Track2 requires separation of beloved songs from others.

As for the dataset, there are three types of items: songs, artists and albums, among which songs and albums are annotated with genres. It has four specific features:

- Scale: biggest public dataset ever. 1 million user, 0.6 million items, 300 million ratings
- Hierarchical item structure: songs belong to albums, albums belong to artists. All of them are annotated with genre tags
- Rich meta data: over 900 genres
- Fine temporal resolution: no previous challenge provided time in addition to date

### 3.2 Related Work

The first task is a link prediction problem while the second one can be considered as a ranking problem. We discuss some related works briefly here. For the former problem, [3, 4] introduce a series of matrix factorization methods to cope with Netflix challenge[1]. The authors introduce a basic SVD technique plus regularization norms before combining the implicit feedbacks from users and *Neighborhood* model to generate a more effective algorithm. To overcome

[1] `www.netflixprize.com/`

the drawback of point estimation, Bayesian method is applied. [8] explores Bayesian method to avoid over-fitting and uses MCMC to estimate the parameters. To take the advantage of kernel method, [5] proposes a non-linear matrix factorization based on Gaussian process latent variable models (GP-LVM). To generate rating values under different contexts, $M^3F$ [6] introduces context dependence rating prediction by allowing each item to select a new topic for each new interaction. The second task is similar to two previous contests: KDD-CUP 2007[2] and CAMRA 2010[3]. Its target is to rank items instead of predicting their scores, namely *Top-K Recommendation*. It can be considered as a collaborative classification problem: beloved vs be-hated. [2] identifies some unique properties of implicit feedback datasets and proposes a factor model which is especially tailored for implicit feedback recommenders. [7] presents a generic optimization criterion for personalized ranking: BPR-Opt, which is the maximum posterior estimator derived from a Bayesian analysis of the problem. [1] analyzes the performance of top-K recommendation system and therefore offers the empirical guidance.

# References


[1] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 39–46, New York, NY, USA, 2010. ACM.

[2] 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*, pages 263–272, Washington, DC, USA, 2008. IEEE Computer Society.

[3] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.

[4] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42:30–37, August 2009.

[5] N. D. Lawrence and R. Urtasun. Non-linear matrix factorization with gaussian processes. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 601–608, New York, NY, USA, 2009. ACM.

[6] L. W. Mackey, D. Weiss, and M. I. Jordan. Mixed membership matrix factorization. In *Proceedings of the 27th International Conference on Machine Learning*, ICML '10, pages 711–718, Haifa, Israel, 2010. Omni Press.

[7] S. Rendle, C. Freudenthaler, Z. Gantner, and S.-T. Lars. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.

[8] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 880–887, New York, NY, USA, 2008. ACM.


[2] `www.sigkdd.org/kdd2007/kddcup.html`

[3] `http://www.wikicfp.com/cfp/servlet/event.showcfp?eventid=9952`