The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Confidence-Aware Matrix Factorization for Recommender Systems

Chao Wang,[†] Qi Liu,[†] Runze Wu,[†] Enhong Chen,[†*]
Chuanren Liu,[‡] Xunpeng Huang,[†] Zhenya Huang[†]
[†]Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China,
{wdyx2012, wrz179, hxpsola, huangzhy}@mail.ustc.edu.cn, {qiliuql, cheneh}@ustc.edu.cn
[‡]Decision Science and Management Information Systems Department, Drexel University, chuanren.liu@drexel.edu

Abstract

Collaborative filtering (CF), particularly matrix factorization (MF) based methods, have been widely used in recommender systems. The literature has reported that matrix factorization methods often produce superior accuracy of rating prediction in recommender systems. However, existing matrix factorization methods rarely consider confidence of the rating prediction and thus cannot support advanced recommendation tasks. In this paper, we propose a Confidence-aware Matrix Factorization (CMF) framework to simultaneously optimize the accuracy of rating prediction and measure the prediction confidence in the model. Specifically, we introduce variance parameters for both users and items in the matrix factorization process. Then, prediction interval can be computed to measure confidence for each predicted rating. These confidence quantities can be used to enhance the quality of recommendation results based on Confidence-aware Ranking (CR). We also develop two effective implementations of our framework to compute the confidence-aware matrix factorization for large-scale data. Finally, extensive experiments on three real-world datasets demonstrate the effectiveness of our framework from multiple perspectives.

1 Introduction

As one of the most successful data mining tasks, recommender systems have been prevailing for decades (Lu et al. 2015; Zhao et al. 2016; Liu et al. 2011). Massive research efforts have been made into two general approaches: content-based models and collaborative filtering (CF) methods. Due to the superior performance for rating prediction that can support the recommendation tasks, collaborative filtering(CF) methods, especially matrix factorization (MF) based CF, have been widely applied in web-based services like Google, Netflix and Amazon (Dror et al. 2012).

By projecting users and items into latent factor space, most of MF methods pay more attention to the accuracy of rating prediction (Koren, Bell, and Volinsky 2009). However, these existing accuracy-oriented MF methods cannot always meet the expectation of end-users. For example, users do not necessarily prefer the items with higher predicted ratings (Shani and Gunawardana 2011). Actually, plenty of useful information in addition to predicted rating is still largely underexploited in recommender systems. Particularly, *confidence* of rating prediction, defined as the recommender system's trust in its prediction (Shani and Gunawardana 2011), bears important information that can be used to improve the recommendations. In practice, a user often has several alternatives in the recommendation results with different confidences. The confidence weighs in the user's decision significantly when the rating itself is not sufficient to make conclusive decisions.

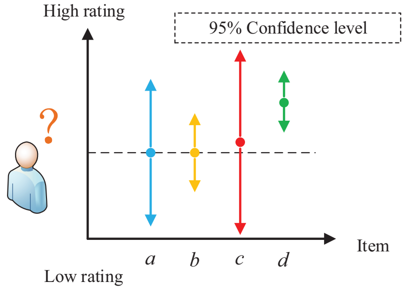


Figure 1: An example of predicted ratings with confidence.

One of the most common measuring methods of confidence is prediction interval (Shani and Gunawardana 2011). It is the interval around the predicted value where the true value lies with a predefined confidence level, e.g. 95% (Hahn and Meeker 2011). A smaller prediction interval means a higher confidence of the prediction. As shown in Figure 1, we can observe the predicted ratings and the related prediction intervals of a user on four movies. The ratings of movie a and b are identical while the confidence of b's rating is higher, which means the user prefers b to a more possibly. However, traditional methods cannot decide which of a and b is more preferred without consideration of confidence. Similarly, the confidence of c's rating is much lower than that of b though c's rating is slightly higher. The user dose not necessarily choose c over b. Thus an ideal recommendation should be the item with both high rating and high confidence such as the movie d.

There are some pilot efforts on utilizing confidence in recommender systems. McLaughlin and Herlocker (2004) proposed to artificially set probability distribution of different

[*]Corresponding author

grades to quantify the rating confidence. Mazurowski (2013) proposed to assemble ratings and empirically find confidence interval endpoints. However, there are several limitations in the existing works. First, previous works tend to use heuristic rules, instead of principled models, to quantify confidence in the rating prediction. Hence, the performance or reliability of the confidence measures can hardly be guaranteed. Second, previous works focus on measuring the rating confidence from user perspective, rather than jointly modeling confidence for both users and items. Third, previous works often quantify the rating confidence in a way independent of the rating prediction. This is not optimal since the confidence of ratings could also be helpful to improve the accuracy of rating prediction.

To address these challenges, in this paper, we present a comprehensive framework focusing on modeling confidence information in MF methods. Specifically, we propose a general *Confidence-aware Matrix Factorization* (CMF) framework to optimize accuracy of rating prediction and measure the prediction confidence simultaneously. Based on the general MF framework, we use complete distribution to infer the rating and confidence structure. First, variance parameters of users and items are introduced for measuring confidence of rating prediction via prediction interval. In this way, influence of users and items on rating variances are both taken into account. Second, we estimate the latent factors for rating prediction and confidence measurement in a unified framework. Specifically, we propose two implementations, i.e., *Confidence-aware Probabilistic Matrix Factorization* and *Confidence-aware Bayesian Probabilistic Matrix Factorization*, with gradient descent and Bayesian inference, respectively. Thus the joint modeling process also improves the accuracy of rating prediction. Third, by combining confidence and accuracy of rating prediction, we provide a *Confidence-aware Ranking* (CR) method to produce top-K recommendations. Finally, extensive experiments on real-world datasets validate the effectiveness of our approach with respect to multiple evaluation metrics.

2 Related Work

Related work can be grouped into two categories. The first category includes the work on matrix factorization models used in recommender systems. The second category covers pilot studies on confidence-aware recommender systems.

Matrix Factorization Models. In the last decade, matrix factorization (MF) has become one of the most popular collaborative filtering (CF) methods due to its superior accuracy and efficiency. The general idea of MF is to use a small amount of latent factors to estimate observed ratings. As one classic MF model, *Probabilistic Matrix Factorization* (PMF) (Mnih and Salakhutdinov 2008) factorized the rating matrix into two factor matrices representing user and item latent features, respectively. Usually, both ratings and latent factors were assumed to follow Gaussian distributions. Many other matrix factorization methods could be viewed as variants of PMF. For example, *Bayesian Probabilistic Matrix Factorization* (BPMF) (Salakhutdinov and Mnih 2008) made further efforts by imposing Gaussian-Wishart priors over the latent factors and using Markov chain Monte Carlo for parameter inference; *Sparse Covariance Matrix Factorization* (SCMF) (Shi et al. 2013) adopted a sparse covariance prior to reflect the semantics more appropriately and to prevent overfitting; *Sparse Probabilistic Matrix Factorization* (SPMF) (Jing, Wang, and Yang 2015) utilized a Laplacian distribution to solve the sparsity and long-tail problem.

However, conventional MF models focused on accuracy optimization neglecting plenty of other information. In contrast, we propose a general framework for MF methods which can combine the optimization of accuracy and the estimation of confidence.

Confidence-aware Recommender Systems. Confidence has been mentioned in recommender systems for a long time. For example, Herlocker, Konstan, and Riedl (2000), Swearingen and Sinha (2001) studied the phenomenon of confidence in recommender systems and argued that confidence scores benefit users in many cases. Shani and Gunawardana (2011) listed confidence as an important evaluation metric and emphasized the requirement for confidence estimates when facing similar performance on other metrics.

For measuring confidence in recommendation results, traditional methods usually relied on heuristic rules instead of principled models. For example, McNee et al. (2003) used the amount of ratings as a confidence metric for each item; McLaughlin and Herlocker (2004) mapped each rating to a predefined difference distribution so that confidence values were available for making recommendations; Hernando et al. (2013), Zhang, Guo, and Chen (2016) defined the confidence estimate using some key features for describing the uncertainty of predictions. For comparison, there are also some works about heteroscedasticity offering more principled solution in the literature. E.g., Shrestha and Solomatine (2006), Le, Smola, and Canu (2005) introduced how to generate prediction uncertainty in Logistic Regression by modifying the form of variance in the model.

Clearly, traditional methods could not provide a principled approach to measure and utilize the confidence in recommendation results. Moreover, it should be noted that all these solutions merely focused on users' influence on the rating confidence while ignoring items' influence. In this paper, we present a comprehensive framework that can optimize accuracy of rating prediction and estimate the prediction confidence, where influences of users and items on rating confidence are both taken into account.

3 Confidence-aware Matrix Factorization

As we have indicated before, traditional MF methods mainly focus on accuracy, which is insufficient for recommendations. In this section, we first propose the framework to capture accuracy and confidence simultaneously. Then we specifically introduce how to obtain confidence of ratings from CMF. At last, one of the applications of confidence, i.e., Confidence-aware Ranking, are proposed.

Framework Definition

MF methods usually use two latent feature matrices $U \in \mathbb{R}^{D\times N}$ and $V \in \mathbb{R}^{D\times M}$ to represent the potential influence on ratings from both users and items, assuming that each rating is independent distributed with different means and the same variance. Suppose there are M items and N users. The rating of user i for item j is denoted by R_{ij}. Let U_i and V_j respectively represent user-specific and item-specific latent feature vectors for user i and item j. And the rating data $R \in \mathbb{R}^{N \times M}$ is supposed to obey a certain distribution $\mathcal{P}$. Thus we could model the probability form via:

$$R_{ij} \sim \mathcal{P}(R_{ij}|U_i^T V_j, \alpha^{-1}), \tag{1}$$

where $\mathcal{P}(x|\mu, \alpha^{-1})$ denotes a certain distribution with mean μ and precision α. Here we use precision instead of variance to facilitate the derivation of mathematical expressions.

Furthermore, the prior distributions $\widetilde{\mathcal{P}}$ over U and V could be given by:

$$U_i \sim \widetilde{\mathcal{P}}(U_i|0, \alpha_U^{-1} I), \tag{2}$$
$$V_j \sim \widetilde{\mathcal{P}}(V_j|0, \alpha_V^{-1} I). \tag{3}$$

Specially, if we suppose that the conditional distribution $\mathcal{P}$ and prior distribution $\widetilde{\mathcal{P}}$ all follow Gaussian distribution, it is transformed to classical PMF model. In PMF, maximizing the logposterior here can be replaced by minimizing the sum-of squares error function with quadratic regularization terms (Mnih and Salakhutdinov 2008). And then we can find a local minimum of the objective function by performing descent methods.

However, all variances of ratings are viewed as the same value, i.e., α^{-1} in Equation (1). It is quite inconsistent with the common sense since distinguishable rating perturbation appears everywhere in the real world. For example, capricious users tend to give an extremely high or low rating due to any aspect of the item while cautious users could obtain a relatively objective rating based on comprehensive evaluation. In addition, variance information is not a simple addendum but an important component in the modeling process since prediction accuracy and uncertainty are complementary to each other.

Therefore, it is meaningful to impose variance effects on ratings of different items by different users both for reflecting the underlying influence of variance properly and for obtaining confidence-aware models. Thus, we propose an integrated framework to solve this problem by modeling confidence via variance information. To achieve this, the confidence-aware distributions can be employed which simultaneously consider the role of users and items. In accordance with the key idea of MF methods, we set γ_{U_i} and γ_{V_j} respectively as the variance parameters of user i and item j. Thus the final variance expression $\sigma^2 = (F(\gamma_{U_i}, \gamma_{V_j}) \cdot \alpha)^{-1}$ is a combination of the two parts where F is the combination function. In this case, the overall likelihood term can be given by:

$$P(R|U, V, \alpha, \gamma_U, \gamma_V) = \prod_i^N \prod_j^M \left[\mathcal{P}(R_{ij}|U_i^T V_j, (F(\gamma_{U_i}, \gamma_{V_j}) \cdot \alpha)^{-1}) \right]^{I_{ij}}, \tag{4}$$

where I_{ij} is the indicator variable. It is equal to 1 if movie j has been rated by user i and equal to 0 otherwise. α is a constant which does not depend on the parameters. $\gamma_U \in \mathbb{R}^N$ and $\gamma_V \in \mathbb{R}^M$ are the variance parameter vectors respectively for users and items.

In this way, we integrate variance information into the model itself. And variance would be controlled automatically in training process.

Measuring Confidence. There are two most common metrics of confidence (Shani and Gunawardana 2011). One metric is the probability that the predicted value is indeed true. It is less popular since it is inconvenient to give the probability for a point in the continuous distribution. Thus we adopt the other one: prediction interval, which is the interval around the predicted value where the true value lies with a predefined confidence level, e.g. 95% (Hahn and Meeker 2011). Prediction interval, as one form of interval estimation (Martin 2012; Gardner and Altman 1986), is usually used for estimating the value of next sample variable. As a result, prediction interval is widely used for measuring prediction uncertainty in machine learning (Shrestha and Solomatine 2006; Kasiviswanathan et al. 2013; Guan et al. 2013).

Prediction interval can be easily generalized when the complete information of target distribution is known. Traditional MF methods are incapable to generate individualized intervals since the variances of ratings are viewed all the same. Just the opposite, our framework is able to fill in the missing part. Let us take the most common distribution, i.e., Gaussian distribution, as an example to show how to calculate a prediction interval for a future observation R_{ij}. With the help of variance information, we could naturally obtain the prediction interval for each rating as the form of:

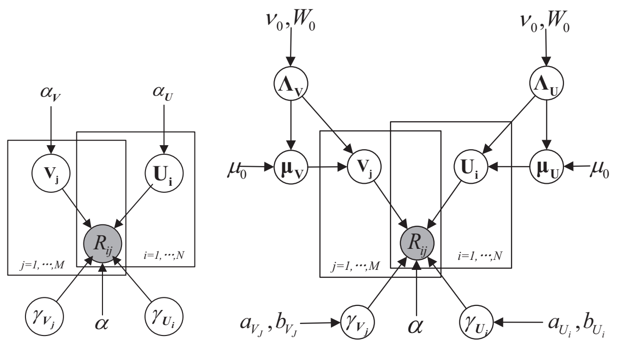
$$\begin{aligned}&[U_i^T V_j - Z(\frac{p}{2}) * (F(\gamma_{U_i}, \gamma_{V_j}) \cdot \alpha)^{-\frac{1}{2}}, \\ &\qquad U_i^T V_j + Z(\frac{p}{2}) * (F(\gamma_{U_i}, \gamma_{V_j}) \cdot \alpha)^{-\frac{1}{2}}],\end{aligned} \tag{5}$$

where $Z(p)$ is the quantile in the standard Gaussian distribution with confidence probability p.

Applications and Confidence-aware Ranking

Given the confidence information, many strategies can be used for recommender systems to provide improvements (McLaughlin and Herlocker 2004). E.g., we can detect low confidence recommendations, produce more valuable recommendation lists, help users achieve their personalized desired risk/benefit trade-off and so forth. Among them, *Confidence-aware Ranking* (CR) we proposed is a combined application of accuracy and confidence.

In recommender systems, many efforts have been made to improve top-K ranking utilizing information besides ratings, e.g. similarity, novelty, diversity, coverage and so forth (Hurley and Zhang 2011; Ostuni et al. 2013; Guan et al. 2013; Wu et al. 2016b). However, as regards ratings, traditional methods are mostly based on just single values. With the



(a) CPMF (b) CBPMF

Figure 2: Graphical models for CPMF and CBPMF.

help of CMF, we could try to obtain a good trade-off between mean and variance rather than only rank mean values.

Quite analogously, it is significant to balance the benefit (usually measured by mean) and risk (usually measured by variance) for portfolios in finance (Shen, Wang, and Ma 2014). And the Sharpe Ratio has become one of the most referenced benefit/risk measures (Sharpe 1994; Ledoit and Wolf 2008). We could also design the CR method by introducing Sharpe Ratio to recommender systems:

$$SharpeRatio = \frac{\hat{R}_{ij} - R_0}{\sigma_{ij}}, \tag{6}$$

where $\hat{R}_{ij} = U_i^T V_j$ is the mean of predicted rating and σ_{ij} is the standard deviation. Here R_0 is a constant and represents the benchmark rating which means items below this boundary may be unacceptable.

To be specific, CR includes two steps: 1) a candidate list of more than K elements is generated by ranking mean of the predicted rating; 2) the candidate list is rearranged by ranking in the order of the defined value of Sharpe Ratio in Equation (6). Thus we could easily obtain the Top-K recommendations according to the ranking. Financially, Sharpe Ratio is usually utilized to determine the priority of products in portfolios to balance risk and benefit. Quite similarly, we adopt Sharpe Ratio to address the trade-off between accuracy and confidence.

4 CMF Implementation

WOur framework can be implemented with multiple MF models. The focus of evaluation part is how our proposed confidence-aware models outperform the original non-confidence-aware methods. Thus, we decided to introduce two cases of CMF implementation, namely *Confidence-aware Probabilistic Matrix Factorization* (CPMF) and *Confidence-aware Bayesian Probabilistic Matrix Factorization* (CBPMF), due to PMF and BPMF's superiority of performance and popularity in recommender systems.

CPMF Model

The structure of CPMF is shown in Figure 2(a), where all distributions are assumed to be Gaussian. We provide a general structure of combining variances. To be specific in this paper, we choose an intuitive structure since it is able to verify the effect of confidence. We define the combination function F in Equation (4) as the product of user-specific and item-specific variance parameters. It follows that the overall likelihood term is given by:

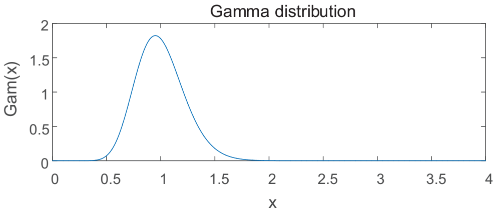


Figure 3: The P.D.F of Gamma distribution with $(a = 20, b = 20)$.

$$P(R|U, V, \alpha, \gamma_U, \gamma_V) = \prod_i^N \prod_j^M \left[\mathcal{N}(R_{ij} | U_i^T V_j, (\gamma_{U_i} \gamma_{V_j} \alpha)^{-1}) \right]^{I_{ij}}. \tag{7}$$

The CPMF has the advantage of simplicity yet the disadvantage of being sensitive to hyperparameters. When inappropriate hyperparameters are used, CPMF may not generalize well due to the issue of overfitting. We can use cross-validation procedures to detect overfitting and optimize the hyperparameters, but the process is computationally expensive for large datasets. An alternative approach is to introduce Bayesian inference which can automatically control the modeling complexity and improve the generalization power of the model. Our confidence-aware Bayesian approach is named by CBPMF as follows.

CBPMF Model

In order to characterize the rating distribution more appropriately, we introduce the variance parameters γ_U and γ_V in CBPMF and place Gamma distribution with shape a and rate b over them:

$$\gamma_{U_i} \sim \boldsymbol{\Gamma}(\gamma_{U_i} | a_{U_i}, b_{U_i}), \tag{8}$$
$$\gamma_{V_j} \sim \boldsymbol{\Gamma}(\gamma_{V_j} | a_{V_j}, b_{V_j}). \tag{9}$$

As shown in Figure 3, Gamma distribution is unimodal when $a > 1$. Both chi-squared distribution and exponential distribution are special cases of Gamma distribution. Gamma distribution is often used as the natural conjugate prior for Gaussian distribution with unknown precision (Salakhutdinov and Mnih 2008; Murphy 2007; Diaconis, Ylvisaker, and others 1979). The probability density function of Gamma distribution is defined as:

$$\boldsymbol{\Gamma}(x|a, b) = \frac{\beta^\alpha}{\Gamma(a)} x^{a-1} e^{-bx}.$$

The mean is given by a/b. Since our model degrades to the traditional BPMF model when γ_{U_i} and γ_{V_j} both equal to 1, we set 1 as the initial mean value for γ_{U_i} and γ_{V_j}.

Similarly to BPMF (Salakhutdinov and Mnih 2008), we introduce Gaussian-Wishart priors for the factor hyperparameters U and V. First, the prior distributions over factors are assumed to be Gaussian:

$$U_i \sim \mathcal{N}(U_i|\mu_U, \Lambda_U^{-1}), \tag{10}$$

$$V_j \sim \mathcal{N}(V_j|\mu_V, \Lambda_V^{-1}), \tag{11}$$

where $\mathcal{N}(x|\mu, \alpha^{-1})$ denotes Gaussian distribution with mean μ and precision α. Then we define factor hyperparameter $\Theta_U = \{\mu_U, \Lambda_U\}$ and $\Theta_V = \{\mu_V, \Lambda_V\}$, following Gaussian-Wishart priors.

On the basis of Bayesian inferences, we can model the posterior distribution over latent factors U and V as:

$$\begin{aligned} P(U, V|R, \alpha,\Theta_U, \Theta_V, \gamma_U, \gamma_V) \propto \\ P(R|\alpha, \gamma_U, \gamma_V)P(U|\Theta_U)P(V|\Theta_V). \end{aligned} \tag{12}$$

Figure 2(b) shows the graphical model of CBPMF. In order to solve the model, we have recourse to approximate inference since the predicted distribution is analytically intractable. Markov Chain Monte Carlo (MCMC) methods (Neal 1993) have been widely applied to solve approximation problems in large dimensional spaces (Andrieu et al. 2003). They aim at achieving a stationary distribution which is equal to the posterior distributions in the model by constructing a Markov chain. Among them, Gibbs sampling algorithm performs great as long as conditional distributions can be easily sampled. On the other hand, Gibbs sampling is highly simple for implementing. It successively samples every variable from its distribution conditional on present values of rest variables.

Parameter Estimation

In CPMF, we could directly perform gradient descent algorithms to optimize the latent factors in the model. Since this procedure is quite standard, we omit the details due to the space limit. In the remainder, we will focus on the parameter estimation and inference in CBPMF.

We use Gibbs sampling procedure to compute CBPMF. Given our model specification, it is easy to sample parameters and hyperparameters from conditional distributions owing to the adoption of conjugate priors. We have accordingly designed an efficient generative process based on Gibbs sampling in Algorithm 1. For sake of simplicity, we only elaborate details on the sampling of user-specific factors (e.g., U). Sampling of item-specific parameters can be implemented similarly.

Algorithm 1 Gibbs sampling for CBPMF

```
Initialize factor parameters $U^0, V^0$.
for $t = 1$ to $T$ do
    Draw factor hyperparameters $\Theta_U^t$ and $\Theta_V^t$ from Gaussian-Wishart distributions: $P(\Theta_U^t|U^{t-1})$ and $P(\Theta_V^t|V^{t-1})$ like Equation (14).
    Draw variance parameters $\gamma_U^t$ and $\gamma_V^t$ from Gamma distributions: $\mathbf{\Gamma}(\gamma_{U_i}^t|a_{U_i}^{*t}, b_{U_i}^{*t})$ and $\mathbf{\Gamma}(\gamma_{V_j}^t|a_{V_j}^{*t}, b_{V_j}^{*t})$.
    Draw factor parameters $U^t$ and $V^t$ from Gaussian distributions: $\mathcal{N}(U^t|\mu_U^{*t}, [\Lambda_U^{-1}]^{*t})$ and $\mathcal{N}(V^t|\mu_V^{*t}, [\Lambda_V^{-1}]^{*t})$.
    Draw each rating $R_{ij}$ from $\mathcal{N}(R_{ij}|(U_i^t)^T V_J^t, (\gamma_{U_i}^t \gamma_{V_j}^t \alpha)^{-1})$.
end for
```

Sample factor hyperparameter Θ_U. According to Bayesian rules, we could model the posterior distribution over Θ_U as:

$$P(\Theta_U|U) \sim P(U|\Theta_U)P(\Theta_U), \tag{13}$$

where

$$P(\Theta_U) \sim \mathcal{N}(\mu_U|\mu_0, (\beta_0\Lambda_U)^{-1})\mathcal{W}(\Lambda_U|W_0, \nu_0),$$

$$\mathcal{W}(\Lambda_U|\nu_0, W_0) = \frac{1}{C}|\Lambda_U|^{\frac{\nu_0 - D - 1}{2}} \exp(-\frac{1}{2}tr(W_0^{-1}\Lambda_U)).$$

Here $\Theta_0 = \{\mu_0, \nu_0, W_0\}$ and C are constants. D is the dimension of latent factor. $\mathcal{W}$ represents Wishart distribution.

And due to the adoption of conjugate prior, we can write $P(\Theta_U|U)$ as the form of Gaussian-Wishart distribution:

$$P(\Theta_U|U) = \mathcal{N}(\mu_U|\mu_0^*, (\beta_0^*\Lambda_U)^{-1})\mathcal{W}(\Lambda_U|W_0^*, \nu_0^*), \tag{14}$$

where

$$\bar{U} = \tfrac{1}{N}\textstyle\sum_{i=1}^N U_i, \quad \bar{S} = \tfrac{1}{N}\textstyle\sum_{i=1}^N (\mu_i - \bar{U})(\mu_i - \bar{U})^T,$$

$$(W_0^*)^{-1} = W_0^{-1} + N\bar{S} + \tfrac{\beta_0 N}{\beta_0 + N}(\mu_0 - \bar{U})(\mu_0 - \bar{U})^T,$$

$$\mu_0^* = \tfrac{\sum_{i=1}^N U_i + \beta_0\mu_0}{\beta_0 + N}, \quad \nu_0^* = \nu_0 + N, \quad \beta_0^* = \beta_0 + N.$$

Sample variance parameter γ_U. For γ_{U_i}, Bayesian inference is used with all related variables. Note that we place Gamma distribution over variance parameters. We denote $R_{i*} = \{R_{i1}, ..., R_{iM}\}$. Thus we can simply write the posterior distribution as

$$\begin{aligned} P(\gamma_{U_i}|R_{i*}) &= \mathbf{\Gamma}(\gamma_{U_i}|a_{U_i}^*, b_{U_i}^*) \\ &\sim \textstyle\prod_j^M P(R_{ij}|\alpha_{U_i})^{I_{ij}} P(\alpha_{U_i}), \end{aligned} \tag{15}$$

where

$$a_{U_i}^* = a_{U_i} + \tfrac{1}{2}\textstyle\sum_j^M I_{ij}, \tag{16}$$

$$b_{U_i}^* = b_{U_i} + \tfrac{1}{2}\textstyle\sum_j^M I_{ij}\alpha\gamma_{U_i}\gamma_{V_j}(R_{ij} - U_i^T V_j)^2. \tag{17}$$

Sample factor parameter U. For U_i, the conditional distribution is still Gaussian on present values of other variables:

Table 1: The statistics of the datasets.

Dataset	Users	Items	Ratings	sparsity
MovieLens	6,040	3,952	1M	4.19%
Netflix	480,198	17,770	100M	1.18%
Jester	59,132	140	1.8M	21.28%

$$P(U_i|R_{i*}, V, \Theta_U, \gamma_U) = \mathcal{N}(U|\mu_U^*, [\Lambda_U^*]^{-1})$$
$$\sim \prod_j^M \left[\mathcal{N}(R_{ij}|U_i^T V_j, (\gamma_{U_i}\gamma_{V_j}\alpha)^{-1})\right]^{I_{ij}} P(U|\Theta_U), \tag{18}$$

where

$$\Lambda_{U_i}^* = \sum_j^M [\alpha\gamma_{U_i}\gamma_{V_j} V_j V_j^T]^{I_{ij}} + \Lambda_U, \tag{19}$$
$$\mu_{U_i}^* = [\Lambda_{U_i}^*]^{-1} \left(\sum_j^M [\alpha\gamma_{U_i}\gamma_{V_j} R_{ij} V_j]^{I_{ij}} + \Lambda_U \mu_U\right). \tag{20}$$

5 Experiments

In this section, we mainly evaluate the proposed CMF framework on three real-world datasets from three perspectives. Specifically, we will discuss: (1) the accuracy of our framework on rating prediction compared with baselines, (2) the effectiveness on confidence measurement, (3) how CR works on top-K recommendations.

Dataset Description

We conduct experiments on three real-world datasets, i.e., MovieLens, Netflix and Jester. MovieLens[1]and Netflix[2] have been extensively exploited in recommender system experiments. They are composed of ordinal values from 1 to 5 for movies. Differently, Jester[3] comprises continuous ratings (-10.00 to +10.00) for jokes. In the experiments, ratings in all datasets are mapped into the same numeric range (0 to 5) for comparative purposes. Table 1 shows the basic statistical properties of the datasets.

Baselines and Evaluation Metrics

We compared our proposed model CBPMF with four baselines: PMF, bias-PMF, CPMF, BPMF. PMF and BPMF have been widely used in recommender systems and perform well in the literature. Bias-PMF (Koren, Bell, and Volinsky 2009) is a variant of PMF considering biases information. CPMF is the realization of our framework on PMF. Though lacking strict theoretical guarantee, empirical result shows that it is easy to converge when exploiting PMF's results as parameter initialization.

To measure the performances of our framework, we adopt different metrics for distinct perspectives. First, for single value estimate evaluation, the widely-used *Root Mean Squared Error* (RMSE) (Mnih and Salakhutdinov 2008) is used on all baselines and datasets to demonstrate the practical effectiveness of CBPMF. Second, we use the *Coverage Percentage* (CP) for interval estimate evaluation compared with BPMF. Here, CP is defined as the percentage that ratings fall into the prediction intervals to measure the quality of interval estimation:

$$CP = \sum_{R_{ij} \in Z} \frac{I(R_{ij} \in H_{ij})}{|Z|}, \tag{21}$$

where Z denotes the rating set of test data, and H_{ij} is the prediction interval for user i and item j. Third, the *Normalised Discounted Cumulative Gain* (NDCG) and *Average precision* (AP) (Zhu et al. 2015) are used from ranking perspective to measure the top-K recommendations.

Experimental Settings

In the proposed CBPMF, we introduced variance parameters γ_U and γ_V which obey Gamma distribution. As discussed in Section 3, the shape and rate of Gamma distribution is initialized via $a = b$ so that the mean value would always be 1. Here, a and b are of great importance to statistical dispersion of Gamma distribution. Larger a and b would lead to more compact sampling results. We tune the value of a and b from the candidate set $\{5, 10, 20, 30, 50, 100, 200\}$.

In PMF, following Mnih and Salakhutdinov(2008), we divide the datasets into mini-batches and update parameters after every mini-batch. In BPMF, following salakhutdinov and Mnih(2008), we set $\mu_0 = 0$ by symmetry, $\nu_0 = D$ where D is the dimension of latent factors and $W_0 = I_{D \times D}$ which is the identity matrix. To speed up the training process, the Gibbs sampler is initialized with PMF's output estimate. All parameters are tuned following the authors.

In our experiments, we randomly select 80% to 90% data as training sets according to different datasets and the rest part as test sets for five times. Each user and item are ensured to be included by training sets in the selecting process. Thus the final result is shown by average.

Experimental Results

Accuracy Evaluation. We first compare the result of rating prediction to validate the prediction accuracy of our framework. As mentioned before, when variance information is introduced into the model, they not only contribute to building prediction intervals, but also help the model fit better. The RMSE results under varying latent dimension D on three datasets are shown in Figure 4. The smaller RMSE signifies the better prediction.

We can observe that CBPMF outperforms all other models in any case, which proves the effect of introduced variance information for model fitting. Variance serves as an important part in our models, instead of additional information in the post-modeling process. Thus, our models could outperform their non-confidence-aware versions. Furthermore, we can observe that the performances of PMF, bias-PMF and CPMF all begin to fall back when factor dimensionality D grows. However, the performances of BPMF and CBPMF are able to keep on improving even facing growing model

[1] http://www.grouplens.org/node/73

[2] www.netflixprize.com

[3] http://eigentaste.berkeley.edu/dataset/

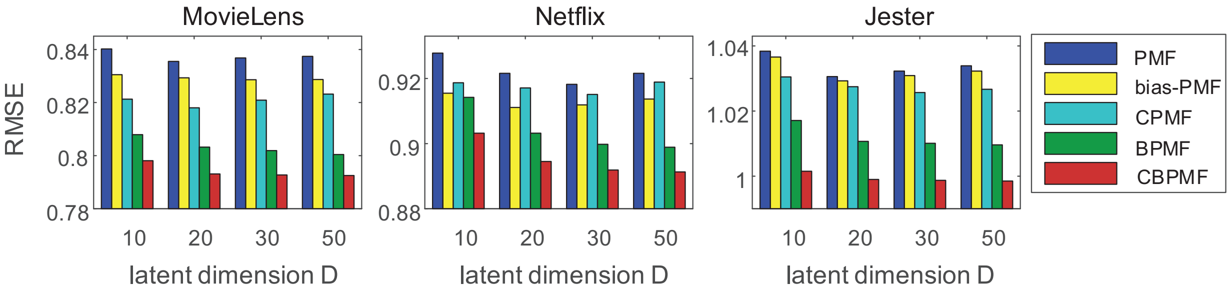


Figure 4: Overall comparison results of RMSE.

complexity. For more details, Figure 5 shows the convergence process of PMF, BPMF and CBPMF on MovieLens when $D = 20$. It can be seen clearly that PMF has been in trouble with overfitting, while BPMF and CBPMF have no such problem. Here BPMF and CBPMF start quite better than PMF because we use the result of PMF as initialization.

Confidence Measurement Evaluation. In this part, we concentrate on confidence measurement which presents confidence information for predicted ratings. Nonetheless variances in BPMF are set all the same, we can still obtain prediction intervals from them. Such intervals would have exactly the same length which are certainly inappropriate. We compare the intervals derived from BPMF and CBPMF when confidence level is 90% and 95%. CP is used to measure the probability that prediction intervals actually cover the true value. It should be noted that a higher CP does not mean a better performance since higher CP may imply too large interval length which is also inappropriate. A model has good performance only when the value of CP is close to the confidence level. The closer, the better.

The CP results are shown in Table 2. The difference bewteen CP and confidence level is listed in parenthesis for comparison. As we can see from the table, prediction intervals derived from CBPMF are remarkably closer to the confidence level, which demonstrates our framework's capacity for describing confidence in recommender systems. For example, BPMF's difference bewteen CP and confidence level is about four times bigger than CBPMF on MovieLens. Besides, it is noticeable that CBPMF gains larger improvements on CP value compared to BPMF on Jester than on MovieLens and Netflix. A possible explanation is that ratings of MovieLens and Netflix are discrete while ratings of Jester could be viewed as approximately continuous, and variance is used for describing the predicted probability distribution which is also continuous.

Top-K Recommendations Evaluation. As introduced in Section 3, CR gives a confidence view for top-K recommendations. We could exploit the accuracy and confidence information of rating distributions instead of mean values to sort recommendation lists.

In the experiment, we believe user likes one item when he or she gives the rating no less than 4 scores. Note that $R_{ij} = 0$ does not imply user i dislikes item j. It just means the absence of record. Thus the data is highly imbalanced with too much negative samples. We deal with the problem by using an effective undersampling technique. In detail, we randomly select q missing ratings for each positive sample. Since the negative samples randomly change every time. In this way, missing ratings would give quite weak negative signals (Wu et al. 2016a).

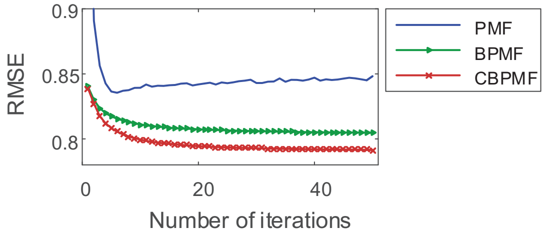


Figure 5: Convergence process on Movielens (D=20).

The results of NDCG@K and AP@K are shown in Figure 6. Here, R_0 is set to 3.8. AP@K means the AP value when we recommend K items to each user and so does NDCG@K. Both AP and NDCG are the larger, the better. We can see that even pure CBPMF has outperformed BPMF, which verifies the function of variance information for model fitting. Moreover, with the help of CR, our model can get even better results. It's interesting to notice that CR's effect decreases with the larger K since CR is more and more closer to single value ranking using precision metrics.

Computational Complexity. The computational complexity of parameter estimation is $O(T(MD^3 + ND^3 + YD^2))$, where T is the number of iterations, N, M and Y are the quantity of users, items and observed ratings. And the computation of Sharpe Ratio is quite convenient and direct in our framework with the complexity $O(Y)$.

Analyzing the Variance Parameters on MovieLens

Our proposed framework is able to capture each movie and user's unique characteristic on confidence in the form of variance parameters. In our framework, bigger variance parameter implies higher confidence. We next present some interesting results on MovieLens since MovieLens provides some extra information besides ratings. Figure 7 shows the variance parameters on different users' age ranges, genders and movies' genres. We can observe that elder users tend to have bigger variance parameters, which indicates less un-

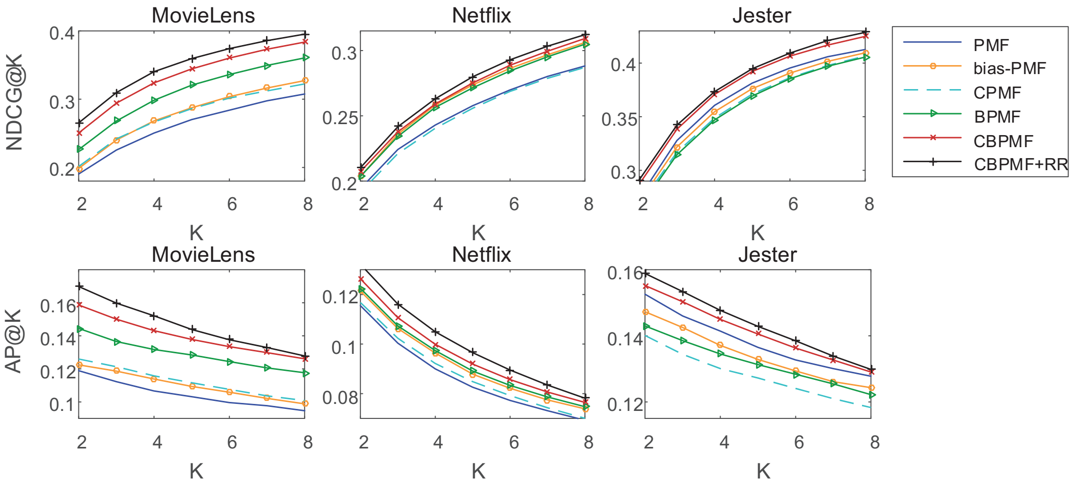


Figure 6: Comparison results of NDCG@K and AP@K.

Table 2: The comparison results of CP.

Dataset	confidence level 90%	
	BPMF	CBPMF
MovieLens	86.26% (3.74%)	**89.24% (0.76%)**
Netflix	82.16% (7.84%)	**86.48% (3.52%)**
Jester	78.06% (11.94%)	**86.57% (3.43%)**
Dataset	confidence level 95%	
	BPMF	CBPMF
MovieLens	91.14% (3.86%)	**94.26% (0.74%)**
Netflix	87.88% (7.12%)	**91.73% (3.27%)**
Jester	84.15% (10.85%)	**92.56% (2.44%)**

certainty on ratings. Besides, males tend to have bigger variance parameters than females, which may because males are more rational, and females are more perceptual. On the other hand, movies with more non-realistic elements seem to show more uncertainty on ratings. Thus we must make diverse recommendation strategies for users of different ages, different genders and movies of different genres. For example, we may avoid recommending items with high uncertainty to elder users, who usually value tradition and stability more. Also, female users may tend to more novelty because they often have a more wide interest spectrum.

6 Concluding Remarks

In this paper, we proposed a general Confidence-aware Matrix Factorization framework, which can simultaneously optimize the accuracy of rating prediction and measure the prediction confidence for recommender systems. Particularly, variance parameters from both users' and items' perspectives were considered, and prediction intervals were effectively utilized for measuring confidence in the recommendation results. We provided two implementations of our framework: Confidence-aware Probabilistic Matrix Factorization and Confidence-aware Bayesian Probabilistic Matrix Factorization. We also designed the Confidence-aware Ranking, a Sharpe Ratio based ranking method for top-K recommendations. By combining accuracy and confidence, our model outperformed alternative methods on prediction accuracy, confidence measurement and top-K recommendations.

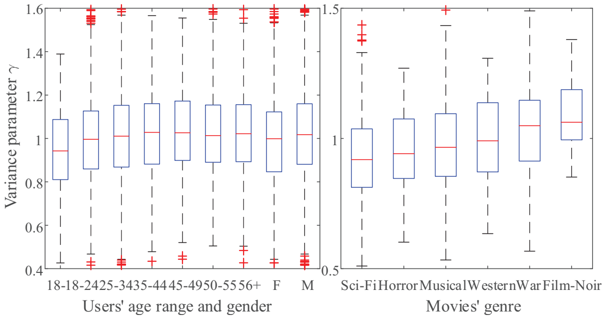


Figure 7: Presentation of variance parameters on users' age ranges, genders and movies' genres. In the right panel, we only present the top 3 and last 3 genres in the order of median variance parameters.

7 Acknowledgements

This research was partially supported by grants from the National Key Research and Development Program of China (Grant No. 2016YFB1000904), the National Natural Science Foundation of China (Grants No. 61727809, 61672483 and U1605251), and the Youth Innovation Promotion Association of CAS (No. 2014299).

References

Andrieu, C.; De Freitas, N.; Doucet, A.; and Jordan, M. I. 2003. An introduction to mcmc for machine learning. *Machine learning* 50(1-2):5–43.

Diaconis, P.; Ylvisaker, D.; et al. 1979. Conjugate priors for exponential families. *The Annals of statistics* 7(2):269–281.

Dror, G.; Koenigstein, N.; Koren, Y.; and Weimer, M. 2012.

The yahoo! music dataset and kdd-cup11. In *Proceedings of KDD Cup 2011*, 3–18.

Gardner, M. J., and Altman, D. G. 1986. Confidence intervals rather than p values: estimation rather than hypothesis testing. *Br Med J (Clin Res Ed)* 292(6522):746–750.

Guan, C.; Luh, P. B.; Michel, L. D.; and Chi, Z. 2013. Hybrid kalman filters for very short-term load forecasting and prediction interval estimation. *IEEE Transactions on Power Systems* 28(4):3806–3817.

Hahn, G. J., and Meeker, W. Q. 2011. *Statistical intervals: a guide for practitioners*, volume 328. John Wiley & Sons.

Herlocker, J. L.; Konstan, J. A.; and Riedl, J. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, 241–250. ACM.

Hernando, A.; Bobadilla, J.; Ortega, F.; and Tejedor, J. 2013. Incorporating reliability measurements into the predictions of a recommender system. *Information Sciences* 218:1–16.

Hurley, N., and Zhang, M. 2011. *Novelty and Diversity in Top-N Recommendation – Analysis and Evaluation*. ACM.

Jing, L.; Wang, P.; and Yang, L. 2015. Sparse probabilistic matrix factorization by laplace distribution for collaborative filtering. In *IJCAI*, 1771–1777.

Kasiviswanathan, K. S.; Cibin, R.; Sudheer, K. P.; and Chaubey, I. 2013. Constructing prediction interval for artificial neural network rainfall runoff models based on ensemble simulations. *Journal of Hydrology* 499(499):275–288.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer* 42(8).

Le, Q. V.; Smola, A. J.; and Canu, S. 2005. Heteroscedastic gaussian process regression. In *Proceedings of the 22nd international conference on Machine learning*, 489–496. ACM.

Ledoit, O., and Wolf, M. 2008. Robust performance hypothesis testing with the sharpe ratio. *Journal of Empirical Finance* 15(5):850–859.

Liu, Q.; Ge, Y.; Li, Z.; Chen, E.; and Xiong, H. 2011. Personalized travel package recommendation. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, 407–416. IEEE.

Lu, J.; Wu, D.; Mao, M.; Wang, W.; and Zhang, G. 2015. Recommender system application developments: a survey. *Decision Support Systems* 74:12–32.

Martin, B. R. 2012. Interval estimation - statistics for physical science - chapter 9. *Statistics for Physical Science* 173191.

Mazurowski, M. A. 2013. Estimating confidence of individual rating predictions in collaborative filtering recommender systems. *Expert Systems with Applications* 40(10):3847–3857.

McLaughlin, M. R., and Herlocker, J. L. 2004. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 329–336. ACM.

McNee, S. M.; Lam, S. K.; Guetzlaff, C.; Konstan, J. A.; and Riedl, J. 2003. Confidence displays and training in recommender systems. In *Proc. INTERACT*, volume 3, 176–183.

Mnih, A., and Salakhutdinov, R. R. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*, 1257–1264.

Murphy, K. P. 2007. Conjugate bayesian analysis of the gaussian distribution. *def* $1(2\sigma2)$:16.

Neal, R. M. 1993. Probabilistic inference using markov chain monte carlo methods.

Ostuni, V. C.; Noia, T. D.; Sciascio, E. D.; and Mirizzi, R. 2013. Top-n recommendations from implicit feedback leveraging linked open data. In *ACM Conference on Recommender Systems*, 85–92.

Salakhutdinov, R., and Mnih, A. 2008. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, 880–887. ACM.

Shani, G., and Gunawardana, A. 2011. Evaluating recommendation systems. *Recommender systems handbook* 257–297.

Sharpe, W. F. 1994. The sharpe ratio. *The journal of portfolio management* 21(1):49–58.

Shen, W.; Wang, J.; and Ma, S. 2014. Doubly regularized portfolio with risk minimization. In *AAAI*, 1286–1292.

Shi, J.; Wang, N.; Xia, Y.; Yeung, D.-Y.; King, I.; and Jia, J. 2013. Scmf: Sparse covariance matrix factorization for collaborative filtering. In *IJCAI*, 2705–2711.

Shrestha, D. L., and Solomatine, D. P. 2006. Machine learning approaches for estimation of prediction interval for the model output. *Neural Networks* 19(2):225–235.

Swearingen, K., and Sinha, R. 2001. Beyond algorithms: An hci perspective on recommender systems. In *ACM SIGIR 2001 Workshop on Recommender Systems*, volume 13, 1–11.

Wu, L.; Ge, Y.; Liu, Q.; Chen, E.; Long, B.; and Huang, Z. 2016a. Modeling users' preferences and social links in social networking services: a joint-evolving perspective. In *Thirtieth AAAI Conference on Artificial Intelligence*, 279–286.

Wu, L.; Liu, Q.; Chen, E.; Yuan, N. J.; Guo, G.; and Xie, X. 2016b. Relevance meets coverage: A unified framework to generate diversified recommendations. *ACM Transactions on Intelligent Systems and Technology (TIST)* 7(3):39.

Zhang, M.; Guo, X.; and Chen, G. 2016. Prediction uncertainty in collaborative filtering: Enhancing personalized online product ranking. *Decision Support Systems* 83:10–21.

Zhao, Z.; Lu, H.; Cai, D.; He, X.; and Zhuang, Y. 2016. User preference learning for online social recommendation. *IEEE Transactions on Knowledge and Data Engineering* 28(9):2522–2534.

Zhu, H.; Xiong, H.; Ge, Y.; and Chen, E. 2015. Discovery of ranking fraud for mobile apps. *IEEE Transactions on knowledge and data engineering* 27(1):74–87.