IJCSN - International Journal of Computer Science and Network, Volume 6, Issue 5, October 2017
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

A Context Aware Recommendation System through Exploring and Optimizing Latent Preferences

[1] Solomon Demissie Seifu; [2] M.Shashi

[1] Department of Computer Science and Systems Engineering, Andhra University
Visakhapatnam, 530003, India

[2] Department of Computer Science and Systems Engineering, Andhra University
Visakhapatnam, 530003, India

Abstract - Recommender system is a tool that provides personalized service to help users to find their desired content. Context-aware recommendations personalize the search for such desired content by considering the contextual information. In many recommendation aspects, incorporating the context of users has been shown to improve the quality of recommendations. In this work, we propose a model that improves the user experience of finding the content they desire by analyzing their contextual information. By conducting an empirical analysis of a dataset from last.fm, we demonstrate the extraction of latent preferences for recommending items under a given context and study how contextual information can be exploited to improve the prediction accuracy of recommender systems. Additionally, we use an optimization function to further minimize the root mean square error (RMSE) measure of the resulted prediction capability of the latent preference models. Finally, we proposed a latent collaborative preference model to predict the final rating of users to items by combining the extracted latent preference models. The experimental results achieved in this work shows our context-aware latent collaborative model can improve the prediction accuracy of recommender systems as compared to state of the art non-context aware approaches.

Keywords - *Collaborative Filtering (CF); context; context-based recommendation; context-aware rating prediction*

1. Introduction

Context-aware recommender systems (CARS) integrate existing contextual information into the recommendation process to model and predict the long-term tastes and preferences of users [5], [6]. The rating information obtained from the user-item relationship doesn't always reflect the user's behavior and may not be accurate. A user, for instance, may not be interested to purchase a product but might like the product and might give a very high rating value. The contextual information, however, helps to overcome the challenge of understanding the rating value in such instances by making it to dynamically reflect user's behavior and be personalized to the context in general.

This paper proposes a solution to get an online content by analyzing the relationship between each user, item, and the context information. By following the work of [19], we propose a context-aware recommendation model which identifies the latent preferences of users toward context, contexts towards items as well as users towards new and selected items. Similar to their work, our model uses tags to explore the latent preferences which mirror the collected contextual information. As different from their work,

however, we evaluate the prediction accuracy of each model through minimizing the root mean squared error (RMSE) and optimize the result via Stochastic Gradient Descent optimization technique to further minimize the prediction error. Finally, we devise a latent collaborative approach for context-based rating prediction by utilizing the extracted latent preference models.

The remainder of this paper is structured as follows. In the next section, we focus on highlighting existing context-aware recommendation methods. In Section 3, we introduce the reader to our proposed model which includes the search for the latent preferences and the defining and solving of an optimization function based on the extracted latent preference models. In Section 4, we present and discuss the results of the experiments and finally we wrap up our work in Section 5.

2. Related Works

Previous pure content or collaborative filtering based recommendation approach is shifting towards a personalized user-centric one that utilizes contextual information due to the reason that this information is useful for providing and improving personalized recommendations [2]. Various recommender approaches boosted with contextual information have been developed for several application domains to enhance recommendation results. To mention some, Shepitsen et al. [1] proposed a context-dependent variant of hierarchical agglomerative clustering of tags to personalize navigational recommendations. Peng et al. [15] devised a joint item-tag recommendation framework by integrating profiles of users and tag information. Tso-Sutter et al. [17] discover a context-based recommendation system by unifying user, item, and tag information into a two-dimension matrix to discover the correlations of unified information. On the other hand, Konstas et al. [11] proposed a context-aware music recommendation system through constructing a social graph by combining users, items and tags information together and then performing Random Walks with Restarts (RWR). A personalized music recommendation in social tagging systems is devised in [11] in which play counts were employed as implicit preferences. Qi et al. [24] computed users' similarities using the inferred tag ratings to infer their preferences. Jamali et al. [23] proposed a random walk model by combining user trust information and item-based collaborative filtering approach for a recommendation.

As different from the context-aware recommendation algorithms mentioned above, the model proposed by this paper doesn't require extraction of features from online contents to analyze user's context. Rather, our model adopted a collaborative approach in situations where user-item-context rating information is not available. Therefore, we construct three latent preference models to search for hidden preferences in other similar users, items, and context logs.

3. Recommendation Model

We conceptually hypothesize that users sharing an item are likely to also share some hidden contextual information. Such contextual information is able to effectively describe the user's preferences toward their selected items. The analysis of the available contextual information coupled with the consumed items enables the analysis of items consumed in similar contexts. Users who consume certain items in a given list of contexts is likely form a contextual pattern to bridge the information gaps between users and new items. The recommendation model thus builds based on such types of associations for predicting and recommending items in different possible contexts. In general, given the contextual tags associated with a user (u) interacting with items (i), the recommendation problem is to identify a list of items I_x that will be of interest to a given user u considering a list of given contexts.

We denote the possible list of contexts as $C=\{c1, c2,\ldots,c_{|C|}\}$, the set of possible items as $I=\{i1, i2,\ldots,i_{|I|}\}$, and the set of users as $U=\{u1, u2,\ldots,u_{|U|}\}$. Fig. 1 depicts the overall workflow for the computation of our recommendation model utilizing the context parameter. Hence, the context parameter we considered in our model is the tag information associated with each artist in last.fm1 dataset which is an online social music radio resource where users can subscribe, listen, and tag their favorite albums or tracks.

3.1 Constructing base matrices

Given a list of users U, a list of items I, and a list of contexts C, we build the required matrices needed for our recommendation model by constructing three main matrices: user-context matrix $\mathbf{UC}_{|U|\times|C|}$, user-item matrix $\mathbf{UI}_{|U|\times|I|}$, and context-item matrix $\mathbf{CI}_{|C|\times|I|}$. Since the relationship between a user U_x and item I_y can exist within a context C_n, the three-dimensional space on users, items, and contexts can be reduced to a two-dimensional space represented by three two-dimensional matrices [19]. The approach adopted in this paper is that matrix $\mathbf{UI}_{|U|\times|I|}$ is built from the rating values assigned to items by users. The matrix $\mathbf{UC}_{|U|\times|C|}$ represents the frequency of user u_x consumed items in context c_y and matrix $\mathbf{CI}_{|C|\times|I|}$ represents the frequency of items I_y selected in a particular context c_y.

3.2 Latent preference modeling

Users who select items in particular contexts are likely to select similar items in similar contexts. Identifying the hidden (latent) features of contexts for both users and items is the main technical issue here. In solving this issue, we analyze the contextual information associated with the interactions of <*user*, *item*> in the dataset. We fill the gap between users and new items as well as between items and new contexts by tracing the patterns of the contextual selection. Our assumption in the proposed recommendation model is that there are items in I for users in U under context C, where the user's preferences are unknown. However, we can build three latent models that represent the latent preferences of users toward contexts $\mathbf{uLc}_{|U|\times|C|}$, the latent preferences of items toward contexts $\mathbf{cLi}_{|C|\times|I|}$, and the users' preferences of items $\mathbf{uLi}_{|U|\times|I|}$.

IJCSN - International Journal of Computer Science and Network, Volume 6, Issue 5, October 2017
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

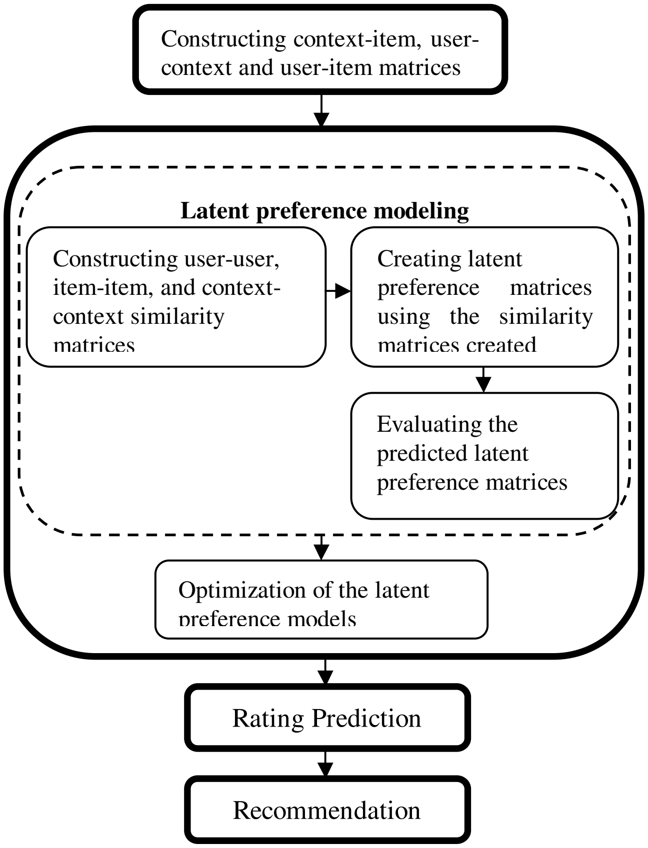

Fig. 1 Pipeline for the recommendation model

3.1.1 Constructing similarity matrices

We construct the user–user, item–item, and context–context similarity matrices for each individual dimension to discover the latent preferences in our model and accordingly leverage relevant items for a user in a particular context. The similarity measure can be computed from the set of three decomposed matrices (**UC**, **CI**, and **UI**) by applying the cosine similarity technique which represents the cosine angle value between two sets of vectors. Cosine similarity approach is preferred for this research due to its constantly good performance in different existing recommendation algorithms [25][26][28].

3.1.1.1 User-user similarity

The reason why we find similar users who share some items is to obtain list of items consumed by given users to find other interesting items consumed by similar users (also called nearest neighbors). Cosine-based similarity takes two vectors of shared items of user u_x, and u_y and quantifies their similarity according to their angle, as in (1). We employ matrix **S** to form the user-user similarity matrix. Hence, we compute such similarity of users $\mathbf{S}_{|U|\times|U|}$ from the decomposed user-item base matrix **UI**.

$$s_{u_x u_y} = \cos(u_x, u_y) = \left(\frac{u_x . u_y}{\|u_x\|^2 . \|u_y\|^2} \right) \quad (1)$$

3.1.1.2 Item-item similarity

Likewise, we compute the similarity of items based on the assumption that user is likely to consume items that are similar to some of what they have already seen in the past. The similarity values can be obtained by measuring the cosine angle between the two column vectors in the matrix **UI**, similar to finding the user-user similarity according to (2).

$$t_{i_x i_y} = \cos(i_x, i_y) = \left(\frac{i_x . i_y}{\|i_x\|^2 . \|i_y\|^2} \right) \quad (2)$$

Based on such similarity computation, the resulted item-item similarity matrix **T** is formed.

3.1.1.3 Context-context similarity

As explained in section 3.1, the context-item matrix **CI** represents the frequency of items selected in a particular context and we utilize this matrix to compute the context-context similarity matrix $\mathbf{X}_{|C|\times|C|}$. Such context similarity represents the semantic relations of two contexts by determining how frequently an item appears in each context. Since the behavior of users toward contexts is not reliable, the similarity of contexts is computed in terms of items. Hence, in our proposed model, we extract the 200 most frequently used contextual tags to perform the similarity computation. Based on cosine similarity computation in (3), the resulting matrix **X** from the collaborative filtering holds the similarity entries $X = \left[x_{c_x c_y} \right]$ where c_x and c_y represent the row and column of context c_x and context c_y.

$$x_{c_x c_y} = cos(c_x, c_y) = \left(\frac{c_x . c_y}{\|c_x\| . \|c_y\|} \right) \quad (3)$$

3.1.2 Exploring latent preferences

The relationship between users and items for different context attributes can pinpoint hidden causes for which an

IJCSN - International Journal of Computer Science and Network, Volume 6, Issue 5, October 2017
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

item is selected in a certain context. It can also identify hidden reasons why users prefer to select certain items in a given context. The similarity matrices we obtained in section 3.2.1 above are helpful to build a model that reflects the hidden preferences of a given user to a context, a given item to a context, and of a given user to an item.

3.1.2.1 Latent user-context preferences extraction

The latent preferences of a user in the current context can be predicted by knowing how a user is behaving in terms of item selection. According to (4), the preferences of a user u for contexts that are similar to another context is captured by matrix $\mathbf{uLc}_{|U|\times|C|}$. Hence, we obtained the latent user–context preferences via dot product of the normalized frequency matrix ($\overline{\mathbf{UC}}$) and the transpose of the similarity matrix $(\mathbf{X})^{\mathbf{T}}$.

$$\mathrm{uLc} = \overline{UC}(X)^T \tag{4}$$

The reason why we normalize the frequency matrix **UC** is that if we only consider the frequency of usage for a particular context within the user's scope, then the accuracy of the recommendation results might be affected by the number of users who repeatedly use items in a large variety of contexts. Accordingly, we would neglect the importance of how many users have consumed items within that context as the opposite of a small number of users who consumed many items in a particular context. We apply column vector normalization to normalize the frequency values in a range between 0 and 1 as in (5):

$$uc(u_x, c_y) = \left(\frac{n_{uc}(u_x, c_y)}{\mathrm{N}_{\mathrm{u},c_y}}\right) \tag{5}$$

where $n_{u,c}(u_x, c_y)$ is the number of occurrences of context c_y in the list of consumed items by u_x and $N_{u,cy}$ represents the number of times the context c_y is used by all users, as in (6).

$$N_{u,c_y} = \sqrt{\sum_{x=1}^{|U|} (\delta_{x,y} f_{x,y})^2} \tag{6}$$

$$\delta_{x,y} = \begin{cases} 1, & c_y \text{ occurred in } u_x \\ 0, & \text{otherwise} \end{cases}$$

Therefore, using (4), we analyze how a particular context affected a user's item selection, where this context is similar to a given particular context. The more similar context c_y is to the user's context, the more influence it has on the user's preference value. Hence, the normalization step in the equation reduces the effect of contexts that were selected by many users to those that were selected by fewer.

3.1.2.2 Latent context-item preferences extraction

The latent preferences of contexts toward items capture how a particular context has occurred with the user's selection of items that are similar to a given particular item. This can be achieved by utilizing the normalized frequency matrix of ($\overline{\mathbf{CI}}$) and item-item similarity matrix **T** to form the new latent context-item matrix **cLi**. Formally, the matrix **cLi** represents the results of the product of the normalized frequency matrix of ($\overline{\mathbf{CI}}$) and the transpose of the similarity matrix $(\mathbf{T})^{\mathbf{T}}$, as in (7). We used the same normalization approach to construct ($\overline{\mathbf{CI}}$) as we did in constructing ($\overline{\mathbf{UC}}$).

$$\mathrm{cLi} = \overline{CI}(T)^T \tag{7}$$

3.1.2.3 Latent user-item preferences extraction

The latent user-item preference matrix constructed based on the assumption that users in a context consume certain items and they will likely consume items that are either similar to their preferences or similar to the choice of their nearest neighbors when they are in the same context in the future. Accordingly, the latent item preferences to a given user u_x can be obtained via dot product of the transpose of the original normalized rating matrix **UI** ($\overline{\mathbf{UI}}^{\mathbf{T}}$) and user-user similarity matrix **S** according to (8). Such product of the two matrices brings the user and their nearest neighbors' preferences to a given item.

$$\mathrm{uLi} = \overline{UI}^T(S)^T \tag{8}$$

As we observe in (8), we normalize the values in matrix **UI** as we did in normalizing matrices **CI** and **UC** because of the reason that some users are more active in rating and consuming different items than other inactive users. This leads to more contributions in the recommendation model from the active users compared to the less active users.

3.3 Optimization of the latent preference models

The key idea in this phase is that the latent preference matrices obtained in the previous section can be equivalently formulated as a stochastic optimization problem. Supposing we have the user-context preferences matrix **uLc** obtained by the dot product of the normalized user-context matrix **UC** and context-context similarity matrix **X** that cover a common latent feature space in which matrix **UC** spans the user space, while matrix **X** spans the context space. We can formulate this problem as an optimization problem in which we aim to minimize an objective function and find optimal **UC** and **X**. In particular, we aim to minimize the regularized squared error for known values in **UC** (the user-context matrix that decomposed to be used as a base for our recommendation model in section 3.1 and the corresponding predicted latent preference matrix **uLc**:

$$\min_{\overline{UC}^*, X^*} \sum_{(u,c)\in k} (\mathrm{UC} - \mathrm{uLc})^2 + \lambda(\|\overline{UC}\|^2 + \|X\|^2) \tag{9}$$

where, k is the set of user-context (u,c) pairs for which **UC** is known (can be obtained from the training set). The system learns the model by fitting the previously observed user-context value. However, the goal is to generalize those previous values in a way that predicts the latent preferences of users towards context. Thus, the system should avoid overfitting in the observed data by regularizing the learned parameters whose magnitudes are penalized. As shown in (9), the constant λ controls the extent of *regularization* and is usually determined by cross-validation and the regularization term $\lambda(\|\overline{UC}\|^2 + \|X\|^2$ is used to avoid overfitting. The most successful methods to solve this optimization problem are Alterative Least Squares (ALS) and Stochastic Gradient Descent (SGD) [30]. Koren et al. [30] argued that SGD based optimization generally performs better than ALS both in terms of model accuracy, run time performance and their flexibility nature for modeling various real-life situations.

The logic of the SGD approach can be expressed by solving (9) in such a way that the training dataset is obtained by partitioning the known user-context matrix **UC**. In addition, the initial **UC** and **X** matrices are given at first. Then the approach consists in looping through the training data and updating/recalculating **UC** and **X** after each training case till the error in prediction converges to a very small value, which implies that the predicted latent preference matrix ***uLc*** approached the known user-context matrix **UC** to the closest. This prediction error is calculated as,

$$e_{uc} = \mathrm{UC} - uLc \tag{10}$$

At each iteration, SGD sweeps over all known user-context value in **UC** and updates the corresponding rows $\overline{\mathbf{UC}}$ and **X**, correcting them in the inverse direction of the gradient of the error, by a factor of γ ≤ 1 – known as step size or learning rate yielding:

$$\begin{aligned} \mathrm{X} &\leftarrow \mathrm{X} + \gamma\,(e_{uc}\,.\overline{UC} - \lambda\,.\mathrm{X}) \\ \overline{UC} &\leftarrow \overline{UC} + \gamma\,(e_{uc}\,.\mathrm{X} - \lambda\,.\overline{UC}) \end{aligned} \tag{11}$$

where γ is the learning rate constant that controls the amount of update [30]. The algorithms either stops after a fixed number of iterations or as soon as no more improvement is observed. We follow the same optimization process for the rest of latent context-item and user-item preference matrices.

3.4 Latent collaborative model for rating prediction

The final phase of our model is a context-aware rating prediction for each user in which we applied the collaborative approach of the latent preference matrices obtained in the previous section. Accordingly, the final rating score of the items for each user can be computed by multiplying the latent user-context model and the latent context-item model. The association between these two models indicates that the final rating prediction can be determined by combining the contextual condition with users and items. Thus, the context-aware user-item rating score can be computed by:

$$Rating\ Score_{u,i} = \alpha optimized_{uLc_{u,c}} \times \beta optimized\_cLi_{c,i} \tag{12}$$

where $optimized\_uLc_{u,c}$ is the matrix that represents the value of the u-th user row and the c-th context column of the optimized uLc matrix. The $optimized\_cLi_{c,i}$ is the other matrix that represents the value of the c-th context row and the i-th item column in the optimized cLi matrix. The parameters α and β are used as shrinking factor in which they minimize the weight factor of less sensitive context. Their values are obtained through tuning and set after some experimental results. The final recommendation can be produced by sorting the rating predictions obtained based on (12) and then recommending top-k items with the highest predicted ratings.

4. Empirical Evaluations

In this section, the performance of the proposed recommendation model is evaluated. We start with a description of the dataset used for the evaluation before focusing on the experimental setup and the evaluation measures.

4.1 Experimental setup and dataset

As we mentioned earlier, the experimental data came from the collection of Last.fm dataset [27]. This dataset is a popular social music website which provides the users with online listening and tagging services. Through this platform, 30 million active users can describe their music tastes by tagging the music they have listened to. Last.fm data includes 522,366 unique tags and 505,216 tracks with at least one tag. We significantly reduce the tag set to the most frequent, general and descriptive tag categories. Hence, we selected the 200 top tags in the dataset which are relevant tag categories representing a unique musical genre, mood, or gender. In addition, we performed the data cleaning process since the dataset was initially too sparse. This data cleaning involves the removal of items that had been selected by a very small number of users, as well as by users who selected a small number of items. In addition, we cleaned items that had been tagged by all users with less than five contexts and removed tags that had been annotated by less than five users.

After accomplishing the data cleaning process, we applied the same offline experimental procedure suggested by the base paper [20] to measure the predictive accuracy of the context-aware recommendation model. Accordingly, we performed a random-based splitting procedure on the dataset by applying an 80:20 splitting ratio for the train and test sets, respectively. We ran our model five times with different splits to ensure that the evaluation is not vulnerable to the randomness of such division step.

4.2 Baseline recommender systems

We compare our proposed latent preference-based context-aware recommendation model to three state-of-the-art methods from the recommender systems literature: user-based collaborative filtering (UBCF), item-based CF (IBCF) and singular value decomposition (SVD)-based recommender systems.

User-based recommender systems predict a user rating on a new item by finding a *neighborhood* of similar users and aggregate their ratings. The neighborhood is defined in terms of similarity between users, either by taking a given number of most similar users (k nearest neighbors) or all users within a given similarity threshold [4]. Such similarity of taste between all users can be calculated using similarity measures such as cosine and Pearson correlation techniques from which we selected the cosine similarity function for our comparative analysis. Therefore, the *Cosine Vector* (*CV*) similarity between two user's u and v is computed as:

$$w_{uv} = CV(u,v) = \text{sim}_{\text{Cosine}}(r_u, r_v) = \frac{\sum_{i \subset I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \subset I_u} {r_{ui}}^2 \sum_{j \subset I_v} {r_{vj}}^2}} \tag{13}$$

where w_{uv} represent the correlation or similarity between ratings, r_{ui} denotes the rating given by user u to item i, r_{vi} denotes the rating given by user v to item i, I_u denotes the subset of items that have been rated by a user u, I_v denotes the subset of items that have been rated by a user v, I_{uv} denotes the subset of items that have been rated by both user u *and* v *and* r_{vi} denotes the rating given by user v to item j.

Once the users in the neighborhood are found, the general form of the prediction function is a weighted mean of the ratings of other users in the neighborhood [4]:

$$\hat{r}_{ui} = \bar{r_u} + \frac{\sum_{v \subset N_i(u)} w_{uv}(r_{vi} - \bar{r_v})}{\sum_{v \subset N_i(u)} |w_{uv}|} \tag{14}$$

where $N_i(u)$ is the *k-nearest-neighbors* of k users which are most similar to u that have rated item i. The fact that some users in the neighborhood are more similar to the active user is reflected into the weights in Equation (14).

On the other hand, the Item-based recommender strategy predict a rating by first looking at each item rated by the target user and find items similar to that item. Similar to the user-based strategy, the same procedure is followed to compute the similarity between items by applying the *Cosine similarity* function. After performing the similarity computation, the item-based strategy compute the final rating prediction as follows [15]:

$$\hat{r}_{ui} = \bar{r_i} + \frac{\sum_{j \subset \mathrm{N_u(i)}} w_{ij}(r_{uj} - \bar{r_j})}{\sum_{j \subset \mathrm{N_u(i)}} |w_{ij}|} \tag{15}$$

where $N_i(u)$ be the k items rated by user u most similar to item i and w_{ij} represent the correlation or similarity between two item's i and j.

Unlike the above CF-based approaches an SVD based recommender system works in such a way that the rating of user preference to items can be predicted by extracting the number of hidden features from the user-item matrix itself which contain all the ratings of users to the items they showed interest to. These hidden (latent) features are computed by factorizing the user-item rating matrix into a product of two lower rank matrices *U* and *V*, one containing the user factors and the other containing the item factors (*R* = *UV'*) where *R* is the user-item rating matrix. These two lower rank matrices can be approximated by minimizing the error to the known ratings using the well-known optimization technique called stochastic gradient descent (SGD) [30].

4.3 Evaluation measures

To evaluate the effectiveness of our proposed recommendation model, we adopted the commonly used error metric, namely root mean squared error (RMSE). It is the main metric that reveals the prediction error variance. That is, the lower the RMSE, the lower the error, the higher the performance. This metric can be defined in (16) as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{p}_i - p_i)^2} \tag{16}$$

where p_i is the observed value for the i^{th} observation, $\hat{p}_i$ is the predicted value and n the total length of the predicted data. The result can be positive or negative as the predicted value under or over estimates the actual value. Squaring the residuals, averaging the squares, and taking the square root gives us the RMSE error.

4.4 Experimental results

Based on the evaluation setup and experimental measures explained in the preceding section, we present the performance of the three latent preference models as well as the final latent collaborative rating prediction model in the following section.

4.4.1 Predictive performance of the latent preference models

This section presents the performance of our proposed latent preference models. We divide the experimental evaluation into two parts: the performance of the models before and after optimization. As shown in Table 1 and Fig. 2, prior to the optimization process, the predictive performance of our latent preference models (uLc, cLi, and uLi) obtains RMSE of 0.480, 0.159 and 0.341, respectively. The RMSE results of each model indicated that from all the latent preference models, the model that is used to predict the latent preferences of users towards items (uLi) achieved a substantial predictive performance over uLc and cLi models.

Table I: RMSE result obtained by the three latent preference models on the last.fm dataset

	uLc (Latent User-Context Preference Model)		***cLi (Latent Context-Item Preference Model)***		***uLi (Latent User-Item Preference Model)***	
	Before Optimi-zation	*After Optim-ization*	*Before Optim-ization*	*After Optim-ization*	*Before Optim-ization*	*After Optim-ization*
RMSE	0.480	0.316	0.159	0.102	0.341	0.244

To evaluate our models using an optimization algorithm, we first fine-tuned the necessary parameters in order to obtain optimal results on prediction performance of the model. These parameters include the regularization coefficient (λ), the learning rate (γ), and a number of features or dimensions (*k*). From all these three parameters, what is interesting is that any shrinking of the learning rate parameter can decrease the squared error which in turn improves the performance of the model, even if we set γ to zero. This means that γcan control the accuracy of the prediction performance of the model. Hence, according to our experiment, the optimum of γ is near 0.0002, at which the RMSE is the lowest. In such a way, the rest of involved parameters were carefully tuned, and the procedures to tune the parameters are equivalent to ensure reasonable comparison. In addition, the parameters with the best performance were used to report the final comparison results. After the parameter tuning process, we evaluate each model with the optimal parameter setting and recorded the value of RMSE along with the evaluation result of each model before the optimization process as shown in Table 1.

Fig. 2 showed the advantage of optimizing the latent preference models based on which the RMSE value can be minimized. Our context-aware latent preference models benefit from the optimization process in that even though the improvement is a slight increase in RMSE value, it is a great improvement from the recommendation system perspective. Accordingly, the change of prediction performance of our latent preference models (uLc, cLi, and uLi) after optimization can be manifested via RMSE result of 0.316, 0.102, and 0.244 respectively.

IJCSN - International Journal of Computer Science and Network, Volume 6, Issue 5, October 2017
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

4.4.2 The predictive accuracy of the final collaborative model and comparative performance

This is the section where we present the result of our evaluation of the predictive accuracy performance of our proposed latent collaborative rating prediction model as well as the performance of the baseline CF-based and SVD-based recommender systems for comparison. After the end of the experimentation, since the evaluation of a recommender system based on a test set is the final criterion of a solution, the performance of our latent collaborative rating prediction model were compared with the results of the baseline algorithms.

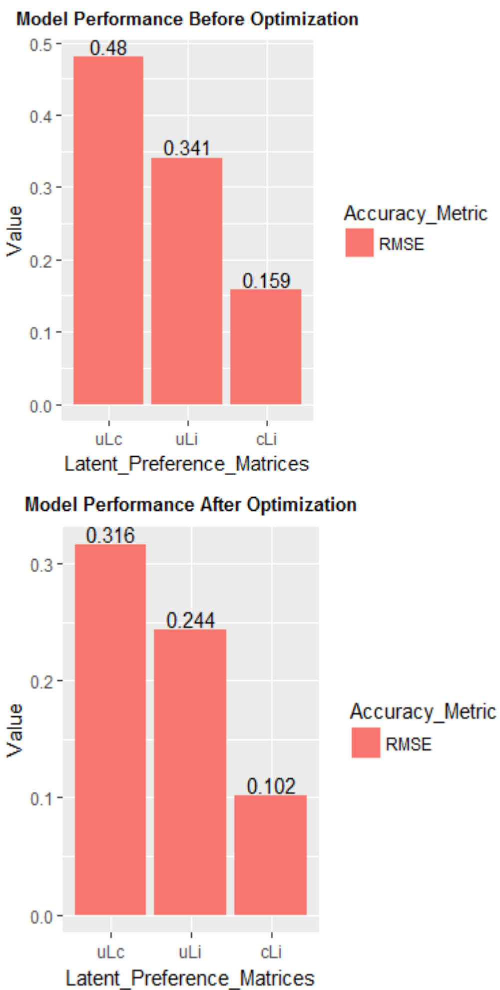

Fig. 2. The comparative predictive performance of the three latent preference models in terms of RMSE value before and after optimization.

Based on this notion, our context-aware latent collaborative model clearly outperforms all the other approaches. This is manifested in Table 2 and Fig. 3 that an SVD-based recommender strategy obtains an RMSE value of 0.283 as compared to 0.305 by user-based CF and 0.329 by item-based CF strategies respectively. However, our proposed latent preference model obtains an RMSE value of 0.214 in terms the rating prediction task which is better than the rest of the baseline recommender models and quite encouraging experimental result in terms of predictive performance. Improvement in RMSE value indicated a significant improvement in the quality of recommendation. Even a small minimization in RMSE value has an impact on the predictive quality of the algorithm. In this sense, our model showed a great achievement.

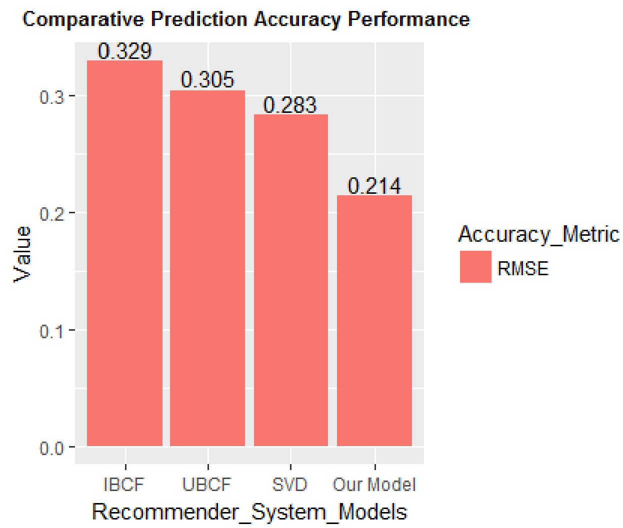

Fig. 3. Predictive Accuracy Performance Comparison

Table II. Evaluation of Rating Prediction

Recommender	***RMSE***
IBCF	0.329
UBCF	0.305
SVD	0.283
Our Latent Collaborative Model	0.214

5. Conclusions

In this paper, we demonstrated the potential of using user's contextual information and analyzing the use of a contextual tag to improve the prediction accuracy and enhance its quality. Through our experiment we showed how we explore the latent preferences based on the user, item, and context dimensions; i.e., latent user's preferences towards contexts, latent contexts preferences towards items, as well as latent user's preferences towards items. Additionally, we demonstrated a prediction performance improvement on latent preference models via optimization process. Based on our model, it is easy to identify the latent context information towards items, the latent context preferences from similar users and find the user-item preference by applying a collaborative filtering technique of those latent models. In general, the context-aware rating prediction and recommendation model that we proposed and demonstrated through experiment helps to improve the prediction accuracy and to provide relevant items based on user's context.

References

[1] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke, "Personalized recommendation in social tagging systems using hierarchical clustering", ACM Conference on Recommender Systems, 2008, pp. 259–266.

[2] Cosimo Palmisano, Alexander Tuzhilin, and Michele Gorgoglione, "Using context to improve predictive modeling of customers in personalization applications", IEEE Transactions on Knowledge and Data Engineering, vol. 11, 2008, pp.1535–1549.

[3] C. J. van Rijsbergen, Information Retrieval, 2nd ed. London, United Kingdom: Butterworths, 1979.

[4] Desrosiers C., and Karypis, A comprehensive survey of neighborhood-based recommendation methods, In Recommender systems handbook. Springer. pp. 107-144, 2011.

[5] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions", IEEE Knowledge and Data Engineering, vol.17, 2005, pp. 734 -749.

[6] G. Adomavicius and A. Tuzhilin, Context-Aware Recommender Systems, Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners, L. Rokach, B. Shapira, P. Kantor, and F. Ricci, Ed. Springer, 2011.

[7] G. Karypis, "Evaluation of item-based top-n recommendation algorithms", in Proc. 10th Int. Conf. Inf. Knowl. Manage., 2001, pp. 247–254.

[8] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering", Knowl.-Based Syst., vol. 56, 2014, pp. 156–166.

[9] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, and J. Tian, "Mining mobile user preferences for personalized context-aware recommendation", ACM Trans. Intell. Syst. Technol., vol. 5, no. 4, 2014, p. 58.

[10] I. Konstas, V. Stathopoulos, and J. M Jose, "On social networks and collaborative recommendation", ACM SIGIR Conference on Research and Development in Information Retrieval, 2009, pp. 195–202.

[11] J. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering", in Proc. 14th Ann. Conf. Uncertainty Artif. Intell., 1998, pp. 43–52.

[12] J. L. Herlocker, J. a. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems", ACM Trans. Inf. Syst., vol. 22, no. 1, Jan. 2004, pp. 5–53.

[13] J.-M. Chen, M.-C. Chen, and Y. S. Sun, "A tag based learning approach to knowledge acquisition for constructing prior knowledge and enhancing student reading comprehension", Comput. Education, vol. 70, 2014, pp. 256–268.

[14] J. S. Breese, D. Heckerman, and C. Kadie (1998), "Empirical analysis of predictive algorithms for collaborative filtering", In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (UAI'98), 2013, pp. 43-52.

[15] J. Peng, D.D. Zeng, H. Zhao, and F. Wang, "Collaborative filtering in social tagging systems based on joint item-tag recommendations", ACM International Conference on Information and Knowledge Management, 2010, pp. 809–818.

[16] J. Su, H. Yeh, P. Yu, and V. Tseng, "Music recommendation using content and context information mining", Intell. Syst., vol. 25, no. 1, 2010, pp. 16–26.

[17] K. H. L. Tso-Sutter, L. B. Marinho, and L. Schmidt-Thieme, "Tag-aware recommender systems by fusion of collaborative filtering algorithms", ACM SIGAPP Symposium on Applied computing, 2008, pp. 1995–1999.

[18] K. Yu, B. Zhang, H. Zhu, H. Cao, and J. Tian, "Towards personalized context-aware recommendation by mining context logs through topic models", in Proc. 16th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining, 2012, pp. 431–443.

[19] Kim, H.-N., Rawashdeh, M., Alghamdi, A., & El Saddik, A., "Folksonomy-based personalized search and ranking in social media services", Information Systems, vol. 37, no. 1, 2012, pp. 61–76.

[20] Mohammed F. Alhamid, Majdi Rawashdeh, Haiwei Dong, M. Anwar Hossain , Abdulmotaleb El Saddik, "Exploring latent preferences for context-aware personalized recommendation systems", IEEE Transactions on Human-Machine Systems, vol.46 n.4, , August 2016, pp. 615-623.

[21] M. F. Alhamid, M. Rawashdeh, H. Al Osman, M. S. Hossain, and A. El Saddik, "Towards context-sensitive collaborative media recommender system", Multimedia Tools Appl., vol. 74, 2015, pp. 11399–11428.

[22] M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms", ACM Trans. Inf. Syst., vol. 22, no. 1, 2004, pp. 143–177.

[23] M. Jamali and M. Ester, "TrustWalker: a random walk model for combining trust-based and item-based recommendation", ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 397–406.

[24] Q. Qi, Z. Chen, J. Liu, C. Hui, and Q. Wu, "Using inferred tag ratings to improve user-based collaborative filtering", ACM SIGAPP Symposium on Applied Computing, 2012, pp. 2008–2013.

[25] Shi, Y., Larson, M., & Hanjalic, A., "Mining contextual movie similarity with matrix factorization for context-aware recommendation", ACM Transactions on Intelligent Systems and Technology, vol. 4, no. 1, 2013, pp.1–19.

[26] Shin, D., Lee, J.-W., Yeon, J., & Lee, S.-G., "Context-aware recommendation by aggregating user context", In Proceedings of the IEEE conference on commerce and enterprise computing, 2009, pp. 423–430.

[27] T. Bertin-Mahieu, Daniel P.W. Ellis, B. Whitman, and P. Lamere, "The million song dataset", In Proceedings of the ISMIR, 2011.

[28] Thollot, R., "Dynamic situation monitoring and context-aware BI recommendations", Ph.D. dissertation, Ecole Centrale Paris, 2012.

[29] V. Zanardi and L. Capra, "Social ranking: uncovering relevant content using tag-based recommender systems", in Proc. ACM Conf. Recommender Syst., 2008, pp. 51–58.

[30] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems", IEEE Comput., vol. 42, no. 8, Aug. 2009, pp. 30-37.

Authors –

Solomon Demissie Seifu received a bachelor degree in the field of computer science and information technology with distinction in 2006 from Adama Science and Technology University (Ethiopia), then he hold a Masters degree in the field of Information Science in 2010 from Addis Ababa University (Ethiopia), and in 2015 he again received a Master of Technology (M.Tech) degree in Computer Science and Technology from Andhra University, India. He has published 2 papers in international Journals. He worked as Lecturer at the University of Debre Berhan (Ethiopia). His field of teaching includes programming languages like C, C++, Java etc, networking, Database and operating systems. His research interests include data mining and machine learning, recommender systems, cloud computing, and software engineering.

M. Shashi is a Professor and Chairperson of Board of Studies of the Department of Computer Science & Systems Engineering, A.U. College of Engineering(A), Andhra University, Visakhapatnam, Andhra Pradesh. She received the AICTE Career Award in 1996, Best Ph.D thesis prize from Andhra University in the year 1994 and AP State Best teacher award in 2016. 13 Ph.D.'s were awarded under her guidance. She co-authored more than 60 technical research papers in International Journals and 50 International Conferences and delivered many invited talks in such academic events. She is a member of IEEE Computational Intelligence group, Fellow of Institute of Engineers (India) and life member of Computer Society of India.. Her current research interests include Data warehousing and Mining, Data Analytics, Artificial Intelligence, Soft Computing and Machine Learning.