Proposing a Highly Accurate Hybrid Component-Based Factorised Preference Model in Recommender Systems

Farhad Zafari and **Rasoul Rahmani** and **Irene Moser**
Swinburne University of Technology, Melbourne, VIC 3122, Australia
{fzafari, rrahmani, imoser}@swin.edu.au

Abstract

Recommender systems play an important role in today's electronic markets due to the large benefits they bring by helping businesses understand their customers' needs and preferences. The major preference components modelled by current recommender systems include user and item biases, feature value preferences, conditional dependencies, temporal preference drifts, and social influence on preferences. In this paper, we introduce a new hybrid latent factor model that achieves great accuracy by integrating all these preference components in a unified model efficiently. The proposed model employs gradient descent to optimise the model parameters, and an evolutionary algorithm to optimise the hyper-parameters and gradient descent learning rates. Using two popular datasets, we investigate the interaction effects of the preference components with each other. We conclude that depending on the dataset, different interactions exist between the preference components. Therefore, understanding these interaction effects is crucial in designing an accurate preference model in every preference dataset and domain. Our results show that on both datasets, different combinations of components result in different accuracies of recommendation, suggesting that some parts of the model interact strongly. Moreover, these effects are highly dataset-dependent, suggesting the need for exploring these effects before choosing the appropriate combination of components.

1 Introduction

The overwhelming number of products (movies, books, music, news, services, etc.) offered by on-line retailers has made it difficult for the customers to decide which products to buy. This problem has incited retailers to invest on improving their recommender systems, and many e-commerce leaders such as Amazon and Netflix have made recommender systems a salient part of their websites [Koren *et al.*, 2009].

Recommender systems are usually based on collaborative filtering (CF) [Koren and Bell, 2011; Aldrich, 2011], where the preferences of a user are predicted by collecting rating information from other similar users or items [Ma *et al.*, 2008]. Among the CF systems, latent factor models have become popular mainly due to their high prediction accuracy and efficiency [Koren, 2008; Koren *et al.*, 2009]. These models explain the ratings by transforming both items and users on a shared latent feature space, which is inferred from the rating patterns [Zafari and Moser, 2016]. These models are very flexible and enable the incorporation of additional feedback information such as social network data, user and item biases, temporal information, contextual information, and user demographics. Many recent studies have contributed extensions to the basic PMF by incorporating additional information. For instance, SocialMF [Jamali and Ester, 2010], SoRec [Ma *et al.*, 2008], and SoReg [Ma *et al.*, 2011] extend PMF while addressing the problem of *social influence* in preferences and cold-start users, and SoCo [Zhao *et al.*, 2015] incorporates social and contextual information into the basic PMF. TimeSVD++ [Koren, 2010] extends the basic PMF (Probabilistic Matrix Factorisation) by incorporating the implicit feedback, *user and item biases*, as well as the drift of user preferences over *time*. Zhang et al. [Zhang *et al.*, 2014] use phrase-level sentiment analysis on user reviews to extract the users sentiments towards specific item *feature values* [Zafari and Nassiri-Mofakham, 2016b; 2017]. Liu *et al.* [Liu *et al.*, 2015] proposed an extension to their previously proposed latent factor model, ListPMF [Liu *et al.*, 2014] so that the *conditional dependencies* between *features* [Zafari *et al.*, 2015; Zafari and Nassiri-Mofakham, 2016a] are also taken into consideration. Therefore, we identify five major components to the preferences. These components include feature value preferences, social influence, temporal dynamics, conditional preferences, and user and item biases. Feature value preferences refer to the relative favourability of each one of the item feature values, social influence refers to the influence of social relationships on the preferences of a user, temporal dynamics means the drift of the preferences over time, conditional preferences refers to the dependencies between item features and their values, and user and item biases refer to the systematic tendencies for some users to give higher ratings than others, and for some items to receive higher ratings than others [Koren and Bell, 2011].

An important research question that arises here is how do the components of preferences interact with each other, and in particular, are these interactions dependent on the preference dataset or domain? Answering this question would enable the research community to design more accurate recommender systems and possibly tailor them to the specific needs of a domain. Although these components have been extensively researched individually, to the best of our knowledge, there is no model that integrates all these components into a unified model. The interaction effects of these components have been disregarded. This work is the first attempt at integrating all the components of preferences in a unified component-based model. In this paper, first we propose a hybrid model that incorporates all the components of preferences. Then we show that different interactions actually exist between preference components, which necessitates the designing of a unified component-based model integrating all the preference components in order to achieve the highest accuracy.

The rest of the paper is organised as follows: In section 2, we first briefly introduce matrix factorisation, which lays the foundation for most of the popular state of the art recommender systems, and then in section 2.1 we explain the proposed model TCSFVSVD, which is a latent factor model based on matrix factorisation that employs all of the five preference components. Then in section 2.2, we briefly explain the evolutionary algorithm that we used to optimise the hyperparameters and learning rates in TCSFVSVD, and then in section 2.3, we introduce our hybrid method to get optimised preference models for every combination of preference components. In section 3, we first explain the experimental setup, and then report on the results. We explain the related work in section 4, and finally we conclude the paper in section 5 and give the future directions.

2 Proposed Model

In rating-based recommender systems, the observed ratings are represented by the rating matrix R, in which the element R_{ij} is the rating given by the user i to the item j. Usually, R_{ij} is a 5-point integer, 1 point means very bad, and 5 points means excellent. Let $U \in \mathbb{R}^{N\times D}$ and $V \in \mathbb{R}^{M\times D}$ be latent user and item feature matrices, with vectors U_i and V_j representing user-specific and item-specific latent feature vectors respectively (N is the number of users, M is the number of items, and D is the number of item features). In probabilistic matrix factorization, the log-posterior over the user and item latent feature matrices with rating matrix and fixed parameters is minimised as shown in Eq. 1.

$$argmin_{U,V}[p(U,V|R,\sigma,\sigma_U,\sigma_V) = \ln p(R|U,V,\sigma)+\ln p(U|\sigma_U)+\ln p(V|\sigma_V)+C] \quad (1)$$

where C is a constant that is not dependent on U and V and σ, σ_U, and σ_V denote the standard deviations in the matrices R, U, and V respectively. Minimising Eq. 1 is equivalent to minimising Eq. 2.

$$argmin_{U,V}[E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{i,j}(R_{i,j}-\hat{R}_{i,j})^2 + \frac{\lambda_U}{2}\sum_{i=1}^{N}\|U_i\|^2_{Frob} + \frac{\lambda_V}{2}\sum_{j=1}^{M}\|V_j\|^2_{Frob}] \quad (2)$$

where $\|.\|_{Frob}$ denotes the Frobenius norm, and $\lambda_U = \frac{\sigma^2}{\sigma_U^2}$ and $\lambda_V = \frac{\sigma^2}{\sigma_V^2}$ are regularisation parameters (model hyperparameters), and $\hat{R}_{i,j} = U_i V_j^T$, and $I_{i,j}$ is the indicator function that is equal to 1 if user i rated item j, and 0 otherwise. *Stochastic Gradient Descent* and *Alternating Least Squares* are usually employed to solve the optimization problem in Eq. 2.

As can be seen in Eq. 2 in probabilistic matrix factorization, R_{ij} is estimated by the inner product of latent user feature vector U_i and latent item feature vector V_j, that is $\hat{R}_{i,j} = U_i V_j^T$. In other words, the goal of matrix factorization is to factorize a matrix into two matrices such that by multiplying the factorized matrices, the original matrix can be approximated.

In basic matrix factorisation, the preferences of a user are only defined as a user feature vector, which represents the importance that the user gives to each item feature (e.g. price is extremely important while quality is less important). However, one of the important properties of probabilistic matrix factorisation is that it enables the incorporation of additional information such as user and item biases, temporal information, and social influence.

2.1 TCSFVSVD

In this section, we introduce an extension to the basic matrix factorisation, which incorporates all the five preference components mentioned earlier. This model is abbreviated to TCS-FVSVD (**T**ime **C**onditional **S**ocial **F**eature **V**alue **S**ingular **V**alue **D**ecomposition). In Fig. 1, **FP** represents preferences over features, which is captured by matrix U in the basic matrix factorisation. **B** represents user and item bias, **F** represents item features captured by matrix V in the basic matrix factorisation. **CP** represents conditional dependencies, **FVP** represents preferences over feature values, **SI** stands for social influence, and finally **T** is an abbreviation for time.

TCSFVSVD incorporates additional matrices and vectors into matrix factorisation, so that all these components can be learnt from the users' ratings and social connections. As this figure shows, the model starts by loading the time-stamped user ratings as well as the social network data into the memory. The main loop accounts for the learning iterations over the model. The first loop within the main loop iterates over the time-stamped user-item ratings matrix, while the second loop iterates over the social network adjacency matrix, to train the socially influenced parts of the model. In each loop, one entry of the input matrix is read and used to update the matrices/vectors related to that input data. As can be seen, the user and item bias values are only updated in loop 1, since they are only related to the user ratings. Both user-item ratings and users' social relationships include information about the users' preferences over features. Therefore, the new values for FP are calculated in both loops and updated in the main loop, when all new values have been calculated. Similarly, the values for SI and FVP depend on both user-item ratings and social relationships. Consequently, their new values are calculated inside both loops 1 and 2, and are updated in the main loop. In contrast, the values of F as well as CP only

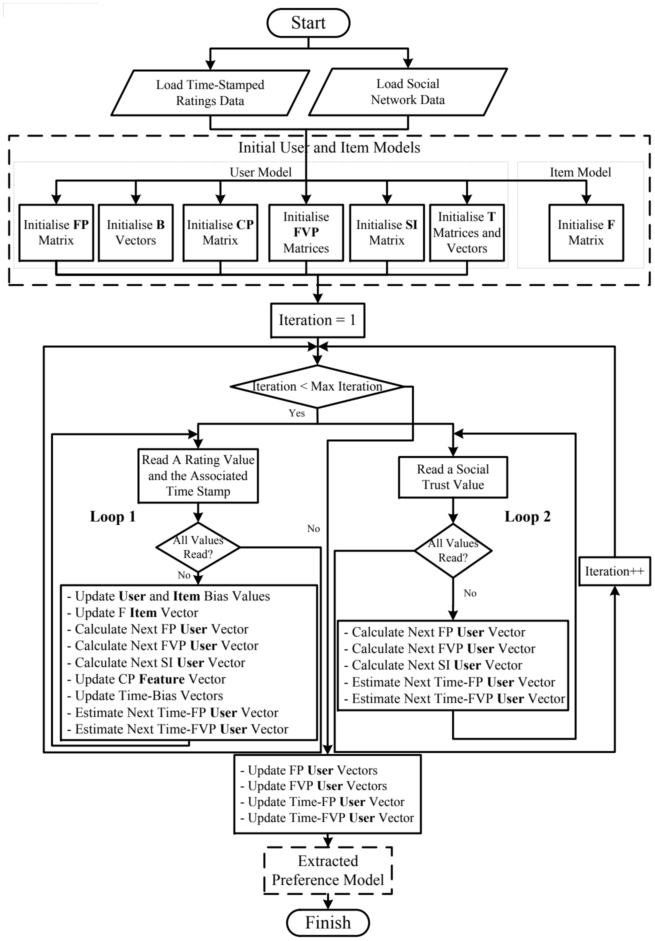


Figure 1: The flow charts of TCSFVSVD

need the user-item ratings to be updated. Therefore, they are immediately updated inside the loop 1. The time component includes parameters that account for the dynamics of user and item biases, feature value preferences, and preferences over features. Since Bias values do not depend on the user-item ratings matrix, they are updated immediately in loop 1. However, the new values for the dynamics of feature value preferences, and preferences over features are updated in the main loop. In the proposed method, every one of the preference components can be arbitrarily switched off and on by setting their learning rates to zero or a non-zero value respectively.

Although social relationships are likely to be time dependent, most data sets do not contain this information. Conditional preferences are related to the feature value preferences, since they model the dependencies between the features and their values, and therefore, are applied to the matrices that account for the users' preferences over feature values. Social influence is applied to the components of preferences over features and preferences over feature values. However, applying social influence to the user and item biases has no observable benefits and user or item biases do not seem to be influenced by social interactions. Therefore, we concluded that user and item biases are not much influenced by the social interactions. One of the problems faced in probabilistic matrix factorisation is that if the regularisation parameters are not tuned carefully, the model is prone to over-fitting because it finds a single point estimate of the parameters [Salakhutdinov and Mnih, 2008]. Therefore, in basic matrix factorisation, it is assumed that the regularisation parameters are known in advance, and they are fed into the model as model inputs. In order to address the problem of finding the optimal parameters, [Salakhutdinov and Mnih, 2008] proposed an extension to the basic PMF by assuming Gaussian-Wishart priors on the user and item regularisation parameters. In their proposed method, these parameters are also learnt along with other model parameters. However, the proposed approach increases the computational complexity. Furthermore, our experiments show that this method provides less accurate recommendations than some other methods. We found that the performance of the TCSFVSVD depends to a large extent on fine-tuning the hyper-parameters and learning rates for each of the preference components. To optimise the hyper-parameters and learning rates, the proposed method employs a hybrid method combining an Evolutionary Algorithm (EA) and Gradient Descent (GD). In the following sections, we first briefly introduce Evolutionary Algorithm (EA), and then explain the proposed hybrid method. This method is abbreviated to ATCSFVSVD (Adaptive TCSFVSVD).

2.2 EA

To optimise the learning rate and regularisation parameter of each model component, we applied an evolutionary algorithm (EA) [Aleti and Moser, 2016] with real encoding, Gaussian mutation and single-point crossover. The parent population was chosen using binary tournament selection. The algorithm receives its feedback from the error value resulting from the designed matrix factorisation model. The algorithm begins with the initialisation of an initial current population (P). Then three empty sets of solutions OP, PP, and CP are created. Then M solutions from the set P are selected according to the tournament selection method, and added to the parents set (PP). The parent solutions in PP are evolved and added to CP, and then all the solutions in CP are added to the offspring set (OP), and this process is repeated until the size of this set exceeds the maximum number of offspring. Then the solutions in OP are evaluated and all the solutions in P are also added to OP. Then the solutions in OP are truncated according to a truncation strategy, and all the solutions in P are replaced by the truncated solutions in OP. Now P includes the next generation of the solutions. This process is repeated until a stopping criteria is met, and finally the best solution in P is chosen as the optimal solution and returned by the algorithm. This best solution includes the optimal hyper-parameters and learning rates for the TCSFVSVD. Optimising the matrices repeatedly with hyper-parameters (regularisation factors) and learning rates produced by the EA is computationally expensive. For the hyper-parameter optimisation, a representative subset of 10%, 3%, and 1% of the matrix entries depending on the dataset were used, sampled uniformly randomly. The interactions between the EA and the matrix optimisation are illustrated in Fig. 2.

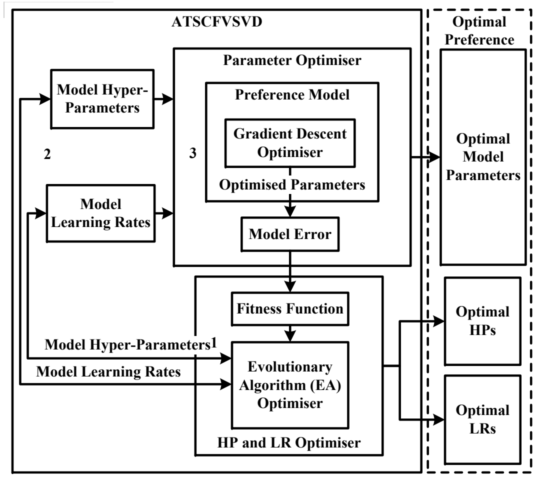


Figure 2: The schematic representation of ATCSFVSVD

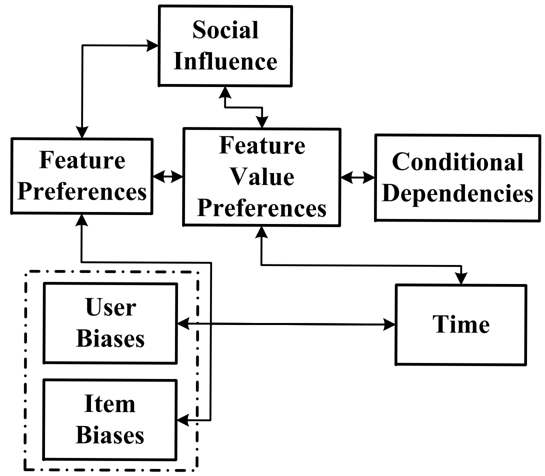


Figure 3: the preference components and their interplay in ATCS-FVSVD

2.3 ATCSFVSVD

Having obtained the near-optimal values for the learning rates and regularisation parameters of the model using a subset of the matrix entries, we can optimise the matrices by applying gradient descent repeatedly on the complete datasets using the pre-optimised hyper-parameters. In recommender systems, there are usually thousands of users who show their preferences over thousands of items. Therefore, the constructed model includes millions of parameters to optimise. Most of the latent factor models that we have reviewed so far employ gradient descent to optimise the model parameters. This method is applicable to the problems in which the solution space is differentiable, and particularly suitable to the problems that include many parameters to optimise. The popularity of this method stems from its efficiency and the high quality solutions that it can find at a reasonable time. In previous work [Zafari and Moser, 2017], it has been observed that recommender systems achieve their best accuracies when different components, represented by matrices, capture different aspects of a dataset. It is intuitively clear that the components of the model interact; for example, we can assume that users' preferences change over time. To investigate the interdependencies between the components, we switched off components one at a time and investigated the effects on the model in an attempt to isolate the contribution of each component.

3 Experiments

In order to evaluate the proposed model and determine the interactions of the preference components, we train separate models for different component combinations. As explained in section 2.1, preferences are comprised of 5 major components. Considering different combinations of components, 32 preference models are possible. In order to analyse the interactions of the components, we first need to obtain the optimal hyper-parameters and learning rates using the ATCS-FVSVD method as in Fig. 2. This gives us 32 sets of hyper-parameters and learning rates, each for one combination of components. Then we run TCSFVSVD using the optimal hyper-parameters and learning rates obtained using ATCS-FVSVD and evaluate the error of each combination of components. The results are compared with TrustSVD [Guo *et al.*, 2016], which has the highest accuracy among a large set of state of the art models.

3.1 Experimental Setting

We tested the proposed method on three popular datasets, Ciao, Epinions, and Flixster. Ciao is a dataset crawled from the ciao.co.uk website. This dataset includes 35,835 ratings given by 2,248 users over 16,861 movies. Ciao also includes the trust relationships between users. The number of trust relationships in Ciao is 57,544. Therefore the dataset density of ratings and trust relationships are 0.09% and 1.14% respectively. The ratings are integer values between 1 and 6. The Epinions dataset consists of 664,824 ratings from 40,163 users on 139,738 items of different types (software, music, television show, hardware, office appliances, ...). Ratings are integer values between 1 and 5, and data density is 0.011%. Epinions also enables the users to issue explicit trust statements about other users. This dataset includes 487183 trust ratings. The density of the trust network is 0.03%. Flixster is a social movie site which allows users to rate movies and share the ratings with each other, and become friends with others with similar movie taste. Flixster dataset which is collected from Flixster website includes 8,196,077 ratings issued by 147,612 users on 48,794 movies. The social network also includes 7,058,819 friendship links. The density of the ratings matrix and social network matrix are 0.11% and 0.001% respectively. In order to reduce the computational expenses, we applied the algorithm to a uniformly randomly selected subset of the users and items. These reduced matrices retain the original density and are a representative sample. In all the experiments, 80% of the datasets are used for training and the remaining 20% are used for evaluation. Each model training is repeated for 30 times and the average values are used.

3.2 Results

To analyse the interaction of the components, as mentioned before, we first optimise the hyper-parameters and learning rates for the 32 models that account for 32 combinations of the 5 components. Then we use the optimised hyper-parameters for each model combination to train the model using TCSFVSVD and evaluate the error on the test set. The MAE values and RMSE values obtained on the Ciao, Epinions, and Flixster datasets have been compared with those of TrustSVD in the tables 1, 2, and 3 respectively. We also applied t test to determine whether the improvements are statistically significant.

As we see, in terms of MAE, the proposed method achieves significantly better results than TrustSVD, in all the combinations that include B (Bias) on the Ciao and Flixster datasets. However, on the Epinions dataset, almost all the combinations that exclude B achieve significantly better results than TrustSVD. This suggests that B has rather deteriorating effect on the Epinions dataset. In terms of MAE, the model that includes all the components achieves the lowest error on the Ciao dataset. However on the Epinions dataset, the model that only includes FVP achieves the lowest error. On the Flixster dataset, the model including the FVP, SI, and B components gets the highest accuracy. In terms of RMSE, the model that only includes CP, T, and B achieves the highest accuracy on the Ciao and Flixster datasets, and the model that includes FVP and SI performs the best on the Epinions dataset.

Furthermore, by considering the interaction plots for every pair of variables, we notice that there are two-way interactions between FVP and SI, SI and T on the Ciao dataset, between FVP and T and CP and B in the Epinions dataset, and between FVP and B, and FVP and T on the Flixster dataset. We also illustrate the interaction plots for the major two-way interactions observed in the three datasets. The non-parallel lines in these figures graphically demonstrate the existing interactions. These figures show the average RMSE of the model for the cases where the model includes or excludes the two components involved in the interaction. From Fig. 4a, we notice that the addition of FVP improves the accuracy regardless of whether the model includes SI or not. We also notice that when SI is helpful at the presence of FVP. When FVP is not present, the exclusion of SI gets a better accuracy. The same kind of interaction in Fig. 5a is observed between FVP and T on the Epinions dataset. Fig. 4b however shows a different kind of interaction between SI and T. We can see that when T is present, the inclusion of SI deteriorates the accuracy, while in case T is not present, the inclusion of SI improves the accuracy. Therefore, the helpfulness of SI actually depends on whether the component T is present or not. In Fig. 5b, we notice that when B is present, the addition of CP improves the accuracy. However, when B is switched off, the inclusion of CP only worsens the accuracy. This means that CP is only helpful when B is also modelled. We can also see that in general, regardless of whether B is switched on or not, the model is better off without any of the components B and CP. Fig. 6a, shows that the inclusion of B regardless of FVP improves the accuracy of the model. This figure also shows that addition of FVP is helpful when B is switched off. However, when B is switched on, adding FVP slightly worsens the accuracy of the model. Fig. 6b illustrates a different type of interaction, where adding FVP always improves the accuracy. This figure also shows that the inclusion of T worsens the accuracy, regardless of whether FVP is switched on or off. It also reveals that the exclusion of T results in a higher accuracy improvement in case FVP is switched on.

From the results, we can conclude that different component combinations should be employed in different datasets. As we observed, various types of interactions can exist between preference components in different datasets. Therefore, the interaction effects of the components should be taken into consideration in order to achieve the best performance.

4 Related Work

Recommender systems can be broadly classified into content-based and collaborative filtering (CF) systems. Content-based filtering, also referred to as cognitive filtering, originates in information retrieval and text processing, and recommends items based on a comparison between the content of the items and a user profile. CF approaches on the other hand predict the preferences of a user by collecting preference information from many users. These methods are widely adopted to build recommender systems and can be broadly classified into memory-based and model-based approaches. Model-based CF learns the parameters of a model and only stores those parameters. Algorithms in the category of model-based CF include the clustering, aspect and latent factor models [Ma *et al.*, 2008; Jiang *et al.*, 2012; Aghdam *et al.*, 2015].

Latent factor models as an example of model-based collaborative filtering explain the ratings by characterising both users and items on a number of latent factors which are inferred from the rating patterns. Recently, latent factor models based on matrix factorisation have gained much popularity as they usually outperform traditional memory-based methods, and have achieved higher performance in some benchmark datasets [Koren *et al.*, 2009].

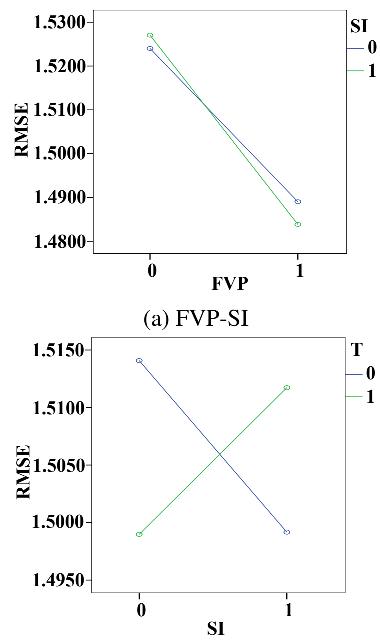
Incorporating user and item bias values into latent factor models have been proposed by Koren [2009; 2011]. According to Koren, user and item biases tend to capture much of the observed signal in user-item ratings data. He proposed a model called SVD++, that incorporated the implicit feedback from the user-item ratings as well as user and item biases into the probabilistic matrix factorisation.

There are only a few recommender systems that consider conditional preferences. Liu *et al.* [2015] proposed a latent factor model that does not require domain knowledge, and rather directly captures the conditional preferences from the user ratings. Using the movielens dataset, they showed that most of the users' preferences in rating-based recommender systems are conditional. Then they proved that quadratic polynomial can model the conditional preferences that cannot be captured by the linear function used in conventional latent factor models based on matrix factorisation. They showed how to integrate the proposed quadratic approximation model of conditional preferences into ListPMF [Liu *et al.*, 2014] in order to obtain more accurate results.

Zhang *et al.* [2014] proposed a model to capture the users' preferences over feature values from user reviews. They used phrase-level sentiment analysis on user reviews to extract explicit item features and user opinions (sentiments) and pro-

	MAE					RMSE				
Components	**Model**	**TrustSVD**	**t value**	**p value**	**Sig.**	**Model**	**TrustSVD**	**t value**	**p value**	**Sig.**
None	1.4275	1.1508	24.6290	0.0000	yes	1.6772	1.3969	25.1063	0.0000	yes
B	1.1032	1.1508	-7.7055	0.0000	yes	1.3725	1.3969	-3.6063	0.0006	yes
T	1.4252	1.1508	22.8287	0.0000	yes	1.6774	1.3969	25.2688	0.0000	yes
T B	1.1086	1.1508	-6.5056	0.0000	yes	1.3733	1.3969	-3.5415	0.0008	yes
SI	1.4152	1.1508	23.4468	0.0000	yes	1.6698	1.3969	22.7944	0.0000	yes
SI B	1.1116	1.1508	-4.8783	0.0000	yes	1.3731	1.3969	-3.0958	0.0031	yes
SI T	1.4225	1.1508	23.8483	0.0000	yes	1.6700	1.3969	24.5632	0.0000	yes
SI T B	1.1215	1.1508	-4.4167	0.0000	yes	1.3815	1.3969	-2.3212	0.0238	yes
CP	1.3965	1.1508	21.7273	0.0000	yes	1.6506	1.3969	25.6979	0.0000	yes
CP B	1.1362	1.1508	-2.1803	0.0333	yes	1.4458	1.3969	7.8582	0.0000	yes
CP T	1.3753	1.1508	20.3797	0.0000	yes	1.6239	1.3969	20.9376	0.0000	yes
CP T B	1.1079	1.1508	-6.0150	0.0000	yes	**1.3714**	1.3969	-3.7379	0.0004	yes
CP SI	1.3868	1.1508	21.2223	0.0000	yes	1.6279	1.3969	20.9090	0.0000	yes
CP SI B	1.1234	1.1508	-4.0826	0.0001	yes	1.3843	1.3969	-1.8253	0.0731	no
CP SI T	1.3560	1.1508	16.7996	0.0000	yes	1.6101	1.3969	18.5375	0.0000	yes
CP SI T B	1.2487	1.1508	12.1932	0.0000	yes	1.4996	1.3969	13.9836	0.0000	yes
FVP	1.3702	1.1508	19.4127	0.0000	yes	1.6204	1.3969	20.2715	0.0000	yes
FVP B	1.1154	1.1508	-4.8547	0.0000	yes	1.3781	1.3969	-2.6843	0.0095	yes
FVP T	1.3525	1.1508	19.1287	0.0000	yes	1.6006	1.3969	19.9911	0.0000	yes
FVP T B	1.1223	1.1508	-4.0554	0.0002	yes	1.3833	1.3969	-1.8287	0.0727	no
FVP SI	1.3630	1.1508	19.7696	0.0000	yes	1.6118	1.3969	18.7106	0.0000	yes
FVP SI B	1.1194	1.1508	-3.8211	0.0004	yes	1.3826	1.3969	-2.0094	0.0492	yes
FVP SI T	1.3368	1.1508	15.2481	0.0000	yes	1.6051	1.3969	16.3300	0.0000	yes
FVP SI T B	1.1118	1.1508	-5.7295	0.0000	yes	1.3746	1.3969	-3.4459	0.0011	yes
FVP CP	1.3475	1.1508	17.0897	0.0000	yes	1.5907	1.3969	17.0615	0.0000	yes
FVP CP B	1.1147	1.1508	-5.3312	0.0000	yes	1.3773	1.3969	-2.9649	0.0044	yes
FVP CP T	1.3386	1.1508	14.4214	0.0000	yes	1.5803	1.3969	14.5007	0.0000	yes
FVP CP T B	1.1217	1.1508	-4.0294	0.0002	yes	1.3816	1.3969	-2.1205	0.0383	yes
FVP CP SI	1.3025	1.1508	13.5713	0.0000	yes	1.5670	1.3969	15.6979	0.0000	yes
FVP CP SI B	1.1086	1.1508	-4.9167	0.0000	yes	1.3768	1.3969	-2.4427	0.0180	yes
FVP CP SI T	1.3081	1.1508	13.2102	0.0000	yes	1.5790	1.3969	16.6703	0.0000	yes
FVP CP SI T B	**1.1021**	1.1508	-6.0648	0.0000	yes	1.3738	1.3969	-3.1400	0.0027	yes

Table 1: The results of t-test for each model combination against TrustSVD for the Ciao dataset



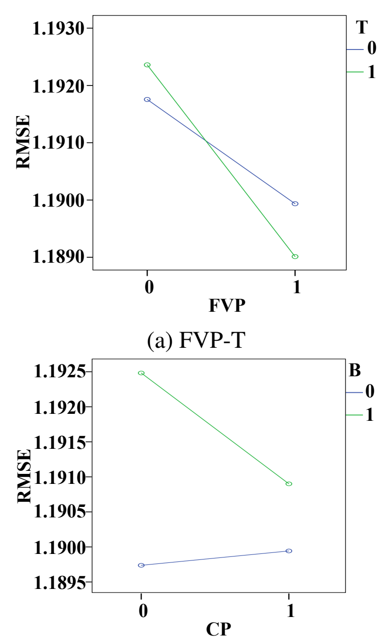
(a) FVP-SI

(b) SI-T

Figure 4: Two-way interaction effects on the Ciao dataset

	MAE					RMSE				
Components	**Model**	**TrustSVD**	**t value**	**p value**	**Sig.**	**Model**	**TrustSVD**	**t value**	**p value**	**Sig.**
None	0.8919	0.9242	-14.7896	0.0000	yes	1.1912	1.1912	-0.0366	0.9710	no
B	0.9213	0.9242	-1.2188	0.2279	no	1.1924	1.1912	0.5134	0.6096	no
T	0.8903	0.9242	-16.4499	0.0000	yes	1.1918	1.1912	0.3185	0.7515	no
T B	0.9219	0.9242	-0.8563	0.3956	no	1.1929	1.1912	0.6549	0.5152	no
SI	0.8923	0.9242	-13.9091	0.0000	yes	1.1921	1.1912	0.3913	0.6970	no
SI B	0.9219	0.9242	-0.8809	0.3821	no	1.1924	1.1912	0.4819	0.6317	no
SI T	0.8932	0.9242	-16.4361	0.0000	yes	1.1930	1.1912	0.9123	0.3660	no
SI T B	0.9231	0.9242	-0.5085	0.6130	no	1.1947	1.1912	1.4135	0.1629	no
CP	0.8923	0.9242	-15.4226	0.0000	yes	1.1897	1.1912	-0.7630	0.4487	no
CP B	0.9196	0.9242	-1.9628	0.0545	no	1.1898	1.1912	-0.5410	0.5907	no
CP T	0.8926	0.9242	-16.1177	0.0000	yes	1.1921	1.1912	0.4440	0.6589	no
CP T B	0.9193	0.9242	-1.9537	0.0557	no	1.1900	1.1912	-0.5330	0.5961	no
CP SI	0.8938	0.9242	-16.4465	0.0000	yes	1.1918	1.1912	0.2864	0.7756	no
CP SI B	0.9226	0.9242	-0.6287	0.5321	no	1.1946	1.1912	1.3634	0.1781	no
CP SI T	0.8937	0.9242	-13.7415	0.0000	yes	1.1918	1.1912	0.2725	0.7864	no
CP SI T B	0.9224	0.9242	-0.7115	0.4798	no	1.1926	1.1912	0.4988	0.6199	no
FVP	**0.8895**	0.9242	-16.1290	0.0000	yes	1.1883	1.1912	-1.4123	0.1634	no
FVP B	0.9226	0.9242	-0.6281	0.5325	no	1.1925	1.1912	0.5279	0.5996	no
FVP T	0.8905	0.9242	-15.9989	0.0000	yes	1.1862	1.1912	-2.0161	0.0485	yes
FVP T B	0.9182	0.9242	-2.4315	0.0182	yes	1.1896	1.1912	-0.6164	0.5402	no
FVP SI	0.8912	0.9242	-15.2663	0.0000	yes	**1.1856**	1.1912	-2.1763	0.0338	yes
FVP SI B	0.9214	0.9242	-1.0594	0.2940	no	1.1936	1.1912	0.9937	0.3245	no
FVP SI T	0.8922	0.9242	-14.9977	0.0000	yes	1.1897	1.1912	-0.6744	0.5027	no
FVP SI T B	0.9201	0.9242	-1.7241	0.0900	no	1.1918	1.1912	0.2361	0.8142	no
FVP CP	0.8947	0.9242	-16.1379	0.0000	yes	1.1927	1.1912	0.6804	0.4990	no
FVP CP B	0.9180	0.9242	-2.9121	0.0052	yes	1.1896	1.1912	-0.7657	0.4471	no
FVP CP T	0.8904	0.9242	-16.0818	0.0000	yes	1.1857	1.1912	-2.2216	0.0303	yes
FVP CP T B	0.9207	0.9242	-1.4380	0.1558	no	1.1932	1.1912	0.8964	0.3738	no
FVP CP SI	0.8933	0.9242	-16.9729	0.0000	yes	1.1880	1.1912	-1.4328	0.1573	no
FVP CP SI B	0.9190	0.9242	-2.0765	0.0424	yes	1.1893	1.1912	-0.8747	0.3854	no
FVP CP SI T	0.8931	0.9242	-15.8001	0.0000	yes	1.1878	1.1912	-1.6135	0.1122	no
FVP CP SI T B	0.9192	0.9242	-1.9090	0.0614	no	1.1881	1.1912	-1.3097	0.1955	no

Table 2: The results of t-test for each model combination against TrustSVD for the Epinions dataset



(a) FVP-T

(b) CP-B

Figure 5: Two-way interaction effects on the Epinions dataset

	MAE					RMSE				
Components	**Model**	**TrustSVD**	**t value**	**p value**	**Sig.**	**Model**	**TrustSVD**	**t value**	**p value**	**Sig.**
None	0.8425	0.7910	19.9469	0.0000	yes	1.0687	0.9932	27.2935	0.0000	yes
B	0.7802	0.7910	-5.0999	0.0000	**yes**	0.9947	0.9932	0.8913	0.3767	no
T	0.8426	0.7910	17.5169	0.0000	yes	1.0729	0.9932	39.9944	0.0000	yes
T B	0.7812	0.7910	-4.1499	0.0001	**yes**	0.9933	0.9932	0.0095	0.9925	no
SI	0.8441	0.7910	25.2272	0.0000	yes	1.0722	0.9932	42.7515	0.0000	yes
SI B	0.7774	0.7910	-5.5178	0.0000	**yes**	0.9926	0.9932	-0.3129	0.7555	no
SI T	0.8413	0.7910	20.2648	0.0000	yes	1.0702	0.9932	29.1248	0.0000	yes
SI T B	0.7823	0.7910	-3.6435	0.0006	**yes**	0.9940	0.9932	0.4329	0.6667	no
CP	0.8457	0.7910	26.1998	0.0000	yes	1.0698	0.9932	34.8148	0.0000	yes
CP B	0.7807	0.7910	-4.0814	0.0002	**yes**	0.9917	0.9932	-0.8123	0.4200	no
CP T	0.8448	0.7910	22.4132	0.0000	yes	1.0701	0.9932	33.1685	0.0000	yes
CP T B	0.7784	0.7910	-5.1980	0.0000	**yes**	**<u>0.9895</u>**	0.9932	-1.6860	0.0978	no
CP SI	0.8463	0.7910	23.9656	0.0000	yes	1.0705	0.9932	28.5411	0.0000	yes
CP SI B	0.7799	0.7910	-5.3745	0.0000	**yes**	0.9922	0.9932	-0.5437	0.5888	no
CP SI T	0.8504	0.7910	26.1554	0.0000	yes	1.0706	0.9932	25.3960	0.0000	yes
CP SI T B	0.7814	0.7910	-4.5509	0.0000	**yes**	0.9931	0.9932	-0.0912	0.9277	no
FVP	0.8141	0.7910	7.5863	0.0000	yes	1.0501	0.9932	18.8726	0.0000	yes
FVP B	0.7774	0.7910	-6.6830	0.0000	**yes**	0.9944	0.9932	0.5983	0.5521	no
FVP T	0.8136	0.7910	6.4479	0.0000	yes	1.0480	0.9932	17.3676	0.0000	yes
FVP T B	0.7776	0.7910	-6.4858	0.0000	**yes**	0.9948	0.9932	0.8192	0.4161	no
FVP SI	0.8143	0.7910	6.6098	0.0000	yes	1.0516	0.9932	18.7214	0.0000	yes
FVP SI B	**<u>0.7733</u>**	0.7910	-7.6412	0.0000	**yes**	0.9945	0.9932	0.6670	0.5075	no
FVP SI T	0.8201	0.7910	8.2561	0.0000	yes	1.0586	0.9932	18.9316	0.0000	yes
FVP SI T B	0.7754	0.7910	-6.1368	0.0000	**yes**	0.9953	0.9932	1.1278	0.2641	no
FVP CP	0.8125	0.7910	6.7558	0.0000	yes	1.0466	0.9932	14.7821	0.0000	yes
FVP CP B	0.7803	0.7910	-5.1025	0.0000	**yes**	0.9944	0.9932	0.6886	0.4938	no
FVP CP T	0.8169	0.7910	7.3838	0.0000	yes	1.0514	0.9932	17.6998	0.0000	yes
FVP CP T B	0.7809	0.7910	-4.8757	0.0000	**yes**	0.9936	0.9932	0.2039	0.8392	no
FVP CP SI	0.8218	0.7910	8.3818	0.0000	yes	1.0557	0.9932	18.7621	0.0000	yes
FVP CP SI B	0.7779	0.7910	-5.2527	0.0000	**yes**	0.9932	0.9932	-0.0033	0.9974	no
FVP CP SI T	0.8183	0.7910	8.1475	0.0000	yes	1.0553	0.9932	19.4216	0.0000	yes
FVP CP SI T B	0.7762	0.7910	-5.8009	0.0000	**yes**	0.9964	0.9932	1.6012	0.1150	no

Table 3: The results of t-test for each model combination against TrustSVD for the Flixster dataset

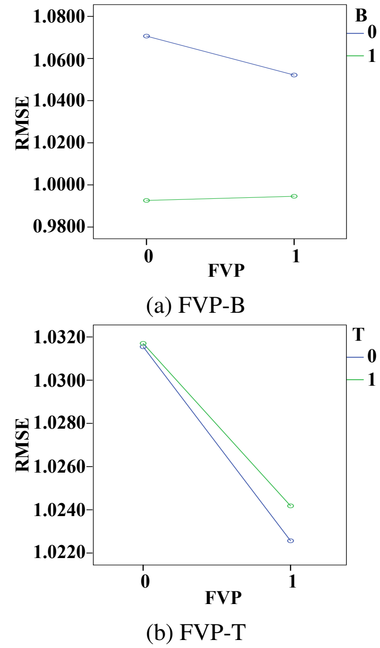


Figure 6: Two-way interaction effects on the Flixster dataset

posed EFM. The extracted explicit preferences were incorporated into basic probabilistic matrix factorisation in order to improve the recommendation accuracy and explainability. The advantage of this method over other methods based on basic MF is that it takes the preferences over feature values into account, which improves the accuracy. Its disadvantage is that it operates based on user reviews and in cases when this information is not available, it can not be applied. Furthermore, it is based on phrase-level sentiment analysis, which ignores the context of the sentence in which a term is used.

To incorporate the influence of social friends in preferences, [Guo *et al.*, 2013; 2016] have proposed TrustSVD. This method incorporates the explicit and implicit influences of ratings as well as trust between users in a social network into matrix factorisation. Comprehensive experimental results have shown that TrustSVD outperforms both trust- and rating-based methods in predictive accuracy. However, TrustSVD does not include conditional preferences over feature values and it ignores the dynamicity of preferences.

To capture the dynamic preferences, Koren proposed TimeSVD++ [Koren, 2010; Koren and Bell, 2011]. This model builds on SVD++, which extends the basic matrix factorisation by adding user and item biases and implicit feedback. In TimeSVD++, additional matrices and vectors are added to the model, so that the dynamicity of user and item biases and preferences over features are also modelled.

To the best of our knowledge, there is currently no model that integrates all the aforementioned preference components into a single model. Our work is the first attempt in modelling all the preference components in a single recommender. This enables us to design a model that can achieve the highest

5 Conclusion and Future Work

In this paper, we proposed a novel hybrid latent factor model, incorporating different components of preferences, i.e. user and item biases, feature preferences, feature value preferences, conditional dependencies, social influence, and preference dynamicity. The proposed hybrid model employed an Evolutionary Algorithm to optimise the model hyperparameters and the learning rates to be used in Gradient Descent. Then using two popular datasets, we showed that the proposed method achieves significantly better results than TrustSVD, which to the best of our knowledge, is the most accurate state of the art recommender system. We further analysed the interaction effects between the preference components in those two datasets. We concluded that different types of interactions may exist between different components in different datasets. This finding emphasised the importance of designing a component-based approach in preference modelling, which enables understanding of these interaction effects in different datasets and domains, in order to achieve the highest accuracy.

An interesting future direction that we want to follow is related to quantifying the significance of the components and their interactions in different datasets and domains.

Acknowledgments

This project is partially funded by the SunCorp Group.

References

[Aghdam *et al.*, 2015] Mehdi Hosseinzadeh Aghdam, Morteza Analoui, and Peyman Kabiri. A novel nonnegative matrix factorization method for recommender systems. *Applied Mathematics & Information Sciences*, 9(5):2721, 2015.

[Aldrich, 2011] Susan E Aldrich. Recommender systems in commercial use. *AI Magazine*, 32(3):28–34, 2011.

[Aleti and Moser, 2016] Aldeida Aleti and Irene Moser. A systematic literature review of adaptive parameter control methods for evolutionary algorithms. *ACM Comput. Surv.*, 49(3):56:1–56:35, October 2016.

[Guo *et al.*, 2013] G. Guo, J. Zhang, and N. Yorke-Smith. A novel bayesian similarity measure for recommender systems. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2619–2625, 2013.

[Guo *et al.*, 2016] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. A novel recommendation model regularized with user trust and item ratings. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1607–1620, 2016.

[Jamali and Ester, 2010] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142. ACM, 2010.

[Jiang *et al.*, 2012] Meng Jiang, Peng Cui, Rui Liu, Qiang Yang, Fei Wang, Wenwu Zhu, and Shiqiang Yang. Social contextual recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 45–54. ACM, 2012.

[Koren and Bell, 2011] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender systems handbook*, pages 145–186. Springer, 2011.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[Koren, 2008] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[Koren, 2010] Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[Liu *et al.*, 2014] Juntao Liu, Caihua Wu, Yi Xiong, and Wenyu Liu. List-wise probabilistic matrix factorization for recommendation. *Information Sciences*, 278:434–447, 2014.

[Liu *et al.*, 2015] Wenyu Liu, Caihua Wu, Bin Feng, and Juntao Liu. Conditional preference in recommender systems. *Expert Systems with Applications*, 42(2):774–788, 2015.

[Ma *et al.*, 2008] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[Ma *et al.*, 2011] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

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

[Zafari and Moser, 2016] Farhad Zafari and Irene Moser. Feature-aware factorised collaborative filtering. In *Australasian Joint Conference on Artificial Intelligence*, pages 561–569. Springer, 2016.

[Zafari and Moser, 2017] Farhad Zafari and Irene Moser. Modelling socially-influenced conditional preferences over feature values in recommender systems based on factorised collaborative filtering. *Expert Systems with Applications*, Accepted, 2017.

[Zafari and Nassiri-Mofakham, 2016a] Farhad Zafari and Faria Nassiri-Mofakham. Bravecat: iterative deepening distance-based opponent modeling and hybrid bidding in nonlinear ultra large bilateral multi issue negotiation domains. In *Recent Advances in Agent-based Complex Automated Negotiation*, pages 285–293. Springer, 2016.

[Zafari and Nassiri-Mofakham, 2016b] Farhad Zafari and Faria Nassiri-Mofakham. Popponent: Highly accurate, individually and socially efficient opponent preference model in bilateral multi issue negotiations. *Artificial Intelligence*, 237:59–91, 2016.

[Zafari and Nassiri-Mofakham, 2017] Farhad Zafari and Faria Nassiri-Mofakham. Popponent: Highly accurate, individually and socially efficient opponent preference model in bilateral multi issue negotiations (extended abstract). In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[Zafari *et al.*, 2015] Farhad Zafari, Faria Nassiri-Mofakham, and Ali Zeinal Hamadani. Dopponent: A socially efficient preference model of opponent in bilateral multi issue negotiations. *Journal of Computing and Security*, 1(4), 2015.

[Zhang *et al.*, 2014] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92. ACM, 2014.

[Zhao *et al.*, 2015] Haiyan Zhao, Shengsheng Wang, Qingkui Chen, and Jian Cao. Probabilistic matrix factorization based on similarity propagation and trust propagation for recommendation. In *2015 IEEE Conference on Collaboration and Internet Computing (CIC)*, pages 90–98. IEEE, 2015.