Are Features Equally Representative? A Feature-Centric Recommendation

Chenyi Zhang[1,2], Ke Wang[2*], Ee-peng Lim[3], Qinneng Xu[4], Jianling Sun[1] and Hongkun Yu[5]
[1]College of Computer Science, Zhejiang University
[2]School of Computing Science, Simon Fraser University
[3]School of Information Systems, Singapore Management University
[4]Department of Systems Engineering and Engineering Management, City University of Hong kong
[5]Department of Computer Science, University of Illinois at Urbana-Champaign
{chenyiz,wangk}@sfu.ca, eplim@smu.edu.sg, qinnengxu3-c@my.cityu.edu.hk, sunjl@zju.edu.cn, hyu50@illinois.edu

Abstract

Typically a user prefers an item (e.g., a movie) because she likes certain features of the item (e.g., director, genre, producer). This observation motivates us to consider a *feature-centric recommendation* approach to item recommendation: instead of directly predicting the rating on items, we predict the rating on the features of items, and use such ratings to derive the rating on an item. This approach offers several advantages over the traditional item-centric approach: it incorporates more information about why a user chooses an item, it generalizes better due to the denser feature rating data, it explains the prediction of item ratings through the predicted feature ratings. Another contribution is turning a principled item-centric solution into a feature-centric solution, instead of inventing a new algorithm that is feature-centric. This approach maximally leverages previous research. We demonstrate this approach by turning the traditional item-centric latent factor model into a feature-centric solution and demonstrate its superiority over item-centric approaches.

1 Introduction

The objective of recommender systems is to predict user preferences on items based on preferences observed on other items in the past. Two basic approaches are content-based filtering and collaborative filtering:

Content-based filtering: In this approach, keywords or features are used to describe the items and a user profile is built using the past item ratings to summarize the types of items this user likes. This approach follows the design principle that **liking a feature in the past leads to liking the feature in future**. A drawback is that if a user A has not liked any feature of an item X in the past, X will not be recommended to the user, even though many users with the same profile as A like X.

Collaborative filtering: This approach addresses the above problem by collaborative learning: if a user A liked some items that were also liked by user B, A is likely to share the same preference with B on another item. The design principle of this approach is that **liking same items (as other users) leads to liking more same items**.

Hybrid approaches combine content-based filtering and collaborative filtering. Despite their differences, all these approaches are *item-centric* in that major activities such as rating collection, model building, and rating prediction are centered around items. In this work, we consider another design principle: **liking same features (as other users) leads to liking more same features**, which leads to a new approach called *feature-centric recommendation*. Let us motivate this design principle and our approach.

1.1 Motivation

A user likes an item because of some *specific features* of the item. When a user likes an item, she may like some features of the item but is not impressed with other features; consequently, two users may like the same item for different reasons. For example, a user may like the rate of a hotel but not its service, while another user may like the cleanliness of the hotel but nothing else. If the above observation holds, the design principle "liking same items (as other users) leads to liking more same items" practised by the standard collaborative filtering may not work.

To explain our point, let us consider the toy example with four items W={anti-allergy, rose}, X={anti-allergy, rose}, Y={anti-allergy, orange}, and Z={sun proof, rose} where {} contains the features for the item. In the history, suppose that user A loves W and X due to "anti-allergy" and user B loves W, X, and Z due to "rose". In traditional collaborative filtering, it will recommend Z to user A since A and B both like W and X whereas A would rather love Y. On the other hand, for content-based filtering, Y has the feature "anti-allergy" and Z has the feature "rose" as in W and X. Therefore, the score is the same, but it is obvious that A will love Y more than Z from the view of feature level.

User's preferences on features are available in many real life rating systems. For example, in the hotel rating systems a user can rate specific features (cleanliness, service, etc.) of a hotel. Other systems (Sen, Vig, and Riedl 2009) allow users to explicitly express preferences on features by attaching personal tags to an item. The ratings of features (e.g., director, actor) can also be implicitly inherited from the features of items (e.g., movie). Such information manifests the user's reasons for rating the item, led by different preferences on features.

[*] Work done while visiting the Living Analytics Research Centre, Singapore Management University

1.2 Our Approach

Our approach is motivated by the above discussion, therefore, if we can predict user's preference on features, we are able to predict user's preference on an item; if we can express observed preference on features in the same format as observed preference on items, any principled collaborative filtering algorithm for items can be applied to predict user's preference on features. This thinking leads to a *feature-centric recommendation* approach in which features of items are the "King": given the usual user-item rating matrix R, we first convert item ratings into feature ratings and obtain a user-feature rating matrix R'. We directly perform collaborative filtering on R', which practices our design principle "liking same features (as other users) leads to liking more same features". The output is a model for predicting a user's rating on a feature.

To predict a user's rating on an unrated item, we need to integrate the predicted feature ratings to derive the rating for the item. While sum and average are obvious choices, features are not equally representative, e.g., the feature "anti-allergy" clearly has a more significance than the other features for user A. So we present two novel integration approaches, one is heuristic based and one is regression based, to give different significance for feature preferences.

The key innovation is that our approach transforms the item ratings into feature ratings and later the modeling is fully at the feature level. To this end, our approach takes the advantages of collaborative filtering and content-based filtering, e.g., even if items are not shared among users, features of items may still be shared, which helps address the cold start problem of a new item. As the final prediction on the item rating is an integration of a set of feature ratings, our modeling actually involves no "items", which is fundamentally different to the modeling of the existing feature based recommendations (Chen et al. 2012; Han and Karypis 2005). Their modelings all have "items" (e.g., latent item vectors) which directly affect the final prediction while features are used as a finer description to regularize the item ratings. However, we fully rely on the feature ratings and may avoid the side effects introduced by items.

The rest of the paper is organized as follows: Section 2 discusses related works. Section 3 introduces our model. Section 4 presents our experimental studies. Finally we conclude the paper.

2 Related Work

2.1 Item-Centric Approaches

Many content-based filtering, collaborative filtering and hybrid approaches (Adomavicius and Tuzhilin 2005) are item-centric in that rating, model building, similarity, and filtering are centered around items, and feature information is only used to measure similarity of items (for content-based filtering). (Han and Karypis 2005) proposed that "user who bought products with features also bought a product with the same features", but still treated items as central roles and features as side information. Our approach is feature-centric: it captures the preference of features at the rating time, which is different from learning the user profile as the outcome of content-based filtering, and it performs collaborative filtering on feature ratings and predicts the rating of features. The prediction of item ratings is performed only at the final step by integrating the predicted ratings of features.

2.2 Latent Factor Models

Recent works use the latent factor model to extract low-dimensional latent user and item vectors for predicting the rating of items (Salakhutdinov and Mnih 2008a; 2008b; Zhang et al. 2014a), and several works (Zhang et al. 2014b; Wang and Blei 2011) extend items with content in this model. The regression-based latent factor model (Agarwal and Chen 2009) incorporated features and past interactions to regress the latent vectors. Items with similar features tend to have similar latent vectors, so features have indirect impact on the final ratings. (Agarwal, Chen, and Pang 2011) further extended (Agarwal and Chen 2009) by modeling user-generated opinionated texts. In (Gantner et al. 2010), the features of users and items are used to predict the latent factors of new users and new items; existing users and items do not benefit from the available feature information. The collaborative topic regression (CTR) (Wang and Blei 2011) studies the recommendation for scientific articles with each article being modeled by topic modeling on the text content of the article. The factorization machine (FM) (Rendle 2012) models multidimensional variable interactions (user, item, feature, etc.) through latent vectors. The tensor factorization (Karatzoglou et al. 2010) generalizes the rating matrix with additional context information. The co-factorization machine (Hong, Doumith, and Davison 2013) couples the learning of two FMs to study two aspects of tweeter data. All these methods treat the features of an item equally as side information and no preference is captured on features.

2.3 Tag-Aware Recommendation

(Sen, Vig, and Riedl 2009) predicts users' ratings for items based on inferred preferences for tags, but the preferences are global for all items, that is, a tag is either liked or disliked for all items. The work in (Gedikli and Jannach 2010; 2013) improves upon this by predicting tag preferences in the context of an item. All these methods infer the preferences for the user's own tags; if an unrated item is attached with tags that the user has never used before, no prediction can be made for the item. Our method does not have this problem because it employs collaborative filtering on feature ratings, which can predict a rating for any pair of user and feature. The tag-aware recommendation in (Tso-Sutter, Marinho, and Schmidt-Thieme 2008) models a 3-way relation $< user, item, tag >$ by 2-way relations $< user, tag >$, $< item, tag >$, and $< user, item >$, which cannot express the 3-way information that a user i tags an item j using a tag t. There is a similar problem with (Zhou et al. 2009). In (Zhen, Li, and Yeung 2009), the tag information acts as a new regularization term in matrix factorization to constrain the latent vectors between users who used similar tags. Features play a more central role in our model in that we learn latent vectors for features and predict the rating of features. If a user prefers an item because of specific features of the

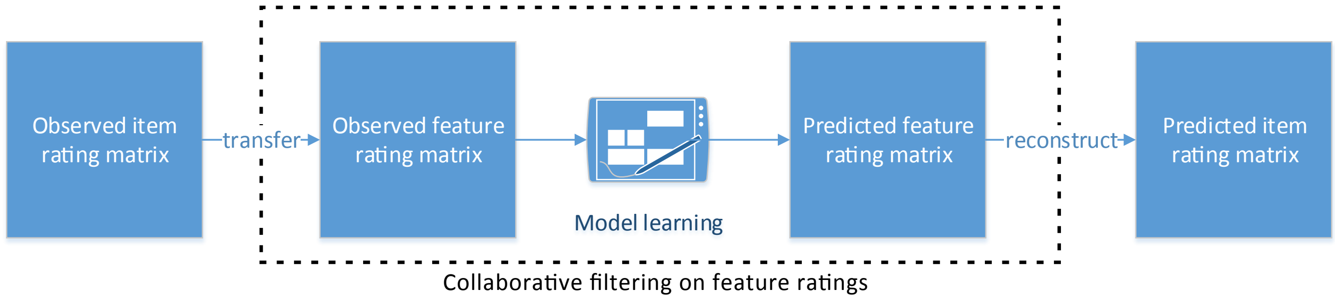
Observed item rating matrix
transfer
Observed feature rating matrix
Model learning
Predicted feature rating matrix
reconstruct
Predicted item rating matrix
Collaborative filtering on feature ratings

Figure 1: The feature-centric recommendation approach

item, our feature-centric approach is more sensitive to user preferences.

3 Feature-Centric Recommendation

We present a feature-centric recommendation approach to utilize user's feature preferences to improve recommendation of items. This approach is shown in Figure 1 where the dashed box encompasses a standard collaborative filtering method. We consider the latent factor model for this box, but it could be replaced with any other collaborative filtering methods. We discuss the key steps of extracting an observed feature rating matrix and predicting item ratings from feature ratings in the following sections.

3.1 Extracting User-Feature Rating Matrix

We assume there are I users, J items, T features. Each item j is associated with a bag of features, denoted by B_j. For a feature t, S_t denotes the set of items j such that $t \in B_j$. The original rating data can be represented by an $I \times J$ user-item rating matrix R in which each element r_{ij} indicates user i's rating value to item j.

In addition, when the user i rates the item j, the user may optionally rate or select (such as tagging) some features t of item j. If the user rates the feature t for item j, $h_{it}(j)$ denotes this rating. If the user i selects the feature t but does not rate t, $h_{it}(j) = r_{ij}$, which is user i's rating on item j. If the user i does not select any feature at all when rating the item j, $h_{it}(j) = r_{ij}$ for all features t of item j as we believe that the user implicitly selects all features. In all other cases, $h_{it}(j)$ is undefined.

User-feature rating matrix. For user i and feature t, $\{h_{it}(j)\}$ denotes the bag of defined ratings $h_{it}(j)$ for all items $j \in S_t$. We extract an $I \times T$ *user-feature rating matrix* denoted by R', there is one row for each user i, one column for each feature t, and the entry for (i, t) is equal to $\{h_{it}(j)\}$. As the toy example in Section 1.1, if user A rates the item W with the rating 4 and the item X with the rating 4, and selects the feature "anti-allergy" for both, so $h_{A,anti-allergy}(W) = 4$ and $h_{A,anti-allergy}(X) = 4$, and the entry for ($A$,"anti-allergy") is $\{4, 4\}$.

We adopt latent factor model (Salakhutdinov and Mnih 2008b) on R' to produce the latent user vector u_i for each user i and the latent feature vector f_t for each feature t, and the objective is to minimize $\sum_{j \in S_t} \varepsilon_{ijt}(h_{it}(j) - u_i^T f_t)^2$ where ε_{ijt} is equal to 1 if $h_{it}(j)$ is defined, and is equal to 0 otherwise. User i's predicated rating on feature t is given by $u_i^T f_t$. This part is the standard method for the latent factor model except that items are substituted by features. The interested reader please refer to (Salakhutdinov and Mnih 2008b) for the detailed inference.

3.2 Predicting Item Ratings by Heuristic

We present two heuristic strategies for integrating the predicted feature ratings to derive the predicted item rating for a user in this section. One strategy is using the average of feature ratings to predict the rating $\hat{r}_{ij}$ for item j, that is,

$$\hat{r}_{ij} = \frac{1}{|B_j|} \sum_{t \in B_j} u_i^T f_t \tag{1}$$

This prediction treats all features in B_j equally because each $u_i^T f_t$ has the weight $1/|B_j|$. It does not take into account, for example, whether the user occasionally selects the feature t by chance or consistently selects the feature. Below, we present another strategy that consider such differences.

The second strategy is to introduce a weighting scheme specific to individual users. If a user i selects a feature t frequently, the user is more interested in t. Besides relative frequency of selection, the absolute number of selection also matters. For example, selecting a feature twice out of 3 ratings has the same frequency as selecting a feature 20 times out of 30 ratings, but the latter has more statistical significance. This strategy suggests that features are not equally representative.

We propose a single weighting scheme to account for both relative frequency and statistical significance. Suppose that a user i has rated N items, among them, the feature t was selected s times. We can regard this as one sample where the event t was observed s times in N trials, where $\hat{p} = \frac{s}{N}$ is the *observed proportion*. We want to bound the *true proportion* that the user selects the feature t. Let $CI(s, N)$ denote the confidence interval for user i selecting feature t. We adopt the following Wilson score (Wilson 1927) interval since it is an improvement over the usual normal approximation.

$$CI(s, N) = [c - \sigma, c + \sigma] \tag{2}$$

where

$$c = \frac{1}{1 + \frac{1}{N} z^2} (\hat{p} + \frac{1}{2N} z^2)$$

$$\sigma = \frac{1}{1+\frac{1}{N}z^2} z \sqrt{\frac{1}{N}\hat{p}\left(1-\hat{p}\right) + \frac{1}{4N^2}z^2}$$

z is the $1-\frac{1}{2}\alpha$ percentile of a standard normal distribution and α is the error percentile. For a 95% confidence level the error α is 5%, so $1-\frac{1}{2}\alpha = 0.975$ and $z = 1.96$.

A larger c and a smaller interval size σ represent a more significant selection of t. Therefore, we use the mid-point of the lower bound $c-\sigma$ and the interval center c to measure the weight of selecting t by user i: $l_{it} = \frac{1}{2}(c-\sigma+c) = c-\frac{1}{2}\sigma$. The predicted rating $\hat{r}_{ij}$ of item j by user i is then defined as

$$\hat{r}_{ij} = \frac{1}{L_j}\sum_{t\in B_j} l_{it} u_i^T f_t \tag{3}$$

where $L_j = \sum_{t\in B_j} l_{it}$. Note that this weighting scheme is unique to features, not items, because only a feature can be selected by a user multiple times.

3.3 Predicting Item Ratings through Regression

In the previous section, the weighting schemes are computed by heuristic, which may not capture the essential by model fitting. In this section, we propose a regression model to automatically learn the global weighting for features. We assume that there exists a weighting vector $\mathbf{w}$, with $w_t \in \mathbf{w}$ representing the importance for each feature t. The training set contains historic ratings, where $\mathbf{x}_k$ is the kth input data and y_k is the kth output data, in particular, if this rating is made by user i, $y_k = r_{ij}$, $\mathbf{x}_k(t) = u_i f_t$ if $t \in B_j$ and $\mathbf{x}_k(t) = 0$ otherwise. The regression model is trained by the relationship between input and output data: $y_k = \mathbf{w}^T\mathbf{x}_k + b_i$. Following support vector regression (Smola and Scholkopf 2004), our goal is to find optimal weighting $\mathbf{w}$ and user specific bias b_i that fit the model best. The optimization problem becomes,

$$\min_{\mathbf{w},\xi,\xi^*} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_k(\xi_k^2 + \hat{\xi}_k^2) \tag{4}$$

$$s.t. \quad |\mathbf{w}^T\mathbf{x}_k + b_i - y_k| \leq \varepsilon + \xi_k; \tag{5}$$

$$\xi_k, \hat{\xi}_k \geq 0; \forall k. \tag{6}$$

where $\xi_k, \hat{\xi}_k$ are slack variables and C is a constant.

The primal Lagrangian is,

$$\mathcal{L}_P = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_k \alpha_k(\mathbf{w}^T\mathbf{x}_k + b_i - y_k - \varepsilon - \xi_k) + \sum_k \hat{\alpha}_k(y_k - \mathbf{w}^T\mathbf{x}_k - b_i - \varepsilon - \hat{\xi}_k) + \frac{C}{2}\sum_k(\xi_k^2 + \hat{\xi}_k^2) \tag{7}$$

where $\alpha_k, \hat{\alpha}_k$ are Lagrange multipliers. We take the derivatives with respect to $\mathbf{w}, b_i, \xi_k, \hat{\xi}_k$, leading to the KKT conditions (Kuhn and Tucker 1951) as follows:

$$\mathbf{w} = \sum_i(\alpha_k - \hat{\alpha}_k)x_k, \quad \xi_k = \alpha_k/C, \quad \hat{\xi}_k = \hat{\alpha}_k/C \tag{8}$$

We skip the comprehensive inference as SVR is a principled method. More details refer to (Smola and Scholkopf 2004). Once the optimal weighting vector $\mathbf{w}$ and the bias term b_i are found, the predicted rating for specific item j is given by:

$$\hat{r}_{ij} = \sum_{t\in B_j} w_t u_i^T f_t + b_i \tag{9}$$

Table 1: Statistics of data sets

	Delicious	**Lastfm**	**DBLP**	**Movielens**
User	1867	2100	6815	1857
Item	69223	18744	78745	4721
Feature	40897	12647	81901	8288
Item ratings	104799	71064	436704	20607
Feature ratings	437593	186479	2554597	36885
Density (item)	8.1×10^{-4}	1.8×10^{-3}	8.1×10^{-4}	2.3×10^{-3}

Feature selection. We conduct a 2-step feature selection that aims at the representative features. (1) we calculate the cosine similarity to quantify correlations between various predictors (feature ratings) and the item ratings to identify the best predictors. According to the result, we remove features with relatively low similarity by a threshold. (2) we fit the model with the remaining features. If the model performance is close to the original one, we believe that the removed features are less representative. This procedure is optional but helps the prediction accuracy.

4 Experimental Evaluation

We report our findings on the evaluation of the proposed feature-centric recommendation against well known baselines using real life data sets. We first introduce data sets, baseline methods, and evaluation metrics.

4.1 Data Sets

We employed four data sets: *Delicious*, *Lastfm*, *DBLP*, and *Movielens*. The first two data sets were recommended as benchmark data sets for studying recommender systems by the 2011 HetRec conference[1]. These data sets contain user's tagging information on bookmarks and music songs, which expresses user's ratings or preferences on items. We treat tags as the features of an item. Delicious contains 1867 users' ratings on 69223 items with 40897 unique features. Lastfm contains 2100 users' ratings on 18744 items with 12647 unique features. The third data set *DBLP* contains authors, papers and citation information from an academic network. We treated authors as users, papers as items, each publishing/citation of a paper as user's rating on the paper, and treated the venues and authors of a paper as the features of the paper. After removing the users with fewer than 10 papers from the original DBLP data set[2], the final data set contains 6815 users' ratings on 78475 items with 81858 unique features. All the above data sets have binary ratings. The fourth data set *Movielens*, also recommended by the 2011 HetRec conference, was collected from a movie review system. This data set has the ratings ranged from 1 to 5. We removed those movies without any ratings. The resulting data set has 1857 users' ratings on 4721 items with 8288 unique

[1] http://www.grouplens.org/data sets/hetrec-2011/

[2] http://arnetminer.org/citation

features (i.e., tags). The statistics of these data sets are found in Table 1. We conducted 10-fold cross validation for all data sets.

4.2 Evaluated Methods

The first baseline is the probabilistic matrix factorization that ignores features of items:

Probabilistic matrix factorization (denoted PMF): This method adopts matrix factorization on the user-item rating matrix (Salakhutdinov and Mnih 2008b). Following (Salakhutdinov and Mnih 2008b), we set the parameters $\lambda_u = \lambda_v = 0.01$.

The next four baselines consider features of an item. All the baselines were previously proposed in the literature.

Collaborative topic regression (denoted CTR): This is matrix factorization with topic modeling applied to features of items (Wang and Blei 2011). Following (Wang and Blei 2011), we set the parameters $\lambda_u = \lambda_v = 0.01, \alpha = {}^{50}/{}_{D}$ and $\beta = 0.01$.

Factorization machine (denoted FM): This is the factorization machine approach in (Rendle 2012). FM takes selected features into consideration equally while our model gives more importance to representative features. We run the code of (Rendle 2012) with the default settings. Note that we need not compare with (Chen et al. 2012) since it can be modeled by FM as indicated in (Rendle 2012).

Regression latent factor model (denoted RLFM): This is the regression based latent factor model in (Agarwal and Chen 2009). RLFM incorporated features as side information to regress the latent vectors so as to improve the performance. We run the code of (Agarwal and Chen 2009) with the default settings.

Similarity based method (denoted SIM): This is a content-based filtering approach that uses the SVM regression (Gedikli and Jannach 2013) to predict the user's ratings on items according to inferred feature preferences. Note that the computation of SIM is not in the latent space.

The next method is the feature-centric approach proposed in this work:

Feature-centric recommendation (denoted FCR): This is the feature-centric solution proposed in Section 3. We denote FCR-a, FCR-u and FCR-r for different integration strategies, i.e., averaged heuristic, user-specific heuristic and regression model. FCR-a is computed by Eq. (1), FCR-u is computed by Eq. (3), and FCR-r is computed by Eq. (9).

For all methods except for SIM, we adopt the dimensionality of $D = 20$ for latent vectors and the learning rate of $\eta = 0.0001$.

4.3 Evaluation Metrics

RMSE (root mean squared error) and **MAE** (mean absolute error) quantify the difference between the rating values predicted by a recommender and the true values in the testing set. These two metric are defined as follows: $RMSE = \sqrt{\frac{1}{n}\sum_{i,j}(r_{ij} - \hat{r}_{ij})^2}$, $MAE = \frac{1}{n}\sum_{i,j}|r_{ij} - \hat{r}_{ij}|$, where r_{ij} is the true rating value, $\hat{r}_{ij}$ is the predicted rating value, and n is the number of ratings in the testing set. The smaller these values are, the better the result is. As pointed out in (Koren 2009), achievable RMSE values lie in a quite compressed range and small improvements in RMSE terms can have a significant impact on the quality of the top few presented recommendations.

4.4 Experimental Results

Table 2 shows RMSE and MAE with standard errors for different methods. Among them, FM and RLFM are recent developed models which can incorporate features for better predications. The best performers for each data set are highlighted in bold face. All reported RMSE and MAE are the average of the 10 runs in the 10-fold cross-validation.

First, PMF performs poorly on all data sets since it only considers the item ratings and ignores the feature information. When this matrix is sparse, the ratings as only the similarity information among users are hardly enough to make accurate recommendation. CTR slightly improve the performance over PMF by using features as the side information to regularize the original matrix. The improvement is not significant since the role of features is limited to regularization; there is no direct participation in rating prediction. On Lastfm, their performances are even worse than PMF. The next three baselines, FM, RLFM, SIM, further improves the performance with similar improvement. FAR has a similar performance to CTR as both methods use features as side information to regularize the original matrix.

FCR-a is our proposed feature-centric method with averaged heuristic and performs close to FM because these methods involve features in matrix factorization and infer latent feature vectors for prediction. However, FM involves the pairwise interactions between latent item vectors and latent feature vectors, which may be complex and improper in real-life applications; on the contrary, our model provides a simple and clean solution.

By incorporating the user-specific heuristic, FCR-u achieves significant improvements over FCR-a. This weighting scheme gives more trust to features that are more frequently selected by users. Through a regression model, FCR-r finds proper weighting for features and achieves better results compared to FCR-u.

Overall, FCR is the the best performing method for all data sets, which verifies the effectiveness of the proposed feature-centric approach. Among the three integration strategies, FCR-r performs the best on Delicious and Lastfm, which suggests that the regression model yields better weighting for features. For example, the rating density is very sparse for DBLP and Delicious and matrix factorization on traditional user-item rating matrix works poorly. In the feature-centric FCR, this problem does not occur because the features of items act as a media for collaborative learning between users. For the DBLP data set, the users working in AI areas may focus on AAAI, IJCAI, ICML conferences (which are features of papers), so even the citation/publishing data is sparse, the interests of users are closely related through similar publishing venues. For the Movielens data set, the improvement is least mainly because user's interests are more diverse on movies, leading to less collaborative learning effect through features.

Table 2: $RMSE$ and MAE of four data sets

	Delicious		**Lastfm**		**DBLP**		**Movielens**	
Methods	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
Baselines								
PMF	0.8907 ± 0.0046	0.8081 ± 0.0048	0.4449 ± 0.0040	0.3179 ± 0.0022	0.5060 ± 0.0021	0.3800 ± 0.0020	1.1271 ± 0.0247	0.8622 ± 0.0197
CTR	0.7844 ± 0.0004	0.7431 ± 0.0005	0.5078 ± 0.0025	0.4084 ± 0.0025	0.4943 ± 0.0021	0.3653 ± 0.0018	1.0880 ± 0.0191	0.8315 ± 0.0156
FM	0.3551 ± 0.0017	0.2906 ± 0.0020	0.3239 ± 0.0022	0.2534 ± 0.0030	0.1821 ± 0.0023	0.1167 ± 0.0020	1.2049 ± 0.0229	0.9467 ± 0.0194
RLFM	0.4182 ± 0.0010	0.3978 ± 0.0010	0.3208 ± 0.0014	0.2235 ± 0.0015	0.2297 ± 0.0007	0.1930 ± 0.0007	1.0662 ± 0.0215	0.8056 ± 0.0145
SIM	0.4001 ± 0.0008	0.3872 ± 0.0011	0.3269 ± 0.0013	0.2941 ± 0.0013	0.3064 ± 0.0003	0.3032 ± 0.0003	1.0137 ± 0.0190	0.7616 ± 0.0123
Proposed Methods								
FCR-a	0.3169 ± 0.0020	0.2396 ± 0.0016	0.3790 ± 0.0023	0.3062 ± 0.0022	0.2043 ± 0.0016	0.1393 ± 0.0017	0.9966 ± 0.0148	0.7770 ± 0.0105
FCR-u	0.2572 ± 0.0023	0.1645 ± 0.0014	0.2455 ± 0.0032	0.1468 ± 0.0020	0.1204 ± 0.0010	$\mathbf{0.0739} \pm 0.0006$	$\mathbf{0.9515} \pm 0.0150$	0.7306 ± 0.0092
FCR-r	$\mathbf{0.2176} \pm 0.0011$	$\mathbf{0.1513} \pm 0.0009$	$\mathbf{0.1868} \pm 0.0020$	$\mathbf{0.1066} \pm 0.0016$	$\mathbf{0.1064} \pm 0.0004$	0.0841 ± 0.0003	0.9724 ± 0.0201	$\mathbf{0.7208} \pm 0.0125$

Table 3: Paired t-Test(2-tail) of FCR-a and FCR-r

t-Test	Delicious	Lastfm	DBLP	Movielens
RMSE	2.4×10^{-15}	1.5×10^{-16}	5.1×10^{-16}	1.4×10^{-4}
MAE	2.2×10^{-15}	1.6×10^{-17}	8.1×10^{-15}	1.3×10^{-7}

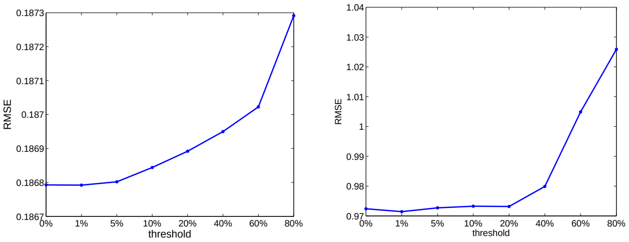

Figure 2: Feature selection for FCR-r on Lastfm (left) and MovieLens (right). y-axis represents RMSE and x-axis represents the percentage of the features with lower correlations removed

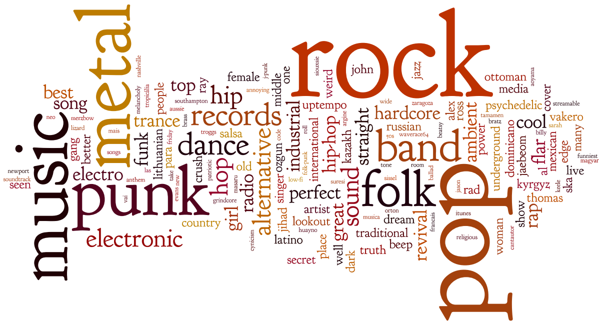

Figure 3: Representative features on Lastfm

t-Test. To further verify the statistical significance of the improvement introduced by the regression model, we conducted the paired t-Test (2-tail) on FCR-a and FCR-r over 10 folds. As shown in Table 3, the t-Test results (p-values) are less than 0.01, which suggests that the improvement of FCR-r over FCR-a is statistically significant.

Representative features. We perform the 2-step feature selection for FCR-r and study the representative features. Figure 2 presents the RMSE results on Lastfm and MovieLens when removing those features with lower correlations. We observe that the lowest 1% features have no contributions to the model since the RMSE is slightly improved without them. With more features removed, the RMSE is steady at first, indicating those features are less representative, i.e., lowest 5% on Lastfm and lowest 20% on Movielens. Then the RMSE goes up steeply when removing those features with higher correlations. Note that the results are still better than most baselines even if the lowest 80% of the features are removed. This also coincides with our thinking that the features with higher correlations are more representative. Figure 3 demonstrates the representative features on Lastfm with visualization tools.

Discussions. In summary, the proposed feature-centric approach demonstrates superiority over item-centric approaches. This superiority is especially obvious for a sparse rating matrix in which case collaborative filtering on features is a much better option than collaborative filtering on items because feature ratings are denser than item ratings. Content-based filtering, i.e., SIM, extends items with content/features, and more recently, several works extend collaborative filtering (i.e., the latent factor model) to items with content and features, i.e., CTR and RLFM. The improvement is limited because features are used as auxiliary information such as a new regularization term in matrix factorization. Our feature-centric approach acknowledges the upmost importance of features in item preferences by allowing the features to play a central role from rating capturing to model building to rating prediction, which yields significant improvements. The studies also suggested that a model based regression is a viable weighting strategy for integrating feature ratings and finding representative features. Those representative features should be highly considered in real-life recommendations.

5 Conclusion

Traditional recommender systems are item-centric in that all steps are centered around items: rating collection, model extraction, and rating prediction. Even if features of items are considered, they serve only the description of items in order to enhance the similarity comparison between items and users. This work is based on a simple observation: a user prefers an item because she likes certain features of the item; therefore, a good prediction on features would lead to a good prediction on items. This observation motivates a feature-centric approach that aims to predict feature ratings in order to predict item ratings. Our contribution is to reformulate the item prediction problem into the feature prediction problem and turn the solution into a solution for item prediction. Two clear benefits of this approach are: it enables a principled item prediction (e.g., latent factor model) on feature prediction with little changes, and it enables collaborative filtering on the denser feature ratings, therefore, maximizes the effect of collaborative filtering and addresses the well known sparsity of rating data.

Acknowledgements

This work is partially supported by China Knowledge Centre for Engineering Sciences and Technology (No. CKCEST-2014-1-5), partially supported by a Discovery Grant from Natural Sciences and Engineering Research Council of Canada, and partially supported by the Singapore National Research Foundation under its International Research Centre@ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA). Ke Wang's work is partially done when he visited SA Center for Big Data Research hosted in Renmin University of China. This Center is partially funded by a Chinese National 111 Project Attracting International Talents in Data Engineering and Knowledge Engineering Research.

References

Adomavicius, G., and Tuzhilin, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transcation on Knowledge and Data Engineering (TKDE)* 17(6):734–749.

Agarwal, D., and Chen, B.-C. 2009. Regression-based latent factor models. In *KDD*, 19–28. ACM.

Agarwal, D.; Chen, B.-C.; and Pang, B. 2011. Personalized recommendation of user comments via factor models. In *EMNLP*, 571–582. Association for Computational Linguistics.

Chen, T.; Zhang, W.; Lu, Q.; Chen, K.; Zheng, Z.; and Yu, Y. 2012. Svdfeature: a toolkit for feature-based collaborative filtering. *The Journal of Machine Learning Research* 13(1):3619–3622.

Gantner, Z.; Drumond, L.; Freudenthaler, C.; Rendle, S.; and Schmidt-Thieme, L. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *ICDM*, 176–185. IEEE.

Gedikli, F., and Jannach, D. 2010. Rating items by rating tags. In *Proceedings of the 2010 Workshop on Recommender Systems and the Social Web at ACM RecSys*, 25–32.

Gedikli, F., and Jannach, D. 2013. Improving recommendation accuracy based on item-specific tag preferences. *ACM Transactions on Intelligent Systems and Technology (TIST)* 11:1–11:19.

Han, E.-H. S., and Karypis, G. 2005. Feature-based recommendation system. In *CIKM*, 446–452. ACM.

Hong, L.; Doumith, A. S.; and Davison, B. D. 2013. Co-factorization machines: modeling user interests and predicting individual decisions in twitter. In *WSDM*, 557–566. ACM.

Karatzoglou, A.; Amatriain, X.; Baltrunas, L.; and Oliver, N. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Recsys*, 79–86. ACM.

Koren, Y. 2009. Collaborative filtering with temporal dynamics. In *KDD*, 447–456. New York, NY, USA: ACM.

Kuhn, H. W., and Tucker, A. W. 1951. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, 481–492. University of California Press.

Rendle, S. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3(3):57:1–57:22.

Salakhutdinov, R., and Mnih, A. 2008a. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML*, 880–887. ACM.

Salakhutdinov, R., and Mnih, A. 2008b. Probabilistic matrix factorization. In *NIPS*, 1257–1264.

Sen, S.; Vig, J.; and Riedl, J. 2009. Tagommenders: connecting users to items through tags. In *WWW*, 671–680. ACM.

Smola, A. J., and Scholkopf, B. 2004. A tutorial on support vector regression. *Statistics and computing* 14(3):199–222.

Tso-Sutter, K. H.; Marinho, L. B.; and Schmidt-Thieme, L. 2008. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the 2008 ACM symposium on Applied computing*, 1995–1999. ACM.

Wang, C., and Blei, D. M. 2011. Collaborative topic modeling for recommending scientific articles. In *KDD*, 448–456. ACM.

Wilson, E. B. 1927. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association* 22(158):209–212.

Zhang, C.; Wang, K.; Yu, H.; Sun, J.; and Lim, E.-p. 2014a. Latent factor transition for dynamic collaborative filtering. In *SDM*, 452–460.

Zhang, C.; Zhao, X.; Wang, K.; and Sun, J. 2014b. Content+ attributes: A latent factor model for recommending scientific papers in heterogeneous academic networks. In *ECIR*, 39–50.

Zhen, Y.; Li, W.-J.; and Yeung, D.-Y. 2009. Tagicofi: tag informed collaborative filtering. In *Recsys*, 69–76. ACM.

Zhou, T. C.; Ma, H.; King, I.; and Lyu, M. R. 2009. Tagrec: Leveraging tagging wisdom for recommendation. In *International Conference on Computational Science and Engineering*, volume 4, 194–199. IEEE.