IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 1, 2021, accepted April 9, 2021, date of publication April 20, 2021, date of current version May 5, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3074365*

# FeatureMF: An Item Feature Enriched Matrix Factorization Model for Item Recommendation

**HAIYANG ZHANG[1], IVAN GANCHEV[2,3,4], (Senior Member, IEEE), NIKOLA S. NIKOLOV[4,5], ZHANLIN JI[4,6], (Member, IEEE), AND MÁIRTÍN O'DROMA[4], (Life Senior Member, IEEE)**

[1]Department of Computer Science, The University of Sheffield, Sheffield S10 2TN, U.K.
[2]Department of Computer Systems, University of Plovdiv Paisii Hilendarski, 4002 Plovdiv, Bulgaria
[3]Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, 1040 Sofia, Bulgaria
[4]Telecommunications Research Centre (TRC), University of Limerick, Limerick, V94 T9PX Ireland
[5]Department of Computer Science and Information Systems, University of Limerick, Limerick, V94 T9PX Ireland
[6]College of Information Engineering, North China University of Science and Technology, Tangshan 063210, China

Corresponding authors: Ivan Ganchev (ivan.ganchev@ul.ie) and Zhanlin Ji (zhanlin.ji@ncst.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFE0135700, in part by the Bulgarian National Science Fund (BNSF) under Grant KP-06-IP-CHINA/1, and in part by the Telecommunications Research Centre (TRC), University of Limerick, Ireland.

**ABSTRACT** Matrix Factorization (MF) is one of the most successful Collaborative Filtering (CF) techniques used in recommender systems due to its effectiveness and ability to deal with very large user-item rating matrix. However, when the rating matrix sparseness increases its performance deteriorates. Expanding MF to include side-information of users and items has been shown by many researchers both to improve general recommendation performance and to help alleviate the data-sparsity and cold-start issues in CF. In regard to item feature side-information, most schemes incorporate this information through a two stage process: intermediate results (e.g., on item similarity) are first computed based on item attributes; these are then combined with MF. In this paper, focussing on item side-information, we propose a model that directly incorporates item features into the MF framework in a single step process. The model, which we name FeatureMF, does this by projecting every available attribute datum in each of the item features into the same latent factor space with users and items, thereby in effect enriching item representation in MF. Results are presented of comparative performance experiments of the model against three state-of-the-art item information enriched models, as well as against four reference benchmark models, using two public real-world datasets, Douban and Yelp, with four training:test ratio scenarios each. It is shown to yield the best recommendation performance over all these models across all contexts including data-sparsity situations, in particular, achieving over 0.9% to over 6.5% MAE recommendation performance improvement over the next best model, HERec. FeatureMF is also found to alleviate cold start and to scale well, almost linearly, in regard to computational time, as a function of dataset size.

**INDEX TERMS** Collaborative filtering, matrix factorization, item features, cold start, data sparsity.

## I. INTRODUCTION

In the modern 'Big Data' era, recommender systems have become the essential tools that address a myriad of services for Internet users, e.g., in the context of the vast range of information and services readily available such as web browsing [1], [2] or IoT scenarios [3], by assisting users to discover what they need or receive timely personalized suggestions and recommendations. Collaborative Filtering (CF) based recommendation techniques are among the most widely used in recommender systems, e.g., [3]–[7]. CF is fundamentally based on assessing and making use of relations and interactions between users and items over large populations of both to make recommendations to other users or of other items, effectively to identify items that would be preferred by a particular user. However, it is still a challenge to deal with the increasing sparseness of user-item rating matrix and the cold-start problem, which occurs when users/items are just added to a system with only a few ratings [6], [7].

To tackle such issues, hybrid CF methods that combine CF methods with various kinds of additional information sources, sometimes referred to as side-information, related

The associate editor coordinating the review of this manuscript and approving it for publication was Wen-Sheng Zhao.

to both the users and the items in the recommendation process have been extensively studied in recent years. Among different CF methods, Matrix Factorization (MF) is one of the more popular ones due to its scalability and its flexibility in incorporating additional information; cf, for instance, [4], [5], [8]. User side-information, e.g. extracted via social networks [9], [10], user demographics [11], [12] or reviews [13], [14], has been widely used to incorporate into MF and proven being helpful at improving the recommendation performance. However, for natural personal and privacy rights and reasons, users are usually resistant to having their personal information accessed by a third-party agent or to spending extra time on supplying their opinions (i.e. tags, comments, reviews, etc.) [15]. Compared to user side-information, item side-information is more readily available and easier to collect in real-world applications. Item side-information, such as item features that are generated by domain experts to represent item characteristics, has been popular as a source of information to be tapped and incorporated into MF schemes [16], [17].

Among the more notable MF-based approaches incorporating item side-information are those either using item features to compute item similarities [16]–[19] or learning embedding representations of users and items based on item features [20]. These approaches predict users' ratings in two stages, where the item similarities or user/item embeddings are first computed based on item features and then incorporating the result into MF. Besides the complexity of it being a two step process, the approach brings an additional uncertainty. This may be inferred from the variety of similarity measures that have been proposed, and it is still unclear which of these works best reflects the closeness of two items.

In this paper, a novel MF-based model, FeatureMF, that incorporates item features directly, without a pre-computational stage, into the MF framework, is proposed. It is primarily focused on alleviating the data-sparsity problem but with an expectation of helping to alleviate the cold-start *item* problem also. FeatureMF treats the feature information of items as item-feature relations and aims to project available attributes in each of the features into the same latent factor space as the one for users and items. Therefore, each item is represented by not only its own latent-factor vector but also its related attributes. Performance comparison experiments using two real-world datasets are presented. These demonstrate that FeatureMF performs better than both the traditional MF models (where no item side-information is taken into account) -as of course might be expected- and more recent the state-of-the-art item information enriched MF-based models [18]–[20].

The key contributions of this work are summarised in the following:

1) A novel MF-based model, FeatureMF, that integrates item features with the MF framework is proposed. This model represents attribute data in each of the item features as latent vectors that share the same latent space as the one for users and items, and does this in a single computation stage.
2) In a performance comparison of FeatureMF against three established competing CF models which exploit item features information, it is shown to yield better recommendation performance. It is also shown to do better in alleviating the data-sparsity and cold-start problems, and to be approximately linearly scalable.

The remainder of this paper is organized as follows. Section II covers relevant related work, introducing the MF technique and various published approaches taken to incorporating additional information into the recommender scheme. Section III introduces the preliminary knowledge utilized in this work. The proposed recommendation model, FeatureMF, is presented in Section IV. Section V presents the performance comparison experiments and their analysis. Finally, our conclusions and suggestions for future research directions are summarized in Section VI.

## II. RELATED WORK

### A. COLLABORATIVE FILTERING ENRICHED WITH SIDE INFORMATION

CF has achieved success in recommender systems because it only requires user-item interactions to make recommendations [6]. However, CF approaches usually are challenged in the performances they yield in *data-sparsity* and *cold-start* situations. In these circumstances, many works have proposed to adopt information from additional sources, also known as side information, to improve recommendation performance [5], [6]. For example, side information that may be extracted via social networks [9], [10], [21], user demographics [11], [12] or reviews from users [13], [14], has been successfully incorporated into CF and has been proven helpful at improving recommendation performance. In terms of approaches that exploit side information, works such as [18]–[20], [22], [23] propose the construction of a heterogeneous information network (HIN) based on user/item side information in order to learn relations between user/items. Other works, taking a different approach, i.e. [24]–[26], propose the use of deep learning models for learning latent features from side information of users and items. Recommendation models using deep learning techniques have become a particularly popular area of research in recent years, following their success in multiple application domains such as computer vision and natural language processing (NLP) [27]. However, recent results have pointed out that models based on deep learning are not as strong as expected while also being computationally complex [28], [29]. Matrix Factorization (MF) is still among the most popular CF methods that allow integration of additional information [5], [6], [18], [19].

MF consists in factorizing the user-item rating matrix into two low-rank matrices, which present users and items, respectively. The basic form, along with variants of that form, of the MF model popular among Netflix Prize contestants [8], [30] is regularized singular value decomposition, RegSVD, proposed by Paterek in [31] in 2007 and by others around the same time. The SVD concept quickly became the core for many other published models, such as NMF [32], PMF [33] and BPMF [34], which still today enjoy standing as reference baselines for current MF-based recommendation research and hybrid models that make use of side information related to users/items that achieve further improvements, such as [9], [20].

### B. MATRIX FACTORIZATION ENRICHED WITH ITEM FEATURES

Item features, which are generated by domain experts to represent item characteristics [17], have been utilized with different techniques proposed for combining them with MF to improve the recommendation performance. For instance, Nguyen and Zhu [16] propose the incorporation of content information into MF, whereby the similarity between items, utilizing the 'Simple Matching Similarity' measurement, is first computed, followed by extending the MF framework with the computed similarities. Yu *et al.* [17] propose a similar enhanced MF model, where item similarities are calculated by a more accurate measure, 'Coupled Object Similarity.' These MF-based approaches first calculate item similarity and then incorporate the pre-processed similarity into the basic MF models as a regularization term to ensure that item-latent feature vectors are close to those of similar items. More recently, several works report modeling item features as a HIN, and incorporate results from a HIN into MF to address the *data-sparsity* and *cold-start* problems, and yield improved performances on [16] and [17]. Shi *et al.* [18] propose a semantic path based personalised recommendation model, SemRec, that computes entity similarity in a HIN on weighted meta-paths. In [19], Zheng *et al.*, in their model DSR, combine similarities between users and items calculated from a HIN with MF using dual similarity regularization, which can impose a constraint on users and items with either low or high similarities. Shi *et al.* [20], in their model HERec, used a scheme to learn embeddings for users and items from HINs, which are then integrated with MF.

Most of the aforementioned item features enriched models involve a two stage process, whereby intermediate results (e.g. item similarities) are first calculated based on item features, which are then incorporated in the second stage into MF in various kinds of forms. However, there is no guarantee that the intermediate results are accurate, which observation is reflected in the variety of similarity measures proposed, with each seeming to propose a different measure. Hence, it is unclear which is the best one. Meanwhile, some of the similarity measures have limitations for features, for example, similarity measures proposed in Nguyen and Zhu *et al.* [16] and Yu *et al.* [17], can only deal with the situation that, for each item, every feature can only contain one attribute value. This characteristic of the scheme constrains its use in performance comparison experiments such as those reported on here below. This does not apply to the latter three, so in this paper we compare the performance of our model with these in Section V.

In this work, the novel item feature enriched MF model proposed incorporates item information directly into the MF framework, without a pre-processing stage, and allows for one or more attributes of each item feature.

## III. PRELIMINARY KNOWLEDGE

In this section, the preliminary knowledge related to the proposed item feature enriched recommendation model is introduced. First, the notations used in the remainder of this paper are presented, and then a brief introduction to MF is given.

### A. NOTATIONS AND NOMENCLATURE

A typical recommendation scenario involves a set of $m$ users $U = \{u_1, u_2 \ldots u_m\}$, a set of $n$ items $I = \{i_1, i_2 \ldots i_n\}$, and their interactions represented by a rating matrix $R \in \mathbb{R}_{m,n}$. Typically, $R$ is very sparse, since the observed number of user-item interactions is much smaller than $m \times n$. The set of $(u, i)$ pairs with observed ratings is denoted by $K$.

In the case when item side information is available, each item $i \in I$ can be represented by a set of item **features** $F = \{F_1, F_2 \ldots, F_N\}$, where $N$ is the number of features. These features are extracted from item content information, and are predefined based on the item domain. For example, a movie item can be represented by features such as *Director, Actor, Type*, etc. In addition, each feature has categorical values, referred to as **attributes** in this work, e.g., *Action, Crime, Drama*, etc. for the *Type* feature. The set of attributes belonging to an item $i \in I$ for each feature $F_t$ is denoted as $F_t(i)$. Table 1 demonstrates a structured representation for a movie instance using its content information.

**TABLE 1.** Content information structure for the movie instance *The Godfather.*

| Feature | Attribute |
|---|---|
| Director | Coppola |
| Actor | Brando, De Niro, Pacino |
| Type | Action, Crime, Drama |

The nomenclature used in this paper is presented in Table 2.

### B. MATRIX FACTORIZATION

The goal of MF is to map both users and items into the same low-rank latent factor space by approximating the observed ratings, where users and items are represented by a set of feature vectors, $P \in \mathbb{R}_{m,d}$ and $Q \in \mathbb{R}_{n,d}$, respectively, where $d$ is the number of latent factors, $\mathbb{R}_{m,d}$ is an $m \times d$ matrix and $\mathbb{R}_{n,d}$ is an $n \times d$ matrix. The predicted rating of item $i$ by user $u$ is computed as:

$$\tilde{r}_{ui} = P_u Q_i^T = \sum_{f=1}^{d} P_{u,f} Q_{i,f}^T \tag{1}$$

**TABLE 2. Nomenclature.**

| Symbol | Definition and/or Description |
|---|---|
| $U$ | the set of users $U = \{u_1, u_2...u_m\}$ |
| $I$ | the set of items $I = \{i_1, i_2...i_n\}$ |
| $r_{ui}$ | True rating of item $i$ by user $u$. |
| $\tilde{r}_{ui}$ | Predicted rating of item $i$ by user $u$. |
| $K$ | Set of observed ratings. |
| $m$ | Total number of users. |
| $n$ | Total number of items. |
| $d$ | Number of latent factors. |
| $P_u/Q_i$ | Latent factor for user $u$ / item $i$. |
| $BU_u/BI_i$ | Bias for user $u$ / item $i$. |
| $\mu$ | Global average rating. |
| $F$ | Item feature set. |
| $N$ | Total number of item features. |
| $t$ | Index of item features. |
| $F_t(i)$ | Set of attributes of feature $F_t$ for item $i$. |
| $\|F_t\|$ | Number of all possible attributes of feature $F_t$ over all items. |
| $y_t$ | $\|F_t\| \times d$ matrix characterizing the latent factor representation of all attributes of feature $F_t$. |
| $y_t(a)$ | $d$-dimensional vector representing attribute $a$ of feature $F_t$. |
| $\lambda$ | Regularization parameter. |

where row $P_u$ in $P$, and row $Q_i$ in $Q$ represent the vector representation of user $u$ and item $i$, respectively.

For the rating prediction task [6], the user and item latent feature matrices are usually found by minimizing an MF objective function $O_{MF}$, which is the regularized squared error on the observed ratings, as follows in (2):

$$O_{MF} = \sum_{(u,i)\in K} \left((r_{ui} - \tilde{r}_{ui})^2 + \lambda(\|P_u\|^2 + \|Q_i\|^2\right) \qquad (2)$$

where $\|\cdot\|^2$ is the Frobenius norm [35].

## IV. MATRIX FACTORIZATION ENRICHED WITH ITEM FEATURES

In this section, the proposed item feature enriched MF model is described, in which item features are incorporated directly into the MF framework.

### A. THE FEATUREMF MODEL

The assumption of FeatureMF is that the predicted rating for user $u$ to item $i$ is not only related to latent representations of $u$ and $i$, but also relies on the feature attributes of item $i$. In line with this assumption, we propose to project attributes of item features into the same latent factor space as the one for users and items, so as to incorporate attribute representations of items into the MF framework. Let $\{y_t | y_t \in \mathbb{R}_{|F_t|\times d}, t = 1, 2 \ldots, N\}$ be the set of latent-factor matrices representing each of the item features, where each row $y_t(a)$ in $y_t$ is the vector representation of an attribute belonging to $F_t$ and $|F_t|$ is the number of attributes representing each feature. Figure 1 illustrates how item features are incorporated into the item rating prediction of FeatureMF, as per (3) below, i.e., to jointly represent item representations. As shown there, in Figure 1, each attribute $a$ in each of the features $F_t$ (e.g., attribute *Action* in feature *Type* from Table 1) is represented by a latent factor vector in the same latent space as the item factors. For any item $i$, its final representation using FeatureMF consists of two parts: (a) the item-specific latent factors $Q_i$, and (b) sum of the average representations of the attributes in each of the features that item $i$ possesses $\sum_{t=1}^{N} |F_t(i)|^{-1} \sum_{a\in F_t(i)} y_t(a)$. As an example, the movie *The Godfather* in Table 1 may be considered. According to FeatureMF, the movie representation learned from the ratings is augmented by the vector representation of its *Director*, i.e., *Coppola*, average vector representation of its *Actors*, i.e., *Brando, De Niro, Pacino* and average vector representation of its *Type*, i.e., *Action, Crime, Drama*.

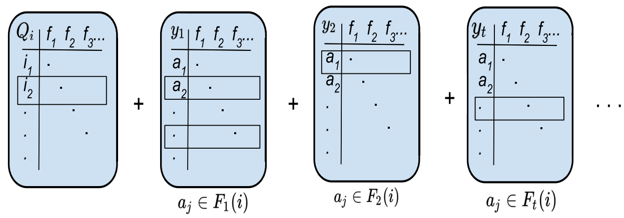



**FIGURE 1.** An illustration of item representation for FeatureMF.

Hence, FeatureMF, with this item-feature influencing term, can predict the rating, $\tilde{r}_{ui}$, of item $i$ by user $u$ as follows:

$$\tilde{r}_{ui} = P_u^T(Q_i + \sum_{t=1}^{N} |F_t(i)|^{-1} \sum_{a\in F_t(i)} y_t(a)) + b_{ui} \qquad (3)$$

where $b_{ui} = \mu + BU_u + BI_i$, $\mu$ is the global average rating, $BU_u$ and $BI_i$ indicate the user and item bias.

Clearly, also as (3) implies, even in a cold-start situation, when an item has received only a few ratings, FeatureMF is capable of computing the rating prediction only based on the item features, thereby alleviating the cold-start *item* problem.

In the literature, CF methods considering item features, e.g. [16]–[20], are also proven to be effective in improving the recommendation accuracy and dealing with cold-start problems. However, as indicated earlier, they typically work in two stages: first utilize item information to estimate item similarities or item embeddings, the result of which is then combined with MF. The computational complexity of the first stage is usually quadratic with respect to the number of items [16], [17]. Unlike the methods mentioned above, as FeatureMF considers the influence of item ratings and item features in one stage, it is less computationally complex and scales linearly with the dataset size, which is an advantage over some of these competing models, especially the best

**TABLE 3.** Computational complexity comparison between HERec and FeatureMF.

| Model | Computational Complexity |
|---|---|
| HERec | $\mathcal{O}(\lvert P\rvert \cdot D\cdot(\lvert U\rvert+\lvert J\rvert))+\mathcal{O}(\lvert K\rvert\cdot\lvert P\rvert\cdot D\cdot d)$ |
| FeatureMF | $\mathcal{O}(\lvert K\rvert \cdot N\cdot \overline{a}\cdot d)$ |

**Algorithm 1:** Learning the FeatureMF Model

**Input** : $R, \lambda, \alpha, \gamma$ *(learning rate)*, $d$,$c$, $iter \leftarrow 0$
**Output:** rating predictions $\tilde{r}_{ui}$

Initialize the low-rank matrices for users ($P$), items ($Q$) and features ($Y = \{y_t | t = 1..N\}$), the bias vectors for users ($BU$) and items ( $BI$)
**while** *iter* $<$ *maxIter* ***or*** *error on validation set decrease* **do**
  **while** $(u, i) \in K$ **do**
    $\tilde{r}_{ui} \leftarrow P_u^T(Q_i + \sum_{t=1}^{N} |F_t(i)|^{-1} \sum_{a\in F_t(i)} y_t(a)) + b_{ui}$
    $e_{ui} \leftarrow \tilde{r}_{ui} - r_{ui}$
    $BU_u \leftarrow BU_u - \gamma(e_{ui} + \lambda \cdot BU_u)$
    $BI_i \leftarrow BI_i - \gamma(e_{ui} + \lambda \cdot BU_i)$
    $P_u \leftarrow P_u - \gamma(e_{ui}(Q_i + \sum_{t=1}^{N} |F_t(i)|^{-\alpha} \sum_{a\in F_t(i)} y_t(a)) + \lambda \cdot P_u)$
    $Q_i \leftarrow Q_i - \gamma(e_{ui}P_u + \lambda \cdot Q_i)$
    **foreach** $y_t \in Y$ **do**
      **foreach** $a \in F_t(i)$ **do**
        $y_t(a) \leftarrow y_t(a) - \gamma(e_{ui}|F_t(i)|^{-1}P_u + \lambda \cdot y_t(a))$
      **end**
    **end**
  **end**
  *iter* $\leftarrow$ *iter* $+ 1$
**end**

of these, HERec, [20]. This is revisited in the following ‘complexity analysis’ subsection C and in the final section of our experimental analysis below.

### B. OPTIMIZATION

Parameters in FeatureMF are learned by minimizing the regularized objective function:

$$O_{FeatureMF} = \sum_{(u,i)\in K} \Big( \big(r_{ui} - \tilde{r}_{ui}\big)^2 + \lambda\big(\|P_u\|^2 + \|Q_i\|^2 + BU_u^2 + BI_i^2 + \sum_{t=1}^{N} \sum_{a\in F_t(i)} \|y_t(a)\|^2\big)\Big) \tag{4}$$

where, as before, $\|\cdot\|^2$ is the Frobenius norm [35].

Since the optimization problem in (4) is biconvex, stochastic gradient descent (SGD) [36] is utilized to search for a local minimum. For a given instance $r_{ui}$ in the training dataset, parameters with opposite direction of the gradient are updated, and loop over all observed ratings in $K$. The pseudocode for learning the FeatureMF model is shown as Algorithm 1.

### C. COMPLEXITY ANALYSIS

Most of the training time is spent on the computation of the objective function and its gradients against different feature vectors. To draw out the analysis of this for FeatureMF, let the number of observed ratings be $|K|$, while $d$ denotes the number of latent factors of the low-rank matrices. Then the worst-case computational complexity for minimizing the objective function is $\mathcal{O}(|K|dN\overline{a})$, where $\overline{a}$ is the average number of attributes an item has for a feature. The costs of computing the gradients for $BU_u$, $BI_i$, $P_u$, $Q_i$ and $y_t(a)$ are $\mathcal{O}(|K|)$, $\mathcal{O}(|K|d)$, $\mathcal{O}(|K|ld)$ and $\mathcal{O}(|K|dN\overline{a})$, respectively. Thus, the overall computational complexity of one iteration of FeatureMF is $\mathcal{O}(|K|dN\overline{a})$.

Since FeatureMF is designed to deal with situations involving small number of features (e.g., $N = 3$ and $N = 2$ for the two datasets we used), and taking into account that $\overline{a}$ is relatively small for a typically sparse rating matrix, the overall computational complexity of FeatureMF is then effectively linear with respect to the total number of observed ratings.

We compare this with HERec, which is the key leading edge competitor considered in our experiments below. As stated in [20], the computation of HERec contains two major parts implemented in a two step process: (1) HIN embedding, the complexity of which is $\mathcal{O}(|P|\cdot D\cdot(|U|+|J|))$, where $|P|$ is the number of meta-paths, $|U|$ and $|J|$ are number of users and items, and $D$ is the embedding dimension; (2) matrix factorization, the complexity of which for each triplet $< u, i, r >$ is $\mathcal{O}(|P| \cdot D \cdot d)$, where $d$ is the number of latent factors. Hence, due to this first process step, HERec complexity and scaling will tend to be quadratic.

Table 3 seeks to capture this computational complexity comparison between both models. As is evident here again, in avoiding the first stage of HERec learning HIN embeddings, a benefit for FeatureMF’s is reduced complexity. Compared to the second stage of HERec, FeatureMF has similar computational complexity depending on the dataset used and customized hyperparameters. Further, in HERec, the number of meta-paths $P$ used and the selection of meta-paths are non-trivial problems and need significant effort to tune. While in FeatureMF, $N$ is a constant based on the dataset used. In summary, by comparison with HERec, FeatureMF is less complex, scales approximately linearly and, hence, will have a faster execution time.

## V. EXPERIMENTS

In this section, we report on comparative performance tests through experiments we conducted using two benchmark datasets. The goal was to test the effectiveness of FeatureMF compared to four established traditional models and three other recent state-of-the-art models. As indicated above, the former are called ‘traditional’ in the sense that they make recommendations using only user-item ratings, and are sometimes treated as reference models for benchmarking other models; whereas the latter three include additional information, besides the ratings, in their algorithms.

**TABLE 4.** Statistics of the two public datasets utilized in the experiments.

| Dataset | Relations (A - B) | Number of A | Number of B | Number of A-B | Avg. degrees of A | Avg. degrees of B |
|---|---|---|---|---|---|---|
| Douban | User-Movie | 13,367 | 12,677 | 1,068,278 | 79.9 | 84.3 |
| | Movie-Director | 10,179 | 2,449 | 11,276 | 1.1 | 4.6 |
| | Movie-Actor | 11,718 | 6,311 | 33,587 | 2.9 | 5.3 |
| | Movie-Type | 12,676 | 38 | 27,668 | 2.9 | 728.1 |
| Yelp | User-Business | 16,239 | 14,284 | 198,379 | 12.2 | 13.9 |
| | Business-City | 14,267 | 47 | 14,267 | 1.0 | 303.6 |
| | Business-Category | 14,180 | 511 | 40,009 | 2.8 | 78.3 |

### A. SETTINGS

This section presents the two well known public datasets used in the experimental work as well as the metrics employed.

#### 1) DATASETS

The first is the Douban dataset, created by Shi *et al.* [19], [20], [37] from Douban,[1] a popular social media platform in China. It provides user ratings and reviews for movies, books, and music. The utilized dataset was collected from 13,367 users with a total of 1,068,278 user ratings (ranging from 1 to 5) of 12,677 movies. The second is the Yelp[2] challenge dataset, which includes 198,397 user ratings (ranging from 1 to 5) of 14,284 local businesses from 16,239 users. Both datasets contain not only the user ratings data, but also content information related to both users and items. Since the assumption in this paper is that only user ratings data and item features are available, only information related to items was selected in both datasets for the experiments. The relevant parameters being considered from these two datasets are summarized in Table 4. There, it may be observed that the rating matrices of both datasets are sparse, and the density of Douban dataset (0.63%) is higher than that of Yelp dataset (0.08%).

#### 2) EVALUATION METRICS

Two popular error-based metrics were used in the evaluation - the mean absolute error (MAE) and the root mean square error (RMSE) [38], [39], which are defined in (5) and (6), respectively:

$$MAE = \frac{1}{|Z|} \sum_{(u,i)\in Z} |r_{ui} - \tilde{r}_{ui}| \tag{5}$$

$$RMSE = \sqrt{\frac{1}{|Z|} \sum_{(u,i)\in Z} (r_{ui} - \tilde{r}_{ui})^2} \tag{6}$$

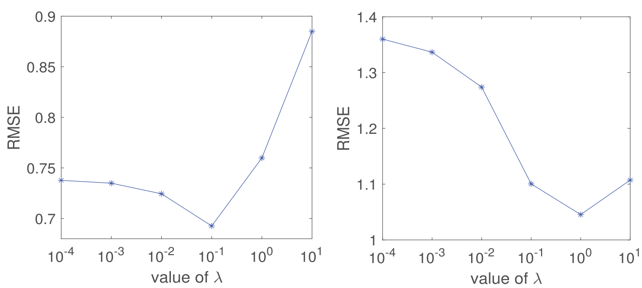
where $Z$ is the set of observed ratings in the test set. The smaller MAE and RMSE are, the higher the predictive power of the model, [40].

[1] http://movie.douban.com/

[2] http://www.yelp.com/dataset challenge/

### B. PERFORMANCE OF FEATUREMF

The proposed model is implemented on the top of LibRec,[3] a popular Java library for developing recommender systems. In order to study the hyperparameters' impact on FeatureMF, we use five-fold cross validation for training and testing on both datasets. Specifically, each dataset is randomly split into five folds, for each iteration, one fold is used as test set and the remaining four folds are used for training. We conduct five iterations to ensure each fold is tested and report the average results. As LibRec uses a learning rate decay technique based on [41], [42], the initial value of the learning rate has limited influence on the model performance. We set it to 0.01 in our experiments, which is a common setting for MF-based methods. Following common parameter settings in other item features enriched MF models [18]–[20], we first set the number of latent factors to 10. Then, we tune the value of the regularization parameter $\lambda$ to a value in the set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$. As the results in Figure 2 show, the best performance of FeatureMF is achieved when $\lambda = 0.1$ on Douban and $\lambda = 1$ on Yelp. These $\lambda$ values are used in the rest of the experiments.

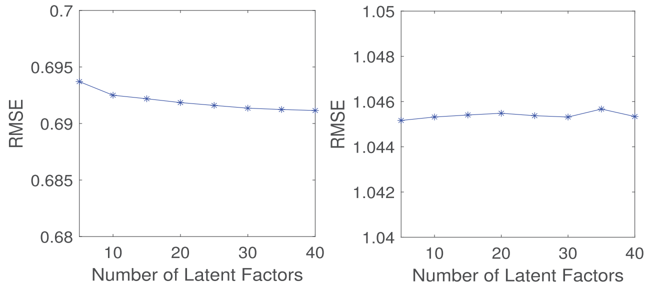



(a) Douban (b) Yelp

**FIGURE 2.** RMSE performance of FeatureMF as a function of the regularization parameter $\lambda$ for the (a) Douban and (b) Yelp datasets with the number of latent factors set to 10.

**Latent Factor sensitivity:** One of the most important hyperparameters of MF-based models is the number of latent factors and the sensitivity of the model's performance to this number. In order to examine the dimension impact of FeatureMF, we vary it from five to forty in steps of five.

[3] https://guoguibing.github.io/librec/index.html

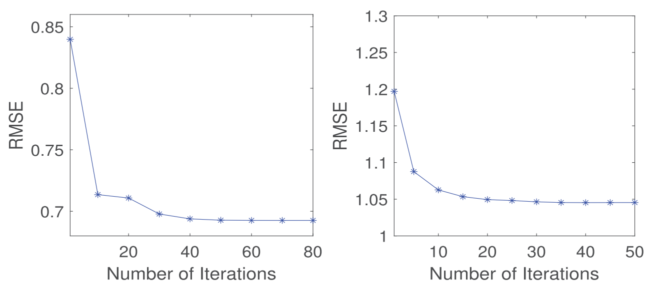



(a) Douban (b) Yelp

**FIGURE 3.** Impact of number of latent factors on the RMSE performance of FeatureMF in the two datasets.

As shown in Figure 3, the values of RMSE for different number of latent factors are relatively close, with standard deviation of 8.5e-4 on Douban and 1.5e-4 on Yelp. This demonstrates the robustness of FeatureMF to variations in feature dimensionality, where the features in our case are both ratings and item features in a unified model [9].

**Performance convergence:** FeatureMF's performance convergence as a function of the number of iterations is here considered. With the number of latent factors set to 10, the convergence behaviour for the relatively dense (Douban) and sparse (Yelp) datasets is shown in Figure 4. Convergence can be seen to be exponential and rapid and quite similar to results obtained by others, e.g, [20]. For the Douban dataset, best performance is achieved at 40 or more iterations and, for the Yelp dataset, at 20 or more. The differences likely relate to the relative denseness/sparseness of the datasets.




(a) Douban (b) Yelp

**FIGURE 4.** FeatureMF's performance convergence as a function of the number of iterations for the (a) Douban and (b) Yelp datasets with the number of latent factors set to 10.

### *C. COMPARATIVE STUDY*

We compare FeatureMF with three recent state-of-the-art MF-based models which make use of item information, albeit in a different way to FeatureMF. These models are:

- **SemRec** [18]: a semantic path based personalized recommendation model, which integrates MF with HINs using weighted meta-paths to obtain the prioritized and personalized user preferences in paths.
- **DSR** [19]: a MF-based recommendation model with a dual similarity regularization, which utilizes multiple types of information in a HIN to impose constraints on user- and item latent factors.
- **HERec** [20]: a heterogeneous information embedding based model, which integrates a fused HIN embedding, learned by using meta-path based algorithms with an MF model.

As may be observed and as briefly indicated earlier, these are all HIN-based models and use a two stage computation process. First, intermediate results (similarity or embedding) are computed from a HIN using meta-path based algorithms and then the results are integrated within the MF framework.

As [20] shows, HERec has a better performance track-record than SemRec and DSR. Hence, we focus on it particularly in our comparative experiments. Thus, we follow the HERec 'training: test' dataset split ratios, i.e., with reference to the training part of the ratio, we experimented with 80%, 60%, 40%, and 20% of the data being used for training on the Douban dataset, and 90%, 80%, 70%, and 60% of the data for training on the Yelp one. Lower training percentages are not used on Yelp as, according to [20], the dataset is much sparser. For each data split ratio, we randomly split the dataset five times, and average the results as the final performance.The number of latent factors was set to 10 for all MF-based models for fair comparison.

The performance results for the SemRec, DSR and HERec models for this experimental set-up are taken from [20], where it is speculated that the likely reason for HERec's better performance is because it considers not only the latent factor representation for users and items learned from MF but also HIN embeddings for them learned from HINs.

As a benchmarking exercise, we also compare FeatureMF's performance with that of four traditional reference models, **RegSVD** [8], [31] **NMF** [32], **PMF** [33], and **BPMF** [34], which are frequently used to benchmark new MF models but which do not exploit item side information. These four models are obtained using the LibRec Library [43]. As would be expected, of course, FeatureMF's performance is much better.

Results for all seven models set against those of our FeatureMF model are presented in Table 5, which also includes FeatureMF's percentage performance improvement (Perf-Imp) over the other models on both the MAE and RMSE metrics. This latter is graphically portrayed in Figure 5.

Key observations which may be noted from these results include:

1) While clearly it may be observed that FeatureMF achieves better recommendation performance than all other seven models, focusing in on HERec [20], the next best performing model, FeatureMF may be seen to achieve an MAE performance improvement of approximately 1% to 2% on the Douban dataset and 3% to 7% on the Yelp dataset over the eight test scenarios. (The actual figures in Table 5 are 0.94% to 2.07% and 3.40% to 6.59% resp.) For the RMSE measure,

**TABLE 5.** Comparative evaluation of the performance of FeatureMF against seven other well-known models. Three, SemRec, DSR and HERec, consider additional item features information and would be directly competitor models. The other four models, RegSVD, NMF, PMF and BPMF, are traditional benchmarking ones which do not use item feature side information. Comparisons are on the (a) Douban and (b) Yelp datasets over four 'training: test' ratio scenarios each, indicated by the percentage 'Training' parameter. The number of latent factors, $d$, was set to 10. 'Perf-Imp' figures are FeatureMF's percentage of performance improvement on the MAE and RMSE performance metrics over the other models. (Note: FeatureMF's optimised $\lambda$ values used are 0.1 on Douban and 1.0 on Yelp).

(a) Douban

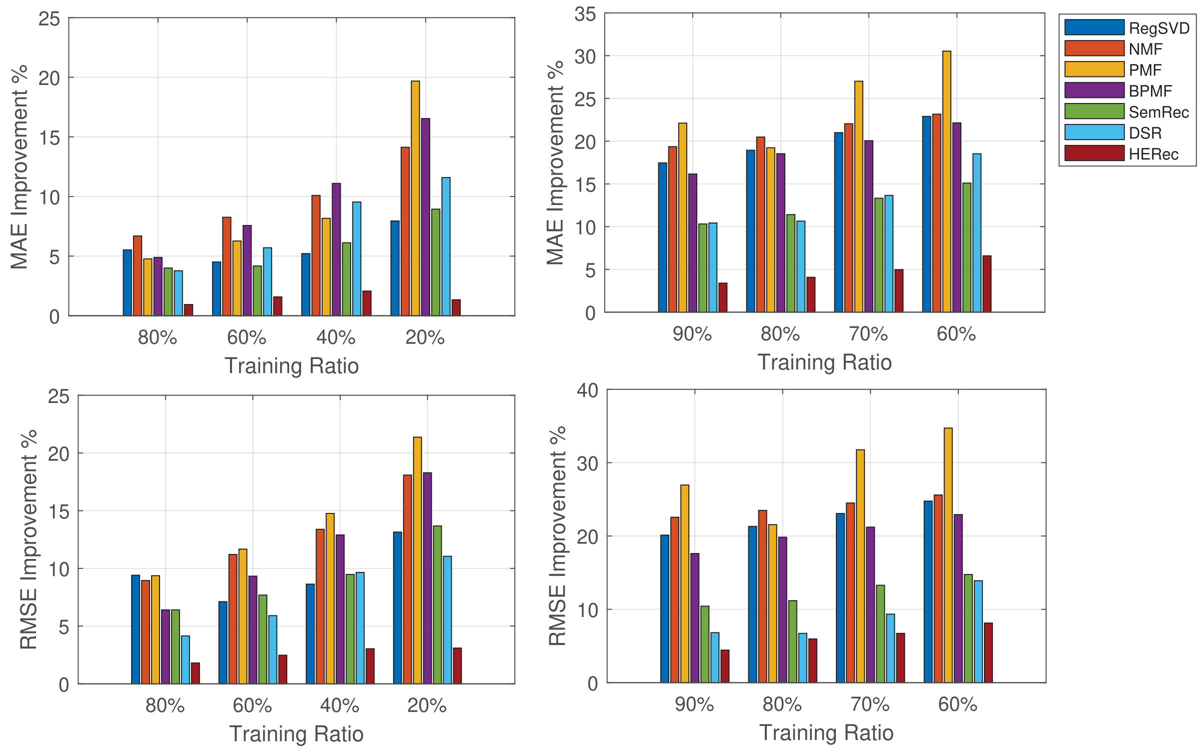
| Training | Metrics | Models without item features | | | | Models with item features | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | RegSVD | NMF | PMF | BPMF | SemRec | DSR | HERec | FeatureMF |
| 80% | MAE | 0.5787 | 0.5859 | 0.5741 | 0.5748 | 0.5695 | 0.5681 | 0.5519 | **0.5467** |
| | **Perf-Imp** | **5.53%** | **6.69%** | **4.77%** | **4.89%** | **4.00%** | **3.77%** | **0.94%** | |
| | RMSE | 0.7644 | 0.7606 | 0.7641 | 0.7399 | 0.7399 | 0.7225 | 0.7053 | **0.6925** |
| | **Perf-Imp** | **9.41%** | **8.95%** | **9.37%** | **6.41%** | **6.41%** | **4.15%** | **1.81%** | |
| 60% | MAE | 0.5759 | 0.5994 | 0.5867 | 0.5949 | 0.5738 | 0.5831 | 0.5587 | **0.5499** |
| | **Perf-Imp** | **4.51%** | **8.26%** | **6.27%** | **7.56%** | **4.17%** | **5.69%** | **1.58%** | |
| | RMSE | 0.7505 | 0.7851 | 0.7891 | 0.7688 | 0.7551 | 0.7408 | 0.7148 | **0.6970** |
| | **Perf-Imp** | **7.13%** | **11.22%** | **11.67%** | **9.34%** | **7.69%** | **5.91%** | **2.49%** | |
| 40% | MAE | 0.5888 | 0.6208 | 0.6078 | 0.6278 | 0.5945 | 0.6170 | 0.5699 | **0.5581** |
| | **Perf-Imp** | **5.21%** | **10.10%** | **8.18%** | **11.10%** | **6.12%** | **9.55%** | **2.07%** | |
| | RMSE | 0.7764 | 0.8189 | 0.8321 | 0.8143 | 0.7836 | 0.7850 | 0.7315 | **0.7093** |
| | **PerfImp** | **8.64%** | **13.38%** | **14.76%** | **12.89%** | **9.48%** | **9.64%** | **3.03%** | |
| 20% | MAE | 0.6323 | 0.6779 | 0.7247 | 0.6974 | 0.6392 | 0.6584 | 0.5900 | **0.5821** |
| | **Perf-Imp** | **7.94%** | **14.13%** | **19.68%** | **16.53%** | **8.94%** | **11.59%** | **1.34%** | |
| | RMSE | 0.8547 | 0.9062 | 0.9440 | 0.9084 | 0.8599 | 0.8345 | 0.7660 | **0.7423** |
| | **Perf-Imp** | **13.15%** | **18.09%** | **21.37%** | **18.28%** | **13.68%** | **11.05%** | **3.09%** | |

(b) Yelp

| Training | Metrics | RegSVD | NMF | PMF | BPMF | SemRec | DSR | HERec | FeatureMF |
|---|---|---|---|---|---|---|---|---|---|
| 90% | MAE | 0.9825 | 1.0057 | 1.0412 | 0.9673 | 0.9043 | 0.9054 | 0.8395 | **0.8129** |
| | **Perf-Imp** | **17.46%** | **19.36%** | **22.11%** | **16.16%** | **10.32%** | **10.43%** | **3.40%** | |
| | RMSE | 1.3049 | 1.3459 | 1.4268 | 1.2652 | 1.1637 | 1.1186 | 1.0907 | **1.0455** |
| | **Perf-Imp** | **20.12%** | **22.56%** | **26.95%** | **17.61%** | **10.43%** | **6.82%** | **4.44%** | |
| 80% | MAE | 1.0029 | 1.0223 | 1.0065 | 0.9978 | 0.9176 | 0.9098 | 0.8475 | **0.8153** |
| | **Perf-Imp** | **18.94%** | **20.48%** | **19.23%** | **18.53%** | **11.41%** | **10.65%** | **4.08%** | |
| | RMSE | 1.3286 | 1.3664 | 1.3327 | 1.3040 | 1.1771 | 1.1208 | 1.1117 | **1.0455** |
| | **Perf-Imp** | **21.31%** | **23.49%** | **21.55%** | **19.83%** | **11.18%** | **6.72%** | **5.96%** | |
| 70% | MAE | 1.0319 | 1.0457 | 1.1170 | 1.0198 | 0.9407 | 0.9443 | 0.8580 | **0.8153** |
| | **Perf-Imp** | **21.00%** | **22.04%** | **27.01%** | **22.06%** | **13.33%** | **13.66%** | **4.98%** | |
| | RMSE | 1.3648 | 1.3908 | 1.5387 | 1.3326 | 1.2108 | 1.1582 | 1.1256 | **1.0500** |
| | **Perf-Imp** | **23.07%** | **24.50%** | **31.76%** | **21.21%** | **13.28%** | **9.34%** | **6.72%** | |
| 60% | MAE | 1.0613 | 1.0648 | 1.1778 | 1.0509 | 0.9637 | 1.0043 | 0.8759 | **0.8182** |
| | **Perf-Imp** | **22.90%** | **23.16%** | **30.53%** | **22.14%** | **15.10%** | **18.53%** | **6.59%** | |
| | RMSE | 1.4027 | 1.4183 | 1.6167 | 1.3692 | 1.2380 | 1.2257 | 1.1488 | **1.0554** |
| | **Perf-Imp** | **24.76%** | **25.59%** | **34.72%** | **22.92%** | **14.75%** | **13.90%** | **8.13%** | |

the performance improvements have a generally quite similar pattern. Intuitively, this effectiveness of FeatureMF could be attributed to the fact that it learns the latent factor representations for item attributes directly without the need of an intermediate computation stage incorporating a similarity measure, –a measure for which there are differences of views as to what should be used, as well as constraints associated with different measures, as indicated earlier.

2) While, as expected, the models, SemRec, DSR, and HERec along with FeatureMF, which consider additional item features, consistently achieve better performance than the traditional MF-based reference models which do not, such as the well-known RegSVD, NMF, PMF, and BPMF chosen here, it may be observed that this happens to a greater degree on the Yelp dataset. This indicates the more special recommendation improvement benefit the use of

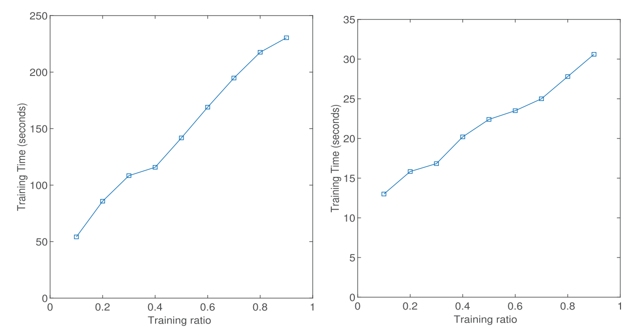



(a) Douban (b) Yelp

**FIGURE 5.** FeatureMF's performance improvement over that yielded by other models for the (a) Douban an (b) Yelp datasets.

additional item features brings to sparser datasets. FeatureMF, performing best here, boosts MAE performances, depending on the percentage training, by 4.5% to 19.7% for the Douban dataset and from 16.2% to 30.5% for the data-sparse Yelp dataset. The RMSE improvement varies similarly.

3) FeatureMF performs well in the case of cold-start prediction, where there are very few rating records. This can be concluded from the recommendation results of our experiments with various item cold-start degrees, i.e. the rating sparsity, [44]. Analysis of the data sets shows that the percentage of items in each dataset having 5 or less numbers of ratings (and so by our definition, classed as a cold-start item) is considerable; it may be shown to be a little more than 38% in the Douban dataset and 55% in the Yelp data set. As shown in Figure 5, the improvement ratio grows as the percentage of data used for training drops and hence the item cold-start challenge increases. For instance, the percentage improvement of FeatureMF over the traditional models on the Yelp dataset, which is sparser than Douban one and thus more challenging from a cold-start alleviation perspective, may be seen, for the MAE metric, to range from over 16% for the 90% training scenario to just over 30% for the 60% training scenario. The results indicate that FeatureMF is able to utilize item features more effectively to make better recommendations.




(a) Douban (b) Yelp

**FIGURE 6.** FeatureMF's near-linear scalability attribute shown across both datasets ($d = 5$), with the Intel i7-4500u CPU computational training time measured in seconds.

### *D. COMPUTATIONAL COMPLEXITY AND DATASET SCALING*

From the analysis in Section IV-C, the computational complexity of the training of a FeatureMF model was shown to increase approximately linearly with the number of observed ratings, leading to a near linear increase in required computational time. This analytical prediction of the scaling attribute of FeatureMF, was investigated by training the model on datasets of different sizes.

With the number of latent factors set to five ($d = 5$), the experiment consisted in using different training data to test data ratio settings on both datasets, ranging from 0.1 to 0.9, in steps of 0.1. The training data itself was randomly selected from the user-item ratings to form the training set. This was repeated ten times for each ratio scenario and the average FeatureMF training times obtained. These times are graphed in Figure 6 for both datasets. The model was programmed in Java and run on an Intel i7-4500u CPU. It may be seen there that, as predicted above in Section IV-C, the training time grows almost linearly with the size of training data.

## VI. CONCLUSION

A novel MF model, FeatureMF, has been proposed and set out in this paper. In a new way, it exploits item features information where available to improve recommendation performance. This novelty primarily lies in the way it diffuses the item latent factors, derived for example from global ratings, with latent factor representations of attributes of item features into a single MF computational stage. This is quite a departure from the two-step process of other recent state-of-the-art MF models, e.g., the HERec, DSR and SemREC models, whereby intermediate results such as item similarities are computed from the item features information in a first step before the MF model proper is run.

It is shown to yield an MAE recommendation performance improvement compared to the next best model of its kind, HERec, of over 0.9% to over 6.5% across all contexts of the two datasets examined (Douban and Yelp) including those of higher data sparsity. It does this with relatively reduced complexity and hence computation resource requirement, with this scaling approximately linearly with dataset size. HERec, which itself was a significant advance on its two main rivals, DSR and SemREC, has a near quadratic scaling attribute.

FeatureMF, which yields better recommendation performance than HERec, its nearest competitor, and better than six other established CF recommendation models, was shown to contribute to an improved alleviation of the data-sparsity and cold-start problems, to be scalable, and to be applicable to any recommender system with even only limited *item* information available.

As a future work, it is planned to exploit richer sources of item information, which are not limited to categorical information as used in this work. Possible resources could be item reviews, where various kinds of natural language processing (NLP) techniques can be used to extract value information related to items. In addition, it will be also interesting and natural to investigate the way of combining item features with deep learning based models, such as auto-encoders.

## REFERENCES

[1] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: Scalable online collaborative filtering," in *Proc. 16th ACM Int. Conf. World Wide Web*, New York, NY, USA, May 2007, p. 271–280.

[2] R. Xiong, J. Wang, N. Zhang, and Y. Ma, "Deep hybrid collaborative filtering for Web service recommendation," *Expert Syst. Appl.*, vol. 110, pp. 191–205, Nov. 2018.

[3] Z. Cui, X. Xu, F. Xue, X. Cai, Y. Cao, W. Zhang, and J. Chen, "Personalized recommendation system based on collaborative filtering for IoT scenarios," *IEEE Trans. Services Comput.*, vol. 13, no. 4, pp. 685–695, Jul. 2020.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[5] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1–45, Jul. 2014.

[6] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013.

[7] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 1–19, Oct. 2009.

[8] A. Paterek, *Predicting Movie Ratings and Recommender Systems: A 195-Page Monograph on Machine Learning, Recommender Systems, and the Netflix Prize*, A. Paterek, Ed. 2012.

[9] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel recommendation model regularized with user trust and item ratings," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1607–1620, Jul. 2016.

[10] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst. (RecSys)*, New York, NY, USA, Sep. 2010, p. 135–142.

[11] I. Fernández-Tobías, M. Braunhofer, M. Elahi, F. Ricci, and I. Cantador, "Alleviating the new user problem in collaborative filtering by exploiting personality information," *User Model. User-Adapted Interact.*, vol. 26, nos. 2–3, pp. 221–255, Jun. 2016.

[12] Y. Liu, H. Qu, W. Chen, and S. M. H. Mahmud, "An efficient deep learning model to infer user demographic information from ratings," *IEEE Access*, vol. 7, pp. 53125–53135, Apr. 2019.

[13] Z. Cheng, X. Chang, L. Zhu, R. C. Kanjirathinkal, and M. Kankanhalli, "MMALFM: Explainable recommendation by leveraging reviews and images," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 1–28, Mar. 2019.

[14] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. Kankanhalli, "$A^3$NCF: An adaptive aspect attention model for rating prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, p. 3748–3754.

[15] M. R. Ghorab, D. Zhou, A. O'Connor, and V. Wade, "Personalised information retrieval: Survey and classification," *User Model. User-Adapted Interact.*, vol. 23, no. 4, pp. 381–443, Sep. 2012.

[16] J. Nguyen and M. Zhu, "Content-boosted matrix factorization techniques for recommender systems," *Stat. Anal. Data Mining*, vol. 6, no. 4, pp. 286–301, Aug. 2013.

[17] Y. Yu, C. Wang, H. Wang, and Y. Gao, "Attributes coupling based matrix factorization for item recommendation," *Int. J. Speech Technol.*, vol. 46, no. 3, pp. 521–533, Apr. 2017.

[18] C. Shi, Z. Zhang, Y. Ji, W. Wang, P. S. Yu, and Z. Shi, "SemRec: A personalized semantic recommendation method based on weighted heterogeneous information networks," *World Wide Web*, vol. 22, no. 1, pp. 153–184, Jan. 2019.

[19] J. Zheng, J. Liu, C. Shi, F. Zhuang, J. Li, and B. Wu, "Recommendation in heterogeneous information network via dual similarity regularization," *Int. J. Data Sci. Anal.*, vol. 3, no. 1, pp. 35–48, Nov. 2016.

[20] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.

[21] H. Zhang, I. Ganchev, N. S. Nikolov, and M. O'Droma, "A trust-enriched approach for item-based collaborative filtering recommendations," in *Proc. IEEE 12th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2016, pp. 65–68.

[22] H. Zhang, I. Ganchev, N. S. Nikolov, Z. Ji, and M. O'Droma, "Hybrid recommendation for sparse rating matrix: A heterogeneous information network approach," in *Proc. IEEE 2nd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Mar. 2017, pp. 740–744.

[23] H. Zhang, I. Ganchev, N. Nikolov, Z. Ji, and M. O'Droma, "A hybrid service recommendation prototype adapted for the UCWW: A smart-city orientation," *Wireless Commun. Mobile Comput.*, vol. 2017, Oct. 2017, Art. no. 6783240.

[24] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising auto-encoder," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, Oct. 2015, p. 811–820.

[25] S. Zhang, L. Yao, and X. Xu, "AutoSVD++: An efficient hybrid collaborative filtering model via contractive auto-encoders," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, Aug. 2017, p. 957–960.

[26] A. Rashed, J. Grabocka, and L. Schmidt-Thieme, "Attribute-aware non-linear co-embeddings of graph features," in *Proc. 13th ACM Conf. Recommender Syst.*, New York, NY, USA, 2019, p. 314–321.

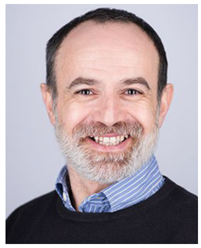
[27] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, Feb. 2019.
[28] M. F. Dacrema, P. Cremonesi, and D. Jannach, "Are we really making much progress? A worrying analysis of recent neural recommendation approaches," in *Proc. 13th ACM Conf. Recommender Syst.*, New York, NY, USA, Sep. 2019, p. 101–109.
[29] S. Rendle, L. Zhang, and Y. Koren, "On the difficulty of evaluating baselines: A study on recommender systems," 2019, *arXiv:1905.01395*. [Online]. Available: http://arxiv.org/abs/1905.01395
[30] J. Bennett and S. Lanning, "The Netflix prize," in *Proc. ACM SIGKDD KDD Cup Workshop*, Aug. 2007, pp. 3–6.
[31] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, KDD Cup Workshop*, Aug. 2007, pp. 39–42.
[32] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. 13th Int. Conf. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, 2000, pp. 535–541.
[33] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2007, pp. 1257–1264.
[34] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, 2008, p. 880–887.
[35] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3. Baltimore, MD, USA: Johns Hopkins Univ. Press, 2012.
[36] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, "Parallelized stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2010, pp. 2595–2603.
[37] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu, "Semantic path based personalized recommendation on weighted heterogeneous information networks," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, Oct. 2015, p. 453–462.
[38] 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, pp. 5–53, Jan. 2004.
[39] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature," *Geoscientific Model Develop.*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014.
[40] H. Zhang, N. S. Nikolov, and I. Ganchev, "Exploiting user feedbacks in matrix factorization for recommender systems," in *Model and Data Engineering*. Cham, Switzerland: Springer, Sep. 2017, pp. 235–247.
[41] F. Niu, B. Recht, C. Re, and S. J. Wright, "Hogwild!: A lock-free approach to parallelizing stochastic gradient descent," in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2011, pp. 693–701.
[42] L. Bottou, *Stochastic Gradient Descent Tricks*. Berlin, Germany: Springer, 2012, pp. 421–436.
[43] G. Guo, J. Zhang, Z. Sun, and N. Yorke-Smith, "LibRec: A Java Library for Recommender Systems," in *Proc. Posters, Demos, Late-Breaking Results Workshop Proc. 23rd Conf. User Modeling, Adaptation Personalization (UMAP)*, vol. 4, 2015, pp. 1–4.
[44] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, "Leveraging meta-path based context for top-N recommendation with a neural co-attention model," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2018, pp. 1531–1540.

**HAIYANG ZHANG** received the bachelor's degree from the School of Software Engineering, Jilin University, China, in 2013, and the Ph.D. degree from the Department of Electronic and Computer Engineering, University of Limerick, Ireland, in 2018.

She is currently a Research Associate with the Department of Computer Science, The University of Sheffield, U.K. Her current research interests include recommender systems, data mining, collaborative filtering, and natural language processing (NLP).

**IVAN GANCHEV** (Senior Member, IEEE) received the degree in engineering *(summa cum laude)* and the Ph.D. degree from the Saint-Petersburg University of Telecommunications, in 1989 and 1995, respectively.

He is currently a Full Professor with the University of Plovdiv "Paisii Hilendarski", an International Telecommunications Union (ITU-T) Invited Expert, and an Institution of Engineering and Technology (IET) Invited Lecturer. He was involved in more than 40 international and national research projects. He has authored/coauthored one monographic book, three textbooks, four edited books, and more than 280 research articles in refereed international journals, books, and conference proceedings. He has served on the TPC of over 320 prestigious international conferences/symposia/workshops. He is on the editorial board of and has served as a guest editor for multiple international journals.



**NIKOLA S. NIKOLOV** received the B.Sc. and M.Sc. degrees from Sofia University, in 1995, and the Ph.D. degree from the University of Limerick, in 2002.

He is currently a Lecturer with the Department of Computer Science and Information Systems and the Co-Head of the Big Data and Analytics Research Group, University of Limerick. He has worked as a principle investigator on research projects funded by the Science Foundation Ireland and the Irish Research Council. He is the author of more than 50 research articles in network visualisation and collaborative filtering. He is a member of the IEEE Technical Committee on Visual Analytics and Communication.

**ZHANLIN JI** (Member, IEEE) received the M.Eng. degree from Dublin City University, in 2006, and the Ph.D. degree from the University of Limerick, Ireland, in 2010.

He is currently a Professor with the North China University of Science and Technology, China. He is also a Researcher of the Telecommunications Research Centre (TRC), University of Limerick. He has authored/coauthored 70 research articles in refereed journals and conferences. His research interests include ubiquitous consumer wireless world (UCWW), the Internet of Things (IoT), cloud computing, big data management, and data mining.

**MÁIRTÍN O'DROMA** (Life Senior Member, IEEE) received the B.E. (Hons.) and Ph.D. degrees from the National University of Ireland, University College Dublin (UCD), in 1973 and 1978, respectively.

He has worked as a Senior Academic and the Director of the Telecommunications Research Center (TRC). He has authored/edited a number of research books and over 250 research publications in refereed international journals and conferences. His research interests include complex microwave and millimeter wave wireless telecommunication systems, future wireless network architectures and protocols, and related research.

Dr. O'Droma served as a member of the Governing Authority at the University of Limerick, Ireland (over two terms to 2016). He is a Fellow of the Institution of Engineering and Technology (IET). He has served or is serving as an IEEE Subject Matter Expert and an International Telecommunications Union (ITU-T) Invited Expert. Since 2008, he has been the President of the Irish Committee of Official Commission Members of the International Union of Radio Science (URSI), and a member of the Royal Irish Academy's Engineering and Computer Science Committee, since 2014.