ATTRIBUTE-AWARE COLLABORATIVE FILTERING: SURVEY AND CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Attribute-aware CF models aims at rating prediction given not only the historical rating from users to items, but also the information associated with users (e.g. age), items (e.g. price), or even ratings (e.g. rating time). This paper surveys works in the past decade developing attribute-aware CF systems, and discovered that mathematically they can be classified into four different categories. We provide the readers not only the high level mathematical interpretation of the existing works in this area but also the mathematical insight for each category of models. Finally we provide our preliminary experiment results comparing the effectiveness of the major works in each category.

1 INTRODUCTION

Collaborative filtering is arguably the most effective idea in building a recommender system. It assumes that a user's preferences on items can be inferred collaboratively from other users' preferences. In practice, users' past records toward items, such as explicit ratings or implicit feedback (e.g. binary access records), are typically used to infer similarity of taste among users for recommendation. In the past decade, *matrix factorization* (MF) has become a widely adopted realization of collaborative filtering. Specifically, MF learns a latent representation vector for a user and an item, and compute their inner products as the predicted rating. The learned latent user/item factors are supposed to embed the specific information about the user/item accordingly. That is, two users with similar latent representation shall have similar taste to items with similar latent vectors.

In big data era, classical MF using only ratings suffer a serious drawback for not being able to exploit other accessible information such as the features of users/items/ratings. For instance, data could contain the location and time about where and when a user rated an item. These rating-relevant attributes, or *contexts*, could be useful in determining the scale of a user liking an item. The *side information* or attributes relevant to users or items (e.g. the demographic information of users or the item genera) can also reveal useful information. Such side information is particularly useful for situation when the ratings about a user or an item is sparse, which is known as the *cold-start* problem for recommender systems. Therefore, researchers have formulated the *attribute-aware recommender systems* (see Figure 1) aiming at leverage not only the rating information but also the attributes associated with ratings/users/items to improve the quality of recommendation.

We have included about 80 papers in this area in the past decade, and found that the majority of the works propose an extension of matrix factorization to incorporate attribute information in collaborative filtering. The main contribution in this paper is to not only provide the review report, but rather a means to classify these works into four categories: (I) *discriminative matrix factorization*, (II) *generative matrix factorization*, (III) *generalized factorization*, and (IV) *heterogeneous graphs*. Inside each category, we provide the probabilistic interpretation of the models. The major distinction of these four categories lies in the representation of the interactions of users, items and attributes. The discriminative matrix factorization models extend the traditional MF by making the attributes prior knowledge input to learn the latent representation of users or items. Generative matrix factorization further considers the distributions of attributes, and learn such together with the rating distributions. Generalized factorization models view the user/item identity simply as a kind of attribute, and various models are designed for learning the low-dimensional representation vectors for rating prediction. The last category of models propose to represent the users, items and attributes using a heterogeneous graph, where a recommendation task can be cast into a link prediction task

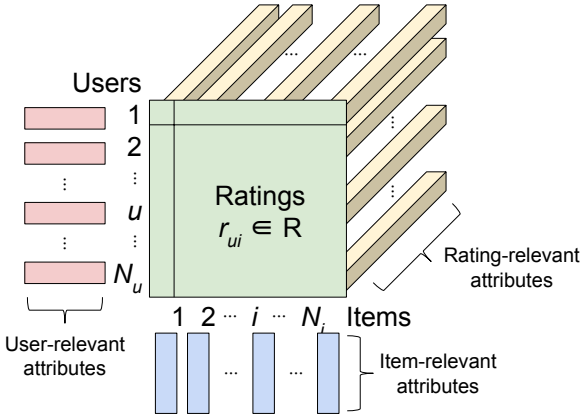


Figure 1: Overview of attribute-aware recommender systems. Attributes can be appended to users, items or ratings (feedback).

on the heterogeneous graph. In the following sections, we will elaborate the general mathematical explanations of the four types of model designs, and discuss the similarity/difference among models. We summarize our classification of models in table 1.

There have been four prior survey works (Adomavicius & Tuzhilin, 2011; Verbert et al., 2012; Bobadilla et al., 2013; Shi et al., 2014) introducing attribute-aware recommender systems. We claim two major differences between our work and the existing papers. First, previous survey mainly focuses on grouping different types of attributes, discussing the distinctions of memory-based collaborative filtering and model-based collaborative filtering, and presenting new challenges to be addressed. In contrast, we are the first that aims at classifying the existing works based on the methodology proposed, instead of the data used. We further provide mathematical representation for different types of models so the readers can better understand the spirit of the design of different models as well as their technical differences. Furthermore, previous survey include works published no later than the year of 2014, which means a large amount of state-of-the-art models, in particular deep-learning technology based models were not covered. This survey includes at least *25* new publications in this area. The publications being surveyed are classified and listed in the following table.

Note that this survey does not cover the hybrid recommender systems that leverage both rating and content information. While the contents typically indicate text attributes, this paper focuses on technique that handles *independent* attributes instead of attributes with local dependency (e.g. text, image attributes).

2 Preliminary: Collaborative Filtering and Matrix Factorization

Please refer to our Appendix A.

3 Classification of Attribute-Aware Recommender Systems

We first observe that most of the existing works use seperated matrices to represent the rating and attribute information. That is, there is a rating matrix between users and items, as well as attribute matrices between user and user-attributes, item and item-attributes, rating and rating-attributes, or a combination of them. Such attribute-matrices then become either the prior knowledge for learning the latent factors (Section 3.1) or the generative outputs from the latent factors (Section 3.2). On the other hand, some of the models can be regarded as a generalization of matrix factorization (Section 3.3). They view user or item IDs just as a kind of attributes and cast the recommendation task as a regression task to predict the ratings. Finally, some works attempt to model the interactions between users and items using a heterogeneous network, which can incorporate attributes by simply adding

Table 1: Our classification of attribute-aware recommender systems.

DMF	Similarity	Li et al. (2010a),Gu et al. (2010),Du et al. (2011),Zhou et al. (2012), Barjasteh et al. (2015), Yu et al. (2017),Adams et al. (2010), Chen et al. (2014), Gönen et al. (2013)
	Linear	Porteous et al. (2010),Menon & Elkan (2010),Menon et al. (2011), He & McAuley (2016), Zhao et al. (2016), Guo (2017),Feipeng Zhao (2017)
	Bilinear	Stern et al. (2009),Li et al. (2010b), Agarwal & Chen (2009),Shin et al. (2015) Yang et al. (2011), Chen et al. (2012),Park et al. (2013), Xu et al. (2013), Kim & Choi (2014), Natarajan & Dhillon (2014),Lu et al. (2016),Chou et al. (2016)
GMF	Multiple Matrix Factorization	Sedhain et al. (2017),Singh & Gordon (2008),Shan & Banerjee (2010), Ma et al. (2011),Yoo & Choi (2011),Fang & Si (2011),Bouchard et al. (2013), Saveski & Mantrach (2014),Gao et al. (2015),Ge et al. (2016),Brouwer & Liò (2017)
	Deep Neural Networks	Li et al. (2015),Wang et al. (2015),Zhang et al. (2016), Wang et al. (2016), Dong et al. (2017), Li & She (2017)
GF	TF	Tengfei Zhou (2017),Karatzoglou et al. (2010),Hidasi & Tikk (2012), Hidasi (2015),Kasai & Mishra (2016)
	FM	He & Chua (2017),Rendle et al. (2011),Cheng et al. (2014), Nguyen et al. (2014),Blondel et al. (2015),Blondel et al. (2016), Juan et al. (2016),Cao et al. (2016),Guo et al. (2017),Lu et al. (2017)
HG		Yu et al. (2014),Zheng et al. (2016),Palumbo et al. (2017)

attribute-representing nodes (Section 3.4). The rating estimation task is hence reduced to a link prediction problem between user and item nodes. Below we will discuss each class of models.

3.1 Discriminative Matrix Factorization (Figure 2)

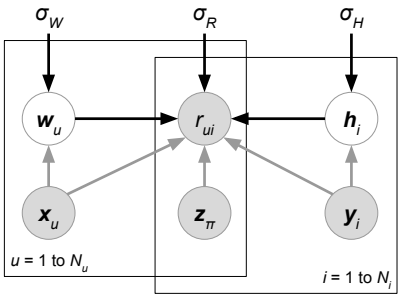


Figure 2: Graphical interpretation of discriminative probabilistic matrix factorization whose attributes $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ are observed for ratings and latent factors. User and item-relevant attributes $\boldsymbol{X}, \boldsymbol{Y}$ could affect the generation of latent factors $\boldsymbol{W}, \boldsymbol{H}$ or ratings $\boldsymbol{R}$, while rating-relevant attributes $\boldsymbol{Z}$ typically determines the rating prediction $\boldsymbol{R}$.

Here we propose to view the problem from a probabilistic perspective. The learning of Probabilistic Matrix Factorization (PMF) attempts to maximize posterior probability $p(\boldsymbol{W}, \boldsymbol{H} \mid \boldsymbol{R})$ of two latent factor matrices $\boldsymbol{W}$ (for users) and $\boldsymbol{H}$ (for items), given observed ratings matrix $\boldsymbol{R}$. Given extra attribute matrix $\boldsymbol{X}$, based on Bayes' rule, the posterior probability can be shown as follows:

$$\begin{aligned}
\underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}} \underbrace{p\left(\boldsymbol{W}, \boldsymbol{H} \mid \boldsymbol{R}, \boldsymbol{X}\right)}_{\text{Posterior}} &= \underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}} \frac{p\left(\boldsymbol{R} \mid \boldsymbol{W}, \boldsymbol{H}, \boldsymbol{X}\right) p\left(\boldsymbol{W}, \boldsymbol{H} \mid \boldsymbol{X}\right)}{p\left(\boldsymbol{R} \mid \boldsymbol{X}\right)} \\
&= \underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}} p\left(\boldsymbol{R} \mid \boldsymbol{W}, \boldsymbol{H}, \boldsymbol{X}\right) p\left(\boldsymbol{W}, \boldsymbol{H} \mid \boldsymbol{X}\right) \\
&= \underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}} \underbrace{p\left(\boldsymbol{R} \mid \boldsymbol{W}, \boldsymbol{H}, \boldsymbol{X}\right)}_{\text{Likelihood}} \underbrace{p\left(\boldsymbol{W} \mid \boldsymbol{X}\right) p\left(\boldsymbol{H} \mid \boldsymbol{X}\right)}_{\text{Prior}}. \qquad (1)
\end{aligned}$$

We ignore the denominator $p(\boldsymbol{R} \mid \boldsymbol{X})$ since it is fully observed. As to the prior, we follow the independence assumption $\boldsymbol{W} \perp \boldsymbol{H}$ of PMF, despite that here the independence is conditioned on the attribute matrix $\boldsymbol{X}$. Compared with the classical PMF, both likelihood $p(\boldsymbol{R} \mid \boldsymbol{W}, \boldsymbol{H}, \boldsymbol{X})$ and prior $p(\boldsymbol{W} \mid \boldsymbol{X})\, p(\boldsymbol{H} \mid \boldsymbol{X})$ could be affected by attributes $\boldsymbol{X}$. Attributes in the likelihood function directly affects the ranking prediction, while attributes in the priors regularize the learning directions of latent factors.

We further generate the sub-categoryies as below.

3.1.1 Attributes in a Linear Model

This is the generalized form to utilize attributes in this category. Given the attributes, a weight vector is applied to perform linear regression on the attributes. Its characteristic in mathematical form is shown in likelihood functions:

$$\underset{\boldsymbol{W},\boldsymbol{H},\theta}{\operatorname{argmax}} \prod_{(u,i)|r_{ui}\in\delta(\boldsymbol{R})} \mathcal{N}\left(r_{ui} \mid \mu_R = \boldsymbol{w}_u^\top \boldsymbol{h}_i + \boldsymbol{a}_u^\top \boldsymbol{x}_u + \boldsymbol{b}_i^\top \boldsymbol{y}_i + \boldsymbol{c}^\top \boldsymbol{z}_{\pi(u,i)}, \sigma_R^2\right) p(\boldsymbol{W} \mid \boldsymbol{X}) p(\boldsymbol{H} \mid \boldsymbol{Y}), \tag{2}$$

where $\theta = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{\alpha}, \boldsymbol{\beta}\}$, $\delta(\boldsymbol{R})$ denotes the non-missing ratings in the training data, and $\pi(u,i)$ is the column index corresponding to user u and item i. $\boldsymbol{X} \in \mathbb{R}^{K\times N_u}$, $\boldsymbol{Y} \in \mathbb{R}^{K\times N_i}$, $\boldsymbol{Z} \in \mathbb{R}^{K\times|\delta(\boldsymbol{R})|}$ respectively denote attribute matrices relevant to user, item and ratings, while $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ are their corresponding weight vectors. A simple linear regression model can be expressed as a likelihood function of normal distribution $\mathcal{N}(r \mid \mu, \sigma^2)$ with mean μ and variance σ^2. Ideally the distributions of latent factors $\boldsymbol{W}, \boldsymbol{H}$ shall have prior knowledge from attributes $\boldsymbol{X}, \boldsymbol{Y}$, but we have not yet observed an approach aiming at designing attribute-aware priors as the last two terms of (2).

Bayesian Matrix Factorization with Side Information (BMFSI) (Porteous et al., 2010) is an example case in this sub-category. On the basis of Bayesian Probabilistc Matrix Factorization (BPMF) (Salakhutdinov & Mnih, 2008a), BMFSI uses a linear combination like (2) to introduce attribute information to rating prediction. It is formulated as:

$$\begin{aligned}
&\underset{\boldsymbol{W},\boldsymbol{H},\theta}{\operatorname{argmax}} \underbrace{p(\boldsymbol{R} \mid \boldsymbol{W}, \boldsymbol{H}, \theta)}_{\text{Likelihood}} \underbrace{p(\boldsymbol{W}) p(\boldsymbol{H})}_{\text{Priors}} \\
&= \underset{\boldsymbol{W},\boldsymbol{H},\theta}{\operatorname{argmax}} \underbrace{\prod_{(u,i)|r_{ui}\in\delta(\boldsymbol{R})} \mathcal{N}\left(r_{ui} \mid \boldsymbol{w}_u^\top \boldsymbol{h}_i + \boldsymbol{a}_u^\top \boldsymbol{x}_u + \boldsymbol{b}_i^\top \boldsymbol{y}_i, \sigma_R^2\right)}_{\text{Matrix factorization using attributes}} \underbrace{\prod_u \mathcal{N}(\boldsymbol{w}_u \mid \boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u) \prod_i \mathcal{N}(\boldsymbol{h}_i \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}_{\text{Regularization}},
\end{aligned} \tag{3}$$

where $\theta = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$ and $\delta(\boldsymbol{R})$ is the set of training ratings. The difference from (2) is that rating attributes $\boldsymbol{z}$ shall be concatenated with either $\boldsymbol{x}_u$ or $\boldsymbol{y}_u$, and thus we drop an independent weight variable c in BMFSI. We ignore other attribute-free designs of BMFSI (e.g. Dirichlet process).

3.1.2 Attributes in a Bilinear Model

This a popular method when two kinds of attributes (usually user and item) are provided. Given user attribute matrix $\boldsymbol{X}$ and item attribute matrix $\boldsymbol{Y}$, a matrix $\boldsymbol{A}$ is used to model the relation between them. The mathematical form can be viewed as the following:

$$\underset{\boldsymbol{W},\boldsymbol{H},\theta}{\operatorname{argmax}} \prod_{(u,i)|r_{ui}\in\delta(\boldsymbol{R})} \mathcal{N}\left(r_{ui} \mid \mu_R = \boldsymbol{x}_u^\top \boldsymbol{A} \boldsymbol{y}_i + \boldsymbol{c}_u^\top \boldsymbol{x}_u + \boldsymbol{d}_i^\top \boldsymbol{y}_i + \boldsymbol{b}, \sigma_R^2\right) p(\boldsymbol{W}|\boldsymbol{X}) p(\boldsymbol{H}|\boldsymbol{Y}), \tag{4}$$

where $\theta = \{\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{\alpha}, \boldsymbol{\beta}\}$. In fact, as mentioned in Lu et al. (2016), $\boldsymbol{c}_u^\top \boldsymbol{x}_u + \boldsymbol{d}_i^\top \boldsymbol{y}_i + \boldsymbol{b}$

can be absorbed into the first term, by appending a new dimension whose value is fixed to 1 for each $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$\mu_R = \boldsymbol{x}_u^\top \boldsymbol{A} \boldsymbol{y}_i + \boldsymbol{c}_u^\top \boldsymbol{x}_u + \boldsymbol{d}_i^\top \boldsymbol{y}_i + \boldsymbol{b} = \widetilde{\boldsymbol{x}}_u^\top \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{y}}_i. \tag{5}$$

Works in this category differ in whether the bilinear term is explicit or implicit. (4) implies that the form of the dot product of two linear-transformed attributes $\boldsymbol{w}_u = \boldsymbol{S}\boldsymbol{x}_u$ and $\boldsymbol{h}_i = \boldsymbol{T}\boldsymbol{y}_i$ since it can

be reformed as $\boldsymbol{w}_u^\top \boldsymbol{h}_i = \boldsymbol{x}_u^\top (\boldsymbol{S}^\top \boldsymbol{T}) \boldsymbol{y}_i$ where $\boldsymbol{A} = \boldsymbol{S}^\top \boldsymbol{T}$. Some works such as Regression-based Latent Factor Model (see below) chooses to softly constrain $\boldsymbol{w}_u \approx \boldsymbol{S}\boldsymbol{x}_u$ and $\boldsymbol{h}_i \approx \boldsymbol{T}\boldsymbol{y}_i$ instead of strict equations.

Regression-based Latent Factor Model (RLFM) (Agarwal & Chen, 2009) . Given three types of attribute matrices: user-relevant $\boldsymbol{X}$, item-relevant $\boldsymbol{Y}$ and rating-relevant $\boldsymbol{Z}$, RLFM models them in different parts of biased matrix factorization. $\boldsymbol{X}, \boldsymbol{Y}$ serve as the hyperparameters of latent factors, while $\boldsymbol{Z}$ joins the regression framework to predict ratings together with latent factors. RLFM can be written as:

$$\begin{aligned}
&\underset{\boldsymbol{W},\boldsymbol{H},\theta}{\operatorname{argmax}} \underbrace{p(\boldsymbol{R} \mid \boldsymbol{W}, \boldsymbol{H}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{\gamma}, \boldsymbol{Z})}_{\text{Likelihood}} \underbrace{p(\boldsymbol{W} \mid \boldsymbol{A}, \boldsymbol{X}) p(\boldsymbol{H} \mid \boldsymbol{B}, \boldsymbol{Y}) p(\boldsymbol{c} \mid \boldsymbol{\alpha}, \boldsymbol{X}) p(\boldsymbol{d} \mid \boldsymbol{\beta}, \boldsymbol{Y})}_{\text{Prior}} \\
= &\underset{\boldsymbol{W},\boldsymbol{H},\theta}{\operatorname{argmax}} \underbrace{\prod_{(u,i)|r_{ui} \in \delta(\boldsymbol{R})} \mathcal{N}\left(r_{ui} \mid \boldsymbol{w}_u^\top \boldsymbol{h}_i + c_u + d_i + \boldsymbol{\gamma}^\top \boldsymbol{z}_{\pi(u,i)}, \sigma_R^2\right)}_{\text{Matrix factorization using attributes}} \\
&\underbrace{\prod_u \mathcal{N}\left(\boldsymbol{w}_u \mid \boldsymbol{A}\boldsymbol{x}_u, \boldsymbol{\Sigma}_W\right) \mathcal{N}\left(c_u \mid \boldsymbol{\alpha}^\top \boldsymbol{x}_u, \sigma_c^2\right) \prod_i \mathcal{N}\left(\boldsymbol{h}_i \mid \boldsymbol{B}\boldsymbol{y}_i, \boldsymbol{\Sigma}_H\right) \mathcal{N}\left(d_i \mid \boldsymbol{\beta}^\top \boldsymbol{y}_i, \sigma_d^2\right)}_{\text{Regularization using attributes}}
\end{aligned} \tag{6}$$

where $\theta = \{\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\}$, and $\delta(\boldsymbol{R})$ is the set of non-missing ratings for training. Biased matrix factorization adds two vectors $\boldsymbol{c}, \boldsymbol{d}$ to learn the biases for each user or item. Parameters $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ map attributes with latent factors (for $\boldsymbol{X}, \boldsymbol{Y}$) or rating prediction (for $\boldsymbol{Z}$).

3.1.3 Attributes in a Similarity Matrix

In this case, a similarity matrix which measures the closeness of attributes between users or between items is presented. Given the user attribute matrix $\boldsymbol{X} \in \mathbb{R}^{D \times N_u}$, where N_u is the number of users and D is the dimension of user attribute, a similarity matrix $\boldsymbol{S} \in \mathbb{R}^{N_u \times N_u}$ is computed. There are many metrics to for similarity calculation such as Euclidean distance or kernel functions. The similarity matrix is then used for matrix factorization or other solutions. The speciality of this case is that human knowledge is involved in determining how the interactions between attributes should be modeled. Kernelized Probabilistic Matrix Factorization is an example which utilizes both user similarity matrix and item similarity matrix.

Kernelized Probabilistic Matrix Factorization (KPMF) (Zhou et al., 2012) . Let K, N_u, N_i be the number of latent factors, users and items. Given user-relevant attribute matrix $\boldsymbol{X} \in \mathbb{R}^{K \times N_u}$ or item-relvant attribute matrix $\boldsymbol{Y} \in \mathbb{R}^{K \times N_i}$, we can always obtain a similarity matrix $\boldsymbol{S}_{\boldsymbol{X}} \in \mathbb{R}^{N_u \times N_u}$ or $\boldsymbol{S}_{\boldsymbol{Y}} \in \mathbb{R}^{N_i \times N_i}$ where each entry stores a pre-defined similarity between a pair of users or items. Then KPMF formulates the similarty matrix as the prior of its corresponding latent factor matrix:

$$\begin{aligned}
&\underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}} \underbrace{p(\boldsymbol{R} \mid \boldsymbol{W}, \boldsymbol{H})}_{\text{Likelihood}} \underbrace{p(\boldsymbol{W} \mid \boldsymbol{X}) p(\boldsymbol{H} \mid \boldsymbol{Y})}_{\text{Prior}} \\
= &\underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}} \underbrace{\prod_{(u,i)|r_{ui} \in \delta(\boldsymbol{R})} \mathcal{N}\left(r_{ui} \mid \boldsymbol{w}_u^\top \boldsymbol{h}_i, \sigma_R^2\right)}_{\text{Matrix factorization}} \underbrace{\prod_k \mathcal{N}\left(\boldsymbol{w}^k \mid \boldsymbol{0}, \boldsymbol{S}_{\boldsymbol{X}}\right) \prod_l \mathcal{N}\left(\boldsymbol{h}^l \mid \boldsymbol{0}, \boldsymbol{S}_{\boldsymbol{Y}}\right)}_{\text{Regularization using attributes}} .
\end{aligned} \tag{7}$$

Here we use subscripts $\boldsymbol{w}_u$ to denote the u-th column vector of a matrix $\boldsymbol{W}$, while superscripts $\boldsymbol{w}^k$ imply the k-th row vector of $\boldsymbol{W}$. Intuitively, the similarity matrices control the learning preferences of user or item latent factors. If two users have similar user-relevant attributes (i.e., they have a higher similarity measure in $\boldsymbol{S}_{\boldsymbol{X}}$), then their latent factors are forced to be closer during the matrix factorization learning.

3.2 Generative Matrix Factorization (Figure 3)

Mathematically, by Bayes' rule, we maximize a posteriori as follows:

$$\underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}} \underbrace{p\left(\boldsymbol{W}, \boldsymbol{H} \mid \boldsymbol{R}, \boldsymbol{X}\right)}_{\text{Posterior}} = \underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}} \frac{p\left(\boldsymbol{R}, \boldsymbol{X} \mid \boldsymbol{W}, \boldsymbol{H}\right) p\left(\boldsymbol{W}, \boldsymbol{H}\right)}{p\left(\boldsymbol{R}, \boldsymbol{X}\right)}$$

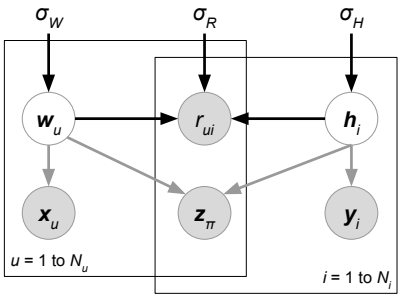


Figure 3: Graphical interpretation of generative probabilistic matrix factorization whose attributes $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ together with ratings are generated by latent factors. Rating-relevant attributes $\boldsymbol{Z}$ is likely to result from both $\boldsymbol{W}$ and $\boldsymbol{H}$. For models of this class, some of the gray arrows are removed to represent their additional independence assumptions about attribute generation.

$$
\begin{aligned}
&= \underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}}\, p\left(\boldsymbol{R}, \boldsymbol{X} \mid \boldsymbol{W}, \boldsymbol{H}\right) p\left(\boldsymbol{W}, \boldsymbol{H}\right) \\
&= \underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}}\, \underbrace{p\left(\boldsymbol{R} \mid \boldsymbol{W}, \boldsymbol{H}\right) p\left(\boldsymbol{X} \mid \boldsymbol{W}, \boldsymbol{H}\right)}_{\text{Likelihood}} \underbrace{p\left(\boldsymbol{W}\right) p\left(\boldsymbol{H}\right)}_{\text{Prior}}. \qquad (8)
\end{aligned}
$$

where $p(\boldsymbol{R}, \boldsymbol{X})$ does not affect the posterior maximization. Typically relevant works assume additional independence $\boldsymbol{R} \perp \boldsymbol{X}$ given latent factors $\boldsymbol{W}, \boldsymbol{H}$ in (8). Furthermore, some existing works choose to consider attributes on either side of $p(\boldsymbol{X} \mid \boldsymbol{W})$ and $p(\boldsymbol{X} \mid \boldsymbol{H})$.

There are two different branches is this direction. On one hand, earlier works use the matrix factorization technique again, to generate attributes from user or item latent factors. It can be seen as a linear mapping between latent factors and attributes. On the other hand, with the help of deep neural networks, recent works combine matrix factorization and deep autoencoders to realize non-linear mappings for attribute generation. We will introduce them in the following sections.

3.2.1 Attributes in Matrix Factorization

Similar to PMF $\boldsymbol{R} \approx \boldsymbol{W}^\top \boldsymbol{H}$ for rating distributions, attributes distributions are modeled using another matrix factorization form. Given user attribute matrix $\boldsymbol{X}$, item attribute matrix $\boldsymbol{Y}$ and rating attribute matrix $\boldsymbol{Z}$, they can be factorized as $\boldsymbol{X} \approx \boldsymbol{A}^\top \boldsymbol{W}, \boldsymbol{Y} \approx \boldsymbol{B}^\top \boldsymbol{H}$ of low rank. Specifically, its objective function is written as:

$$
\begin{aligned}
\underset{\boldsymbol{W},\boldsymbol{H},\boldsymbol{A},\boldsymbol{B}}{\operatorname{argmax}} & \prod_{(u,i)|r_{ui}\in\delta(\boldsymbol{R})} \mathcal{N}\left(r_{ui} \mid \mu_R = \boldsymbol{w}_u^\top \boldsymbol{h}_i, \sigma_R^2\right) \prod_{(j,u)} \mathcal{N}\left(x_{ju} \mid \boldsymbol{a}_j^\top \boldsymbol{w}_u, \sigma_X^2\right) \prod_{(v,i)} \mathcal{N}\left(x_{vi} \mid \boldsymbol{b}_v^\top \boldsymbol{h}_i, \sigma_Y^2\right) \\
& \prod_{(u,i)|r_{ui}\in\delta(\boldsymbol{R})} \mathcal{N}\left(\boldsymbol{z}_{ui} \mid \boldsymbol{w}_u^\top \boldsymbol{C} \boldsymbol{h}_i, \sigma_Z^2\right) p(\boldsymbol{W}) p(\boldsymbol{H}), \qquad (9)
\end{aligned}
$$

where $\delta(\boldsymbol{R})$ denote the non-missing entries of matrix $\boldsymbol{R}$. The insight of (9) is to share the latent factors $\boldsymbol{W}, \boldsymbol{H}$ in multiple factorization tasks. $\boldsymbol{W}$ is shared with user attributes, while $\boldsymbol{H}$ is shared with item attributes. $\boldsymbol{Z}$ requires the sharing of both $\boldsymbol{W}$ and $\boldsymbol{H}$ due to user and item-specific rating attributes. Therefore the side information of both $\boldsymbol{X}, \boldsymbol{Y}$ and $\boldsymbol{Z}$ can indirectly transfer to rating prediction. Auxiliary matrices $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}$ learns the mappings between latent factors and attributes. With respect to the mathematical form of matrix factorization, the expectation of feature values is linearly correlated with its corresponding latent factors.

Collective Matrix Factorization (CMF) (Singh & Gordon, 2008) Here we introduce a common model in this sub-category. The CMF framework relies on the combination of multiple matrix factorization objective functions. CMF first builds the MF for rating matrix $\boldsymbol{R}$. Then user and item-relevant attribute matrices $\boldsymbol{X}, \boldsymbol{Y}$ are appended to the matrix factorization objectives. Overall we have:

$$
\underset{\boldsymbol{W},\boldsymbol{H},\boldsymbol{A},\boldsymbol{B}}{\operatorname{argmax}} \underbrace{p(\boldsymbol{R} \mid \boldsymbol{W}, \boldsymbol{H}) p(\boldsymbol{X} \mid \boldsymbol{W}, \boldsymbol{A}) p(\boldsymbol{Y} \mid \boldsymbol{H}, \boldsymbol{B})}_{\text{Likelihood}} \underbrace{p(\boldsymbol{W}) p(\boldsymbol{H}) p(\boldsymbol{A}) p(\boldsymbol{B})}_{\text{Prior}}
$$

$$= \underset{\boldsymbol{W},\boldsymbol{H},\boldsymbol{A},\boldsymbol{B}}{\operatorname{argmax}} \underbrace{\prod_{(u,i)|r_{ui}\in\delta(\boldsymbol{R})} \mathcal{N}\left(r_{ui} \mid \boldsymbol{w}_u^\top \boldsymbol{h}_i, \sigma_R^2\right)}_{\text{Matrix factorization of } R} \underbrace{\prod_{(j,u)} \mathcal{N}\left(x_{ju} \mid \boldsymbol{a}_j^\top \boldsymbol{w}_u, \sigma_X^2\right)}_{\text{Matrix factorization of } X} \underbrace{\prod_{(v,i)} \mathcal{N}\left(y_{vi} \mid \boldsymbol{b}_v^\top \boldsymbol{h}_i, \sigma_Y^2\right)}_{\text{Matrix factorization of } Y}$$
$$\underbrace{\prod_u \mathcal{N}\left(\boldsymbol{w}_u \mid \boldsymbol{0}, \boldsymbol{\Sigma}_W\right) \prod_i \mathcal{N}\left(\boldsymbol{h}_i \mid \boldsymbol{0}, \boldsymbol{\Sigma}_H\right) \prod_j \mathcal{N}\left(\boldsymbol{a}_j \mid \boldsymbol{0}, \boldsymbol{\Sigma}_A\right) \prod_v \mathcal{N}\left(\boldsymbol{b}_v \mid \boldsymbol{0}, \boldsymbol{\Sigma}_B\right)}_{\text{Regularization}} \tag{10}$$

where $\delta(\boldsymbol{R}), \delta(\boldsymbol{X}), \delta(\boldsymbol{Y})$ denote the non-missing entries of matrix $\boldsymbol{R}, \boldsymbol{X}, \boldsymbol{Y}$ that are generated by latent factor matrices $\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{A}, \boldsymbol{B}$ of zero-mean normal priors (i.e., $l2$ regularization). In (10), $\boldsymbol{W}, \boldsymbol{H}$ are shared by at least two matrix factorization objectives. Attribute information in $\boldsymbol{X}, \boldsymbol{Y}$ is transferred to rating prediction $\boldsymbol{R}$ through sharing the same latent factors. Note that CMF is not limited to three matrix factorization objectives (10).

3.2.2 ATTRIBUTES IN DEEP NEURAL NETWORKS

In deep neural networks, an autoencoder is usually used to learn latent representation of observed data. Specifically the model tries to construct a encoder $\mathcal{E}$ and a decoder $\mathcal{D}$, where the encoder learns to map from a possibly modified attributes $\tilde{\boldsymbol{X}}$ to low-dimensional latent factors, and the decoder recover from latent factors to the original attributes $\boldsymbol{X}$. Moreover, activation functions in autoencoders can reflect non-linear mappings between latent factors and attributes, which may capture the characteristics of attributes more accurately.

To implement an autoencoder, at first we generate another attribute matrix $\tilde{\boldsymbol{X}}$ from $\boldsymbol{X}$. $\tilde{\boldsymbol{X}}$ could be the same as $\boldsymbol{X}$, or different due to corruption, e.g., adding random noise. Autoencoders aim to predict the original $\boldsymbol{X}$ using latent factors that are inferred from generated $\tilde{\boldsymbol{X}}$. Here attributes serve not only as the generation results $\boldsymbol{X}$, but also as the prior knowledge $\tilde{\boldsymbol{X}}$ of latent factors. Let us review Bayes' Rule to figure out where autoencoders appears for generative matrix factorization:

$$\begin{aligned}
\underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}}\, p\left(\boldsymbol{W},\boldsymbol{H} \mid \boldsymbol{R},\boldsymbol{X},\tilde{\boldsymbol{X}}\right) &= \underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}} \frac{p\left(\boldsymbol{R},\boldsymbol{X} \mid \boldsymbol{W},\boldsymbol{H},\tilde{\boldsymbol{X}}\right) p\left(\boldsymbol{W},\boldsymbol{H} \mid \tilde{\boldsymbol{X}}\right)}{p\left(\boldsymbol{R},\boldsymbol{X} \mid \tilde{\boldsymbol{X}}\right)} \\
&= \underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}}\, p\left(\boldsymbol{R},\boldsymbol{X} \mid \boldsymbol{W},\boldsymbol{H},\tilde{\boldsymbol{X}}\right) p\left(\boldsymbol{W},\boldsymbol{H} \mid \tilde{\boldsymbol{X}}\right) \\
&= \underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmax}} \underbrace{\underbrace{p\left(\boldsymbol{R} \mid \boldsymbol{W},\boldsymbol{H},\tilde{\boldsymbol{X}}\right)}_{\text{Matrix factorization}} \underbrace{p\left(\boldsymbol{X} \mid \boldsymbol{W},\boldsymbol{H},\tilde{\boldsymbol{X}}\right)}_{\text{Decoder}\mathcal{D}}}_{\text{Likelihood}} \underbrace{p\left(\boldsymbol{H} \mid \tilde{\boldsymbol{X}}\right) p\left(\boldsymbol{W} \mid \tilde{\boldsymbol{X}}\right)}_{\text{Priors, i.e, Encoder}\mathcal{E}}.
\end{aligned} \tag{11}$$

$p(\boldsymbol{R}, \boldsymbol{Y} \mid \tilde{\boldsymbol{Y}})$ is eliminated due to irrelevance in maximization of (11). By sharing latent factors $\boldsymbol{W}, \boldsymbol{H}$ between autoencoders and matrix factorization, attribute information can affect the learning of rating prediction. Modeling $\mathcal{D}$ with normal distributions, we can conclude that the expectation of attributes $\boldsymbol{X}$ is non-linearly mapped from from latent factors $\boldsymbol{W}, \boldsymbol{H}$. Although latent factors have priors from attributes, we categorize relevant works into generative matrix factorization, since we explicitly model attribute distributions in the decoder part of autoencoders.

Collaborative Deep Learning (CDL) (Wang et al., 2015). The model presents a combination method of collaborative filtering and Stacked Denoising Auto-Encoder (SDAE). Since the model claim to exploit item attributes $\boldsymbol{Y}$ only, in the following introduction we define $\boldsymbol{Y} = \boldsymbol{X}, \tilde{\boldsymbol{Y}} = \tilde{\boldsymbol{X}}$ in (11).

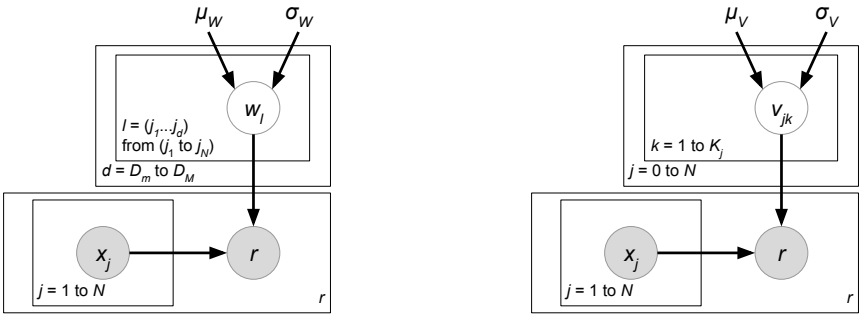
In SDAE, input attributes $\tilde{\boldsymbol{Y}}$ is not equivalent to $\boldsymbol{Y}$ due to adding random noise to $\tilde{\boldsymbol{Y}}$. CDL implicitly adds several independence assumptions $(\boldsymbol{R} \perp \tilde{\boldsymbol{Y}} \mid \boldsymbol{W}, \boldsymbol{H}), (\boldsymbol{Y} \perp \boldsymbol{W} \mid \boldsymbol{H}, \tilde{\boldsymbol{Y}}), (\boldsymbol{W} \perp \tilde{\boldsymbol{Y}})$ to formulate its model. Then using identical notations in CMF introduction, normal distributions $\mathcal{N}$ are again

applied to CDL:

$$\begin{aligned}
&\underset{\boldsymbol{W},\boldsymbol{H},\boldsymbol{\theta},\boldsymbol{\phi}}{\operatorname{argmax}} \underbrace{p\left(\boldsymbol{R},\mid \boldsymbol{W},\boldsymbol{H}\right) p\left(\boldsymbol{Y}\mid \boldsymbol{H},\tilde{\boldsymbol{Y}}\right)}_{\text{Likelihood}} \underbrace{p\left(\boldsymbol{H}\mid \tilde{\boldsymbol{Y}}\right) p\left(\boldsymbol{W}\right)}_{\text{Prior}} \\
= &\underset{\boldsymbol{W},\boldsymbol{H},\boldsymbol{\theta},\boldsymbol{\phi}}{\operatorname{argmax}} \underbrace{\prod_{(u,i)|r_{ui}\in\delta(\boldsymbol{R})} \mathcal{N}\left(r_{ui}\mid \boldsymbol{w}_u^\top \boldsymbol{h}_i,\sigma_R^2\right)}_{\text{Matrix factorization}} \underbrace{\prod_i \mathcal{N}\left(\boldsymbol{y}_i\mid \mathcal{D}_{\boldsymbol{\phi}}(\mathcal{E}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{y}}_i)),\boldsymbol{\Sigma}_Y\right)\mathcal{N}\left(\boldsymbol{h}_i\mid \mathcal{E}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{y}}_i),\boldsymbol{\Sigma}_H\right)}_{\text{Stacked denoising auto-encoder for } \boldsymbol{Y}} \\
&\underbrace{\prod_u \mathcal{N}\left(w_u\mid \boldsymbol{0},\boldsymbol{\Sigma}_W\right)}_{\text{Regularization}}. \qquad (12)
\end{aligned}$$

Functions $\mathcal{E}, \mathcal{D}$ indicate the encoder and the decoder of SDAE. The two functions could be formed by multi-layer perceptrons whose parameters are denoted by $\boldsymbol{\theta}, \boldsymbol{\phi}$. It is clear to see the distribution of attribute matrix $\boldsymbol{Y}$ be modeled in the decoder part. Last but not least, the analysis from (11) to (12) imply that others ideas, user-relevant attributes for example, could be naturally involved in CDL, as long as we remove more independence assumptions.

3.3 Generalized Factorization



(a) w-weighted generalization (b) v-approximate generalization

Figure 4: Graphical interpretation of generalized factorization. Attributes x including user or item indices are weighted with corresponding w in order to fit a true rating r. If we have all the w, v's follow normal distributions of shared hyperparameters, then there are hyperparameters μ_W, σ_W or μ_V, σ_V.

Thanks to the success of matrix factorization in recommender systems, there emerge advanced works asking for generalizing the concept of matrix factorization, in order to extract more information from attributes or interactions between users and items. The works classified in either Section 3.1 or Section 3.2 propose to design attribute-aware components on the basis of PMF. They explicitly express an assumption of vanilla PMF: a latent factor matrix $\boldsymbol{W}$ to represent user preferences and another matrix $\boldsymbol{H}$ for items. However the works classified in this section do not regard $\boldsymbol{W}$ and $\boldsymbol{H}$ as a special existence in models. Rather, such works propose a expanded latent factor space shared by users, items and attributes. Here neither users nor items are special entities in a recommender system. They are simply considered as categorical attributes. Taking rating r_{ui} for example, it implies that we have a one-hot user encoding vector where all the entries are 0 except for the u-th entry; similarly, we also have a one-hot item encoding vector of the i-th entry being 1. Thus external attributes $\boldsymbol{X}$ can be simply involved in the matrix-factorization-based models, because now users and items are also attributes whose interactions commonly predict or rank ratings.

We first propose the most generalized version of interpretation: Given a rating r and its corresponding attribute vector $\boldsymbol{x} \in \mathbb{R}^N$, then we make rating estimate:

$$\underset{w}{\operatorname{argmax}} \prod_{r \in \delta(\boldsymbol{R})} \mathcal{N}\left(r \mid \mu_R = \sum_{d=D_m}^{D_M} \sum_{j_1=1}^{N} \sum_{j_2=j_1+1}^{N} \cdots \sum_{j_d=j_{d-1}+1}^{N} w_{j_1 j_2 \ldots j_d} \left(x_{j_1} x_{j_2} \ldots x_{j_d}\right), \sigma_R^2\right), \tag{13}$$

where $\delta(\boldsymbol{R})$ indicates the set of observed ratings in training data. Variable $d \in \{0\} \cup \mathbb{N}$ determines the dth-order multiplication interaction between attributes x_j. As $d = 0$, we introduce an extra bias weight $w_0 \in \mathbb{R}$ in (13). The large number of parameters $w \in \mathbb{R}$ is very likely to overfit training ratings due to the dimensionality curse. To alleviate overfitting problems, the ideas in matrix factorization are applied here. For higher values of d, it is assumed that each w is a function of low-dimensional latent factors:

$$w_{j_1 j_2 \ldots j_d} = f_d\left(\boldsymbol{v}_{j_1}, \boldsymbol{v}_{j_2}, \ldots, \boldsymbol{v}_{j_d}\right), \tag{14}$$

where $\boldsymbol{v}_j \in \mathbb{R}^{K_j}$ implies the K-dimensional ($K_j \ll N \; \forall j$) latent factor or representation vector for each element x_j of $\boldsymbol{x}$. Function f_d maps these d vectors to a real-valued weight. Then our learning parameters become $\boldsymbol{v}$. The overall number of parameters ($D_m \leq d \leq D_M$) decreases from $\sum_{d=D_m}^{D_M} \frac{n!}{d!(n-d)!} = O(2^N)$ to $\sum_{j=1}^{N} K_j = O(NK)$ where $K = \max_{1 \leq j \leq N} K_j$. Next we prove that matrix factorization is a special case of (13). Let $D_m = D_M = 2$ and $\boldsymbol{x}$ be the concatenation of one-hot encoding vectors of users as well as items. Also we define $f_2(\boldsymbol{v}, \boldsymbol{y}) = \boldsymbol{v}^\top \boldsymbol{y}$. Then for rating r_{ui} of user u to item i, we have:

$$\underset{\boldsymbol{v}}{\operatorname{argmax}} \prod_{r_{ui} \in \delta(\boldsymbol{R})} \mathcal{N}\left(\hat{r}_{ui} \mid \mu_R = \sum_{j_1=1}^{N} \sum_{j_2=j_1+1}^{N} \boldsymbol{v}_{j_1}^\top \boldsymbol{v}_{j_2} \left(x_{j_1} x_{j_2}\right) = \boldsymbol{v}_u^\top \boldsymbol{v}_{N_u+i}, \sigma_R^2\right), \tag{15}$$

where N_u denotes the number of users. (15) is essentially equivalent to matrix factorization.

In this class, the existing works either generalize or improve two early published works: Tensor Factorization (TF) and Factorization Machine (FM). Both models can be viewed as the special case of (13). We introduce TF and FM in the sections below.

3.3.1 TF-EXTENDED MODELS

Tensor Factorization (TF) (Karatzoglou et al., 2010) requires the input featores to be categorical. Attribute vector $\boldsymbol{x} \in \{0, 1\}^N$ is the concatenation of D one-hot encoding vectors. $(D-2)$ categorical rating-relevant attributes form their own binary one-hot representations. The additional two one-hot vectors respectively represent ID's of users and items. As a special case of (13), TF fixes $D_m = D_M = D$ to build a single D-order interactions between attributes. Since weight function f_D in (14) allows individual dimensions K_j for each latent factor vector $\boldsymbol{v}_j$, TF defines a tensor $\mathcal{S} \in \mathbb{R}^{K_1 \times K_2 \times \ldots \times K_D}$ to exploit tensor product of all latent factor vectors. In sum, (13) is simplified as the following:

$$\begin{aligned}
\mu_R &= \sum_{j_1=1}^{N} \sum_{j_2=j_1+1}^{N} \cdots \sum_{j_D=j_{D-1}+1}^{N} f_D\left(\boldsymbol{v}_{j_1}, \boldsymbol{v}_{j_2}, \ldots, \boldsymbol{v}_{j_D}\right)\left(x_{j_1} x_{j_2} \ldots x_{j_D}\right) \\
&= f_D\left(\boldsymbol{v}_{l_1}, \boldsymbol{v}_{l_1}, \ldots, \boldsymbol{v}_{l_D}\right) \text{ as } x_{l_1} = x_{l_2} = \ldots = x_{l_D} = 1, \text{ other } x = 0 \\
&= \left\langle \mathcal{S}, \boldsymbol{v}_{l_1}, \boldsymbol{v}_{l_2}, \ldots, \boldsymbol{v}_{l_D} \right\rangle \\
&= \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \cdots \sum_{k_D=1}^{K_D} s_{k_1 k_2 \ldots k_D} v_{l_1 k_1} v_{l_2 k_2} \ldots v_{l_D k_D}
\end{aligned} \tag{16}$$

where function $f(\cdot) =< \cdot >$ denotes the tensor product. Note that attribute vectors $\boldsymbol{x}$ in TF must consist of exact C 1's due to one-hot encoding. Therefore there exists only match $j_1 = l_1, j_2 = l_2, \ldots, j_D = l_D$ where all the attributes in these positions are set to 1.

3.3.2 FM-EXTENDED MODELS

Factorization Machine (FM) (Rendle et al., 2011) allows numerical attributes $\boldsymbol{x} \in \mathbb{R}^N$ as input, including one-hot representations of users and items. Although higher order interactions between attributes could be formulated, FM focuses on at most second-order interactions. To derive FM from (13), let $0 = D_m \leq d \leq D_M = 2$ and $w_{j_1 j_2} = f_2(\boldsymbol{v}_{j_1}, \boldsymbol{v}_{j_2}) = \boldsymbol{v}_{j_1}^\top \boldsymbol{v}_{j_2}$ in (14) be applied for the second-order interaction. Then we begin to simplify (13):

$$\begin{aligned}\mu_R = & \underbrace{w_0}_{d=0} + \underbrace{\sum_{l=1}^{N} w_l x_l}_{d=1} + \underbrace{\sum_{j_1=1}^{N} \sum_{j_2=j_1+1}^{N} w_{j_1 j_2} (x_{j_1} x_{j_2})}_{d=2} \\ & = w_0 + \sum_{l=1}^{N} w_l x_l + \sum_{j_1=1}^{N} \sum_{j_2=j_1+1}^{N} \boldsymbol{v}_{j_1}^\top \boldsymbol{v}_{j_2} (x_{j_1} x_{j_2}) \end{aligned} \tag{17}$$

which is exactly the formulation of FM. Note that FM implicitly requires all the latent factor vectors $\boldsymbol{v}$ of the same dimension K; however the requirement could be released from the viewpoint of our general form (13). Models in this category mainly differs in two aspects. First, linear mapping can be replaced by deep neural networks, which allows non-linear mapping of attributes. Second, FM only extracts first-order, second-order interactions. Further works such as Cao et al. (2016) extracts higher-order interactions between attributes.

3.4 HETEROGENEOUS GRAPHS

We notice several relevant works that perform low-rank factorization or representation learning in heterogeneous graphs, such as (Yu et al., 2014; Zheng et al., 2016; Palumbo et al., 2017). The interactions of users and items can be represented by a heterogeneous graph of two node types. An edge is unweighted for implicit feedback, while weighted for explicit opinions. External attributes are typically leveraged by assigning them extra nodes in the heterogeneous graph. Heterogeneous graph structure is more suitable for categorical attributes, since each candidate value of attributes can be naturally assigned a node.

In heterogeneous graphs, recommendation can be viewed as a *link prediction* problem. Predicting a future rating corresponds to forecasting whether an edge will be built between user and item nodes. The existing works commonly adopt a two-stage algorithm to learn the model. At first, we perform a random walk or a meta-path algorithms to gather the similarities between users and items from a heterogeneous graph. The similarity information can be kept as multiple similarity matrices or network embedding vectors. Then a matrix factorization model or other supervised machine learning algorithms are applied to extract discriminative features from the gathered similarity information, which is used for future rating prediction.

3.5 MODEL DIFFERENCES

In our previous classification, there are still a number of works in each category. Although Models in the same category share similar mathematical form in terms of the design of objective function, but can vary in certain design aspect. One most important difference is the task they focus on. Some models emphasize on predicting future ratings. Therefore, they usually dedicated to minimize *Root Mean Square Error (RMSE)* to have a more accurate prediction on scores. Some other models care about top-N items that a user may like. Hence, they adopt pairwise ranking to predict the preference of items on a given user.A second difference is based on the types of attributes that are exploited. For example, Yang et al. (2011) takes a social network as its input feature matrices. A third difference is that each model claimed its source of attributes. Some models claim to accept only user attributes while others might be more general for different types of attributes.

4 EXPERIMENT AND FUTURE WORK

Although the major goal of this paper is to provide the mathematical interpretation to summarize the existing models, we still report some experimental results on rating prediction using the Movielens

dataset. 3.1, 3.2 and 3.3. The results are shown in table 2 and it seems FM model yields the superior results. Detailed descriptions are shown in the appendix. Future works include more complete empirical analysis and potentially a unified solution in this direction.

Table 2: RMSE on MovieLens-1M. TF:Karatzoglou et al. (2010); CMF:Singh & Gordon (2008); RLFM:Agarwal & Chen (2009); FIP:Yang et al. (2011); FM:Rendle et al. (2011); CDL:Wang et al. (2015). '*' means its memory usage exceeds machine's limit. '-' means it is not suitable in this task.

Rating	Attribute	TF	CMF	RLFM	FIP	FM	CDL
All (MF: 0.9002)	(1)	0.9315	0.9468	0.8815	0.9488	**0.8793**	-
	(2)	*	1.1671	0.8849	0.9482	**0.8824**	1.6013
	(3)	*	0.9436	0.8824	0.9368	**0.8798**	-
Without cold-start (MF: 0.8986)	(1)	0.9308	0.9435	0.8804	0.9482	**0.8782**	-
	(2)	*	1.1222	0.8840	0.9479	**0.8816**	1.5664
	(3)	*	0.9398	0.8813	0.9363	**0.8788**	-
Cold-start (MF: 1.0419)	(1)	0.9993	1.2215	0.9840	1.0030	**0.9792**	-
	(2)	*	3.3385	0.9672	0.9807	**0.9533**	3.6165
	(3)	*	1.2523	0.9848	0.9821	**0.9679**	-

In this paper, we discuss a novel viewpoint to the domain of recent attribute-aware recommender systems. Most of the proposed works try to make classical matrix factorization methods recongnize either user, item or rating-relevant attributes for recommendation improvement. Using systemmatical introduction to our general mathematical formulations, we review and cover the most popular model designs of a large number of attribute-aware recommender systems. It is expected to help new researchers understand this domain, while figure out the potential to propose new ideas from these model design categories or even not belonging to any of our category. This paper only focuses on the theoretical observations of attribute-aware recommender systems; however we also observe that most previous works prefer to apply certain datasets and much referenced matrix factorization extended approaches in their recommendation experiments. To our knowledge, among the exising survey works on attribute-aware recommender systems, there is only one work (Bobadilla et al., 2013) conducting experiments on *memory-based recommender systems*, but not model-based recommender systems that include matrix factorization. As our futre work, we would like to compare the empirical performance of several well-known matrix factorization extended models on benchmark datasets.

REFERENCES

Ryan Prescott Adams, George E. Dahl, and Iain Murray. Incorporating side information in probabilistic matrix factorization with gaussian processes. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, pp. 1–9, Arlington, Virginia, United States, 2010. AUAI Press. ISBN 978-0-9749039-6-5.

G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, June 2005. ISSN 1041-4347. doi: 10.1109/TKDE.2005.99.

Gediminas Adomavicius and Alexander Tuzhilin. *Context-Aware Recommender Systems*, pp. 217–253. Springer US, Boston, MA, 2011. ISBN 978-0-387-85820-3. doi: 10.1007/978-0-387-85820-3_7.

Deepak Agarwal and Bee-Chung Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pp. 19–28, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. doi: 10.1145/1557019.1557029.

Iman Barjasteh, Rana Forsati, Farzan Masrour, Abdol-Hossein Esfahanian, and Hayder Radha. Cold-start item and user recommendation with decoupled completion and transduction. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pp. 91–98, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3692-5. doi: 10.1145/2792838.2800196.

Mathieu Blondel, Akinori Fujino, and Naonori Ueda. *Convex Factorization Machines*, pp. 19–35. Springer International Publishing, Cham, 2015. ISBN 978-3-319-23525-7. doi: 10.1007/978-3-319-23525-7_2.

Mathieu Blondel, Masakazu Ishihata, Akinori Fujino, and Naonori Ueda. Polynomial networks and factorization machines: New insights and efficient training algorithms. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 850–858. JMLR.org, 2016.

J. Bobadilla, F. Ortega, A. Hernando, and A. GutiéRrez. Recommender systems survey. *Know.-Based Syst.*, 46:109–132, July 2013. ISSN 0950-7051. doi: 10.1016/j.knosys.2013.03.012.

Guillaume Bouchard, Dawei Yin, and Shengbo Guo. Convex collective matrix factorization. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, April 29 - May 1, 2013*, volume 31 of *JMLR Workshop and Conference Proceedings*, pp. 144–152. JMLR.org, 2013.

Thomas Brouwer and Pietro Liò. Bayesian hybrid matrix factorisation for data integration. In Aarti Singh and Xiaojin (Jerry) Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pp. 557–566. PMLR, 2017.

Bokai Cao, Hucheng Zhou, Guoqiang Li, and Philip S. Yu. Multi-view machines. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pp. 427–436, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3716-8. doi: 10.1145/2835776.2835777.

Chaochao Chen, Xiaolin Zheng, Yan Wang, Fuxing Hong, and Zhen Lin. Context-aware collaborative topic regression with social matrix factorization for recommender systems. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pp. 9–15. AAAI Press, 2014.

Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. Svdfeature: A toolkit for feature-based collaborative filtering. *J. Mach. Learn. Res.*, 13(1):3619–3622, December 2012. ISSN 1532-4435.

Chen Cheng, Fen Xia, Tong Zhang, Irwin King, and Michael R. Lyu. Gradient boosting factorization machines. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pp. 265–272, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2668-1. doi: 10.1145/2645710.2645730.

Wei-Sheng Chin, Bo-Wen Yuan, Meng-Yuan Yang, Yong Zhuang, Yu-Chin Juan, and Chih-Jen Lin. Libmf: A library for parallel matrix factorization in shared-memory systems. *J. Mach. Learn. Res.*, 17(1):2971–2975, January 2016. ISSN 1532-4435.

Szu-Yu Chou, Yi-Hsuan Yang, Jyh-Shing Roger Jang, and Yu-Ching Lin. Addressing cold start for next-song recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pp. 115–118, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4035-9. doi: 10.1145/2959100.2959156.

Xin Dong, Lei Yu, Zhonghuo Wu, Yuxia Sun, Lingfeng Yuan, and Fangxi Zhang. A hybrid collaborative filtering model with deep structure for recommender systems. In Satinder P. Singh and Shaul Markovitch (eds.), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pp. 1309–1315. AAAI Press, 2017.

Liang Du, Xuan Li, and Yi-Dong Shen. User graph regularized pairwise matrix factorization for item recommendation. In *Proceedings of the 7th International Conference on Advanced Data Mining and Applications - Volume Part II*, ADMA'11, pp. 372–385, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-25855-8. doi: 10.1007/978-3-642-25856-5_28.

Yi Fang and Luo Si. Matrix co-factorization for recommendation with rich side information and implicit feedback. In *Proceedings of the 2Nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, pp. 65–69, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1027-7. doi: 10.1145/2039320.2039330.

Yuhong Guo Feipeng Zhao. Learning discriminative recommendation systems with side information. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 3469–3475, 2017. doi: 10.24963/ijcai.2017/485.

Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Content-aware point of interest recommendation on location-based social networks. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 1721–1727. AAAI Press, 2015. ISBN 0-262-51129-0.

Hancheng Ge, James Caverlee, and Haokai Lu. Taper: A contextual tensor-based approach for personalized expert recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pp. 261–268, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4035-9. doi: 10.1145/2959100.2959151.

Mehmet Gönen, Suleiman A. Khan, and Samuel Kaski. Kernelized bayesian matrix factorization. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pp. III–864–III–872. JMLR.org, 2013.

Quanquan Gu, Jie Zhou, and Chris H. Q. Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA*, pp. 199–210. SIAM, 2010. doi: 10.1137/1.9781611972801.18.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for CTR prediction. In Carles Sierra (ed.), *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pp. 1725–1731. ijcai.org, 2017. doi: 10.24963/ijcai.2017/239.

Yuhong Guo. Convex co-embedding for matrix completion with predictive side information. In Satinder P. Singh and Shaul Markovitch (eds.), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pp. 1955–1961. AAAI Press, 2017.

Ruining He and Julian McAuley. Vbpr: Visual bayesian personalized ranking from implicit feedback. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 144–150. AAAI Press, 2016.

Xiangnan He and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pp. 355–364, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5022-8. doi: 10.1145/3077136.3080777.

Balázs Hidasi. Context-aware preference modeling with factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pp. 371–374, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3692-5. doi: 10.1145/2792838.2796543.

Balázs Hidasi and Domonkos Tikk. Fast als-based tensor factorization for context-aware recommendation from implicit feedback. In *Proceedings of the 2012 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, ECML PKDD'12, pp. 67–82, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-33485-6. doi: 10.1007/978-3-642-33486-3_5.

F.O. Isinkaye, Y.O. Folajimi, and B.A. Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3):261 – 273, 2015. ISSN 1110-8665. doi: http://dx.doi.org/10.1016/j.eij.2015.06.005.

Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. Field-aware factorization machines for ctr prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pp. 43–50, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4035-9. doi: 10.1145/2959100.2959134.

Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pp. 79–86, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-906-0. doi: 10.1145/1864708.1864727.

Hiroyuki Kasai and Bamdev Mishra. Low-rank tensor completion: A riemannian manifold preconditioning approach. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 1012–1021. JMLR.org, 2016.

Yong-Deok Kim and Seungjin Choi. Scalable variational bayesian matrix factorization with side information. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, volume 33 of *JMLR Workshop and Conference Proceedings*, pp. 493–502. JMLR.org, 2014.

Yehuda Koren and Robert Bell. *Advances in Collaborative Filtering*, pp. 145–186. Springer US, Boston, MA, 2011. ISBN 978-0-387-85820-3. doi: 10.1007/978-0-387-85820-3_5.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009. ISSN 0018-9162. doi: 10.1109/MC.2009.263.

Sheng Li, Jaya Kawale, and Yun Fu. Deep collaborative filtering via marginalized denoising auto-encoder. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pp. 811–820, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3794-6. doi: 10.1145/2806416.2806527.

Xiaopeng Li and James She. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pp. 305–314, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4887-4. doi: 10.1145/3097983.3098077.

Yanen Li, Jia Hu, ChengXiang Zhai, and Ye Chen. Improving one-class collaborative filtering by incorporating rich user information. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pp. 959–968, New York, NY, USA, 2010a. ACM. ISBN 978-1-4503-0099-5. doi: 10.1145/1871437.1871559.

Yize Li, Jiazhong Nie, Yi Zhang, Bingqing Wang, Baoshi Yan, and Fuliang Weng. Contextual recommendation based on text mining. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pp. 692–700, Stroudsburg, PA, USA, 2010b. Association for Computational Linguistics.

Chun-Ta Lu, Lifang He, Weixiang Shao, Bokai Cao, and Philip S. Yu. Multilinear factorization machines for multi-task multi-view learning. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pp. 701–709, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4675-7. doi: 10.1145/3018661.3018716.

Jin Lu, Guannan Liang, Jiangwen Sun, and Jinbo Bi. A sparse interactive model for matrix completion with side information. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 4071–4079, 2016.

Hao Ma, Tom Chao Zhou, Michael R. Lyu, and Irwin King. Improving recommender systems by incorporating social contextual information. *ACM Trans. Inf. Syst.*, 29(2):9:1–9:23, April 2011. ISSN 1046-8188. doi: 10.1145/1961209.1961212.

Aditya Krishna Menon and Charles Elkan. A log-linear model with latent features for dyadic prediction. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pp. 364–373, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4256-0. doi: 10.1109/ICDM.2010.148.

Aditya Krishna Menon, Krishna-Prasad Chitrapura, Sachin Garg, Deepak Agarwal, and Nagaraj Kota. Response prediction using collaborative filtering with hierarchies and side-information. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pp. 141–149, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0813-7. doi: 10.1145/2020408.2020436.

Nagarajan Natarajan and Inderjit S. Dhillon. Inductive matrix completion for predicting genedisease associations. *Bioinformatics*, 30(12):i60–i68, 2014. doi: 10.1093/bioinformatics/btu269.

Trung V. Nguyen, Alexandros Karatzoglou, and Linas Baltrunas. Gaussian process factorization machines for context-aware recommendations. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pp. 63–72, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2257-7. doi: 10.1145/2600428.2609623.

Enrico Palumbo, Giuseppe Rizzo, and Raphaël Troncy. Entity2rec: Learning user-item relatedness from knowledge graphs for top-n item recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, pp. 32–36, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4652-8. doi: 10.1145/3109859.3109889.

Sunho Park, Yong-Deok Kim, and Seungjin Choi. Hierarchical bayesian matrix factorization with side information. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pp. 1593–1599. AAAI Press, 2013. ISBN 978-1-57735-633-2.

Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. 2007.

Ian Porteous, Arthur Asuncion, and Max Welling. Bayesian matrix factorization with side information and dirichlet process mixtures. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pp. 563–568. AAAI Press, 2010.

Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pp. 635–644, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2010002.

Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pp. 1257–1264, USA, 2007. Curran Associates Inc. ISBN 978-1-60560-352-0.

Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 880–887, New York, NY, USA, 2008a. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390267.

Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 880–887, New York, NY, USA, 2008b. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390267.

Martin Saveski and Amin Mantrach. Item cold-start recommendations: Learning local collective embeddings. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pp. 89–96, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2668-1. doi: 10.1145/2645710.2645751.

Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, Lexing Xie, and Darius Braziunas. Low-rank linear cold-start recommendation from social data. In Satinder P. Singh and Shaul Markovitch (eds.), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pp. 1502–1508. AAAI Press, 2017.

Hanhuai Shan and Arindam Banerjee. Generalized probabilistic matrix factorizations for collaborative filtering. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pp. 1025–1030, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4256-0. doi: 10.1109/ICDM.2010.116.

Yue Shi, Martha Larson, and Alan Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv.*, 47(1):3:1–3:45, May 2014. ISSN 0360-0300. doi: 10.1145/2556270.

Donghyuk Shin, Suleyman Cetintas, Kuang-Chih Lee, and Inderjit S. Dhillon. Tumblr blog recommendation with boosted inductive matrix completion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pp. 203–212, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3794-6. doi: 10.1145/2806416.2806578.

Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pp. 650–658, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401969.

David Stern, Ralf Herbrich, and Thore Graepel. Matchbox: Large scale bayesian recommendations. In *Proceedings of the 18th International World Wide Web Conference*, January 2009.

Zebang Shen Chao Zhang Congfu Xu Tengfei Zhou, Hui Qian. Tensor completion with side information: A riemannian manifold approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 3539–3545, 2017. doi: 10.24963/ijcai.2017/495.

K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, and E. Duval. Context-aware recommender systems for learning: A survey and future challenges. *IEEE Transactions on Learning Technologies*, 5(4):318–335, Oct 2012. ISSN 1939-1382. doi: 10.1109/TLT.2012.11.

Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pp. 1235–1244, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3664-2. doi: 10.1145/2783258.2783273.

Hao Wang, Xingjian Shi, and Dit-Yan Yeung. Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 415–423, 2016.

Miao Xu, Rong Jin, and Zhi-Hua Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pp. 2301–2309, USA, 2013. Curran Associates Inc.

Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. Like like alike: Joint friendship and interest propagation in social networks. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pp. 537–546, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0632-4. doi: 10.1145/1963405.1963481.

Jiho Yoo and Seungjin Choi. Bayesian matrix co-factorization: Variational algorithm and cramér-rao bound. In *Proceedings of the 2011th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part III*, ECMLPKDD'11, pp. 537–552, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-23807-9. doi: 10.1007/978-3-642-23808-6_35.

Hsiang-Fu Yu, Hsin-Yuan Huang, Inderjit S. Dhillon, and Chih-Jen Lin. A unified algorithm for one-cass structured matrix factorization with side information. In Satinder P. Singh and Shaul Markovitch (eds.), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pp. 2845–2851. AAAI Press, 2017.

Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pp. 283–292, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2351-2. doi: 10.1145/2556195.2556259.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 353–362, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939673.

Feipeng Zhao, Min Xiao, and Yuhong Guo. Predictive collaborative filtering with side information. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pp. 2385–2390. AAAI Press, 2016. ISBN 978-1-57735-770-4.

Jing Zheng, Jian Liu, Chuan Shi, Fuzhen Zhuang, Jingzhi Li, and Bin Wu. Dual similarity regularization for recommendation. In *Proceedings, Part II, of the 20th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Volume 9652*, PAKDD 2016, pp. 542–554, New York, NY, USA, 2016. Springer-Verlag New York, Inc. ISBN 978-3-319-31749-6. doi: 10.1007/978-3-319-31750-2_43.

Tinghui Zhou, Hanhuai Shan, Arindam Banerjee, and Guillermo Sapiro. Kernelized probabilistic matrix factorization: Exploiting graphs and side information. In *Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim, California, USA, April 26-28, 2012.*, pp. 403–414. SIAM / Omnipress, 2012. doi: 10.1137/1.9781611972825.35.

A Preliminary: Collaborative Filtering and Matrix Factorization

Collaborative filtering (CF) has become the most prevailing technique to realize recommender systems in recent years (Adomavicius & Tuzhilin, 2005; Shi et al., 2014; Adomavicius & Tuzhilin, 2011; Isinkaye et al., 2015). It assumes preferences that users exhibit towards interacted items can be generalized and infer their preferences towards items they never interact with. CF aims to infer a set of items that a user prefers the most but never interacts with from records of other users with similar preferences. This section briefly introduces conventional CF denoting CF techniques that only take advantage of user-item interactions, or rating matrix $\boldsymbol{R}$. In practice, they are commonly categorized into *memory-based* CF and *model-based* CF (Shi et al., 2014; Isinkaye et al., 2015; Adomavicius & Tuzhilin, 2005).

Matrix factorization (MF) (Shi et al., 2014; Koren et al., 2009; Paterek, 2007; Koren & Bell, 2011), in the basic form, represents each user u as a parameter vector $\boldsymbol{w}_u \in \mathbb{R}^K$ and each item i as $\boldsymbol{h}_i \in \mathbb{R}^K$, where K is the dimension of latent factors. The prediction of user u's rating or preference towards item i, denoted as $\hat{r}_{ui}$, can be computed using inner product:

$$\hat{r}_{ui} = \boldsymbol{w}_u^\top \boldsymbol{h}_i, \tag{18}$$

which captures the interaction between them. MF seeks to generate rating predictions as close as possible to those recorded ratings. In matrix form, it can be written as finding $\boldsymbol{W}, \boldsymbol{H}$ such that $\boldsymbol{R} \approx \boldsymbol{W}^\top \boldsymbol{H}$ where $\boldsymbol{R} \in \mathbb{R}^{N_u \times N_i}$. MF is essentially learning a low-rank approximation of the rating matrix since the dimension of representations K is usually much smaller than the number of users N_u and items N_i. To learn the latent factors of users and items, the system tries to find $\boldsymbol{W}, \boldsymbol{H}$ that minimize the regularized square error on the set of known ratings $\delta(\boldsymbol{R})$:

$$(\boldsymbol{W}^*, \boldsymbol{H}^*) = \underset{\boldsymbol{W},\boldsymbol{H}}{\operatorname{argmin}} \sum_{(u,i)\in\delta(\boldsymbol{R})} \frac{1}{2}\left(r_{ui} - \boldsymbol{w}_u^\top \boldsymbol{h}_i\right)^2 + \frac{\lambda_W}{2}\sum_{u=1}^{N_u} \|\boldsymbol{w}_u\|_2^2 + \frac{\lambda_H}{2}\sum_{i=1}^{N_i} \|\boldsymbol{h}_i\|_2^2, \tag{19}$$

where λ_W and λ_H are regularization parameters. MF tends to cluster users or items with similar rating configuration into groups in the latent factor space which implies that similar users or items will be close to each other. Furthermore, MF assumes the rank of rating matrix $\boldsymbol{R}$ or the dimension

of the vector space generated by rating configuration of users is far smaller than the number of users N_u. This implies that each user's rating configuration can be obtained by a linear combination of ratings from a group of other users since they are all generated by K principle vectors. Thus MF entails the spirit of collaborative filtering, which is to infer a user's unknown ratings by ratings of several other users.

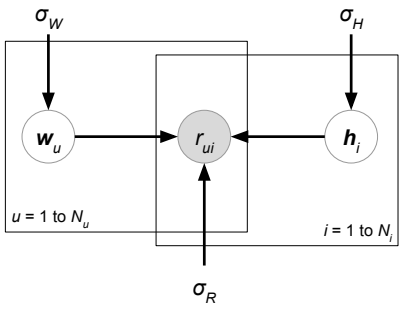


Figure 5: Graphical interpretation of Probabilistic Matrix Factorization (PMF). User or item latent factors $\boldsymbol{W}, \boldsymbol{H}$ are put to generate observed ratings $\boldsymbol{R}$. Parameters $\sigma_W, \sigma_H, \sigma_R$ control the certainty in the generation process.

Probabilistic matrix factorization (PMF, Figure 5) (Salakhutdinov & Mnih, 2007; 2008b) is a probabilistic linear model with observed Gaussian noise and can be viewed as a probabilistic extension of MF. PMF adopts the assumption that users and items are independent and represents each user or each item with a zero-mean spherical multivariate Gaussian distribution as follows:

$$p\left(\boldsymbol{W} \mid \sigma_W^2\right) = \prod_{u=1}^{N_u} \mathcal{N}\left(\boldsymbol{w}_u \mid \boldsymbol{0}, \sigma_W^2 \boldsymbol{I}\right), \qquad p\left(\boldsymbol{H} \mid \sigma_H^2\right) = \prod_{i=1}^{N_i} \mathcal{N}\left(\boldsymbol{h}_i \mid \boldsymbol{0}, \sigma_H^2 \boldsymbol{I}\right), \tag{20}$$

where σ_W^2 and σ_H^2 are observed user-specific and item-specific noise. PMF then formulates the conditional probability over the observed ratings as

$$p\left(\boldsymbol{R} \mid \boldsymbol{W}, \boldsymbol{H}, \sigma^2\right) = \prod_{(i,j) \in \delta(\boldsymbol{R})} \mathcal{N}\left(r_{ui} \mid \boldsymbol{w}_u^\top \boldsymbol{h}_i, \sigma_R^2\right), \tag{21}$$

where $\delta(\boldsymbol{R})$ is the set of known ratings and $\mathcal{N}(x \mid \mu, \sigma^2)$ denotes the Gaussian distribution with mean μ and variance σ^2. Learning of PMF is conducted by maximum a posteriori (MAP) estimation, which is equivalent to maximize the log of the posterior distribution of $\boldsymbol{W}, \boldsymbol{H}$:

$$\begin{aligned}
\log p\left(\boldsymbol{W}, \boldsymbol{H} \mid \boldsymbol{R}, \sigma_R^2, \sigma_W^2, \sigma_H^2\right) &= \log p\left(\boldsymbol{R} \mid \boldsymbol{W}, \boldsymbol{H}, \sigma_R^2\right) + \log p\left(\boldsymbol{W} \mid \sigma_W^2\right) + \log p\left(\boldsymbol{H} \mid \sigma_H^2\right) + C \\
&= -\frac{1}{2\sigma_R^2} \sum_{(u,i) \in \delta(\boldsymbol{R})} \left(r_{ui} - \boldsymbol{w}_u^\top \boldsymbol{h}_i\right)^2 - \frac{1}{2\sigma_W^2} \sum_{u=1}^{N_u} \boldsymbol{w}_u^\top \boldsymbol{w}_u - \frac{1}{2\sigma_H^2} \sum_{i=1}^{N_i} \boldsymbol{h}_i^\top \boldsymbol{h}_i \\
&\quad - \frac{1}{2}\Big(|\delta(\boldsymbol{R})| \log \sigma_R^2 + N_u K \log \sigma_W^2 + N_i K \log \sigma_H^2\Big) + C
\end{aligned} \tag{22}$$

where C is a constant independent of all parameters and K is the dimension of user or item representations. With Gaussian noise $\sigma_R^2, \sigma_W^2, \sigma_H^2$ observed, maximizing the log-posterior is identical to minimize the objective function with the form:

$$\sum_{(u,i) \in \delta(\boldsymbol{R})} \frac{1}{2}(r_{ui} - \boldsymbol{w}_u^\top \boldsymbol{h}_i)^2 + \frac{\lambda_W}{2} \sum_{u=1}^{N_u} \|\boldsymbol{w}_u\|_2^2 + \frac{\lambda_H}{2} \sum_{i=1}^{N_i} \|\boldsymbol{h}_i\|_2^2, \tag{23}$$

where $\lambda_W = \sigma_R^2/\sigma_W^2, \lambda_H = \sigma_R^2/\sigma_H^2$. Note that (23) has exactly the same form as the regularized square error of MF and gradient descent or its extensions can then be applied in training PMF.

B EXPERIMENT DETAILS

We evaluate the effectiveness of each model by examining their performance on MovieLens-1M. We focus on the *rating prediction* task since the majority of models have their objectives designed for this task. We also compare the performance of each competitor under different conditions: with/without user-relevant attributes or item-relevant attributes. Hyperparameters for each model are tuned based on grid search.

We consider several popular models for comparison: Tensor Factorization (TF) Karatzoglou et al. (2010), Collective Matrix Factorization (CMF) Singh & Gordon (2008), Regression-based Latent Factor Model (RLFM) Agarwal & Chen (2009), Friendship-Interest Propagation (FIP) Yang et al. (2011), Factorization Machine (FM) Rendle et al. (2011) and Collaborative Deep Learning (CDL) Wang et al. (2015). We also select Matrix Factorization (MF) Chin et al. (2016) as the baseline model that does not include any attribute. The attribute types that each model accepts are concluded in Table 3.

Table 3: Attribute types that claimed to be used for each model.

Model	User-relevant attributes	Item-relevant attributes	Rating-relecant attributes
TF	✓	✓	✓
CMF	✓	✓	
RLFM	✓	✓	✓
FIP	✓	✓	
FM	✓	✓	✓
CDL		✓	
MF			

Table 4: Notations referring to attribute type combinations used in an experiment case.

Type	User attributes	Item attributes	Rating attributes
(1)	✓		
(2)		✓	
(3)	✓	✓	

B.1 DATASET

MovieLens-1M contain ratings that users give to different movies. It also includes some user information, such as genre, age and occupation, and item information, for example the category a movie belongs to and the year when the movie was produced. Training set and test set are divided by the time that the rating was generated. The latest 10% ratings serve as test set while the others are served as train set.

B.2 EVALUATION METRIC

Adopted by the experiments in these baseline models, *Root Mean Square Error (RMSE)* is selected as the evaluation metric in our experiments. By our observation, RMSE is the most widely used evaluation metric for rating prediction, since most of model-based collaborative filtering methods try to minimize MSE (RMSE without root) as their objectives, including all of our experimented models. In our opinions, it is fair to test all the baseline models using the evaluation metric they all try to optimize.

B.3 COLD-START SETTING

Cold-start is a special case that many recommend systems are designed to deal with. In practical use, it is difficult to recommend items to a user especially when the user has few or even no past rating records. Since it is an important issue to deal with in the real world, we want to compare different models under this condition. Instead of extracting a new train set designed for cold-start setting (for

example, a set formed by randomly reducing the size of the original train set until number of ratings for each user is less than a specific amount), we simulate the cold-start situation by evaluating the performance of a new test set. The new test set is formed by repeatedly extracting all test instances of a user from the original test set where the user has few ratings in train set. The extracting procedure halts when the size of the new test set reaches a threshold. The threshold is set to 1000 in our experiment setting. The other ratings that are not extracted form another set, called "without cold-start" in the following, to compare the result with cold-start. Compared with extracting a new train set, this evaluation metric saves the time to train a new dataset while preserving cold-start property.