International Journal of Approximate Reasoning 53 (2012) 447–466

Contents lists available at SciVerse ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

A latent model for collaborative filtering

Helge Langseth [a,*], Thomas Dyhre Nielsen [b]

[a] Department of Computer and Information Science, The Norwegian University of Science and Technology, Trondheim, Norway
[b] Department of Computer Science, Aalborg University, Aalborg, Denmark

ARTICLE INFO

Article history:
Received 2 May 2011
Received in revised form 11 November 2011
Accepted 14 November 2011
Available online 20 November 2011

Keywords:
Recommender systems
Collaborative filtering
Graphical models
Latent variables

ABSTRACT

Recommender systems based on collaborative filtering have received a great deal of interest over the last two decades. In particular, recently proposed methods based on dimensionality reduction techniques and using a symmetrical representation of users and items have shown promising results. Following this line of research, we propose a probabilistic collaborative filtering model that explicitly represents all items and users simultaneously in the model. Experimental results show that the proposed system obtains significantly better results than other collaborative filtering systems (evaluated on the MovieLens data set). Furthermore, the explicit representation of all users and items allows the model to, e.g. make group-based recommendations balancing the preferences of the individual users.

1. Introduction

Recommender systems are designed to help users cope with vast amounts of information. They do so by presenting only a certain subset of items that is believed to be relevant for the user. These types of systems are usually grouped into two categories: content-based systems make recommendations based on a user preference model that combines the user's ratings with, e.g. content information and textual descriptions of the items. Collaborative filtering uses the ratings of like-minded users to make recommendations for the user in question.

Over the last decade recommender systems based on collaborative filtering have enjoyed a great deal of interest. Collaborative filtering systems are often characterized as either being model-based or memory-based [5], although hybrid systems have also been developed [42]. Roughly speaking, memory-based algorithms use the whole database of user ratings and rely on a distance function to measure user similarity. On the other hand, model-based algorithms learn a model for user preferences, which is subsequently used to predict a user's rating for a particular item that he or she has not seen before.

The simplest type of model-based algorithms uses a multinomial mixture model (corresponding to a naive Bayesian network [13]) for either grouping users into user-groups or items into item-categories. More elaborate model-based algorithms have also been developed, having both probabilistic (see, e.g. [52]) and non-probabilistic foundations (see [50] for one example). In particular, where earlier model classes relied on a single item-model and/or user-model for predicting preferences, more recently proposed model classes combine these two perspectives and treat users and items symmetrically by representing them explicitly in the model. In this paper we pursue this idea further and propose a new type of probabilistic graphical model (represented by a linear Gaussian Bayesian network) for collaborative filtering. The model explicitly includes all users and items simultaneously in the model, and can therefore also be seen as a relational probabilistic model combining an item perspective and a user perspective [54]. The generative properties of the model support a natural model interpretation, and by having all users represented in the same model, the system can provide joint recommendations for several users. Empirical results based on the MovieLens data set and the Jester data set demonstrate that the proposed model outperforms other memory-based and model-based approaches.

* Corresponding author.
E-mail addresses: helgel@idi.ntnu.no (H. Langseth), tdn@cs.aau.dk (T.D. Nielsen).

0888-613X/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ijar.2011.11.002

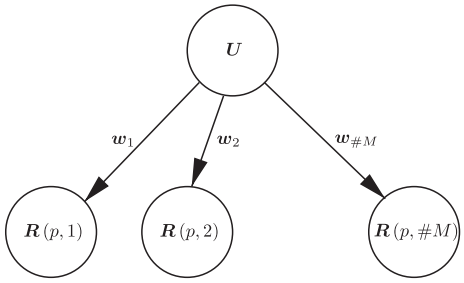


Fig. 1. The user-based perspective on a collaborative filtering model.

The remainder of the paper is structured as follows. In Section 2 we introduce Bayesian networks; the statistical modeling framework that will be used throughout the paper. Related research is explored in Section 3, before our model is presented in Section 4. An algorithm for learning the proposed model from data is described in Section 5, and we investigate its predictive ability in Section 6. In Section 7 we conclude and give directions for future research.

2. Bayesian networks

A Bayesian network [41,24] is a probabilistic graphical model that provides a compact representation of a joint probability distribution and supports efficient probability updating.

A Bayesian network (BN) over a set of variables $\{X_1, \ldots, X_n\}$ consists of both a qualitative part and a quantitative part. The qualitative part is represented by an acyclic directed graph (traditionally abbreviated DAG) $G = (\mathcal{V}, \mathcal{E})$, where the nodes $\mathcal{V}$ represent the random variables $\{X_1, \ldots, X_n\}$ and the links $\mathcal{E}$ specify direct dependencies between the variables. An example of the qualitative part of a BN is shown in Fig. 1. Since there is a one-to-one correspondence between the nodes in the network and the corresponding random variables, we shall use the terms node and variable interchangeably. Considering $\mathcal{E}$, we call the nodes with outgoing edges pointing into a specific node X the parents of X (denoted $\boldsymbol{\pi}_X$), and we say that a variable X_j is a descendant of X_i if and only if there exists a directed path from X_i to X_j in the graph. The edges in the graph encode (in)dependencies between the variables, and, in particular, the assertion that a variable is conditionally independent of its non-descendants given its parents.

The quantitative part of a BN consists of conditional probability distributions or density functions s.t. each node is assigned one (and only one) probability distribution/density function conditioned on its parents. In the remainder of this paper we shall assume that all variables are continuous, and that each variable X_i with parents $\boldsymbol{\pi}_i$ is assigned a conditional linear Gaussian distribution:

$$f(x_i|\boldsymbol{\pi}_i) = \mathcal{N}(\mu_i + \boldsymbol{w}_i^{\mathsf{T}}\boldsymbol{\pi}_i, \sigma_i),$$

i.e., the mean value is given as a weighted linear combination of the values of the parent variables and the variance is fixed. The underlying conditional independence assumptions encoded in the BN allow us to calculate the joint probability function as

$$f(x_1, \ldots, x_n) = \prod_{i=1}^{n} f(x_i|\boldsymbol{\pi}_i)$$

and with linear Gaussian distributions assigned to all the variables it follows that the joint distribution is a multivariate Gaussian distribution. The precision matrix (the inverse of the covariance matrix) for this multivariate distribution directly reflects the independencies encoded in the BN; the entry for a pair of variables is zero if and only if the two variables are conditionally independent given the other variables in the network.

3. Model-based collaborative filtering

Probabilistic graphical models for collaborative filtering include general unconstrained models such as standard Bayesian networks [5] and dependency networks [18]. These types of models have, however, received only modest attention in the collaborative filtering community, mainly due to the complexity issues involved in learning these models from data. Instead research has focused on models, which explicitly incorporate certain independence and generative assumptions about the domain being modeled.

The most simple probabilistic model for collaborative filtering is the multinomial mixture model [5], where like-minded users are clustered together in the same user classes, and given a user class a user's ratings are assumed independent (i.e., the model basically corresponds to a naive Bayes model [13]). The independence assumptions underlying the multinomial mixture model do usually not hold, and have been studied extensively, in particular w.r.t. models targeted towards

classification [12,30]. However, for collaborative filtering the model has mainly been analyzed w.r.t. its generative properties: The multinomial mixture model assumes that all users have the same prior distribution over the user classes, and given that a user is assigned to a certain class, that class is used to predict ratings for all items.

The aspect model [20–22] addresses some of the inherent limitations of the mixture model by allowing users to have different prior distributions over the user classes.[1] This idea is further pursued in [34], which introduces the user rating profile (URP) model that expands on the generative semantics of the aspect model, and allows different latent classes to be associated with different item ratings. The URP model shares the same computational difficulties as the latent Dirichlet allocation model [4], and relies on approximate methods like variational methods or Gibbs sampling for inference and parameter learning. This model has been further explored in [47] that extends the latent model structure to cover both users and items. The joint modeling of users and items is also found in low-rank matrix approximation methods, where the user-item rating matrix is represented in factorized form as a product of a user-matrix and an item-matrix. Such factorized representations can be obtained using singular value decompositions (SVD) based methods that support missing entries in the rating matrix [50]. Recently, probabilistic extensions to the SVD-based methods have also been proposed to address the problem of over-fitting. This is realized by assigning suitable prior distributions to the model parameters, thereby achieving a form of regularization [33,45].

There has also been investigations into so-called hybrid recommendation systems, where recommendations are based on a unification of collaborative and content-based information. For example, Pennock et al. [42] proposed a personality diagnosis method, which can be seen as combining memory-based and model-based approaches; a naive Bayes model is used to calculate the probability that the active user is of the same personality type as other users. Wanger et al. [53] proposed a method for unifying the user-based and item-based collaborative filtering approaches within a memory-based context, [51] combined content-based filtering and collaborative filtering in a conditional Markov random field model, and [15] considered methods for integrating content information based on a weighted non-negative matrix factorization [6].

Finally, collaborative filtering has also received attention within the relational learning community. Notably, and which structure-wise is somewhat related to the model we propose in this paper, is the infinite hidden relational model [54]. In this model, there is a latent variable associated with each entity in the domain, and this latent variable appears as parent of all attributes of that entity as well as of the attributes of the relations in which the entity participates. As will become apparent later, the model proposed in this paper shares some similarities with this relational structure. It should be noted, though, that the infinite relational model is not specifically targeted towards collaborative filtering, but rather relational domains in general.

4. A mixed generative model

In this section we will describe our collaborative filtering model, but first we need to introduce some notation. We will denote the matrix of ratings by $\boldsymbol{R}$, which is of size $\#U \times \#M$; $\#U$ is the number of users and $\#M$ is the number of movies that are rated. $\boldsymbol{R}$ is sparsely filled, meaning that it (to a large degree) contains missing values. The observed ratings are either realizations of ordinal variables (discrete variables with ordered states, e.g. "`Bad`", "`Medium`", "`Good`") or real numbers. In the following we will consider only continuous ratings (ratings given as ordinal variables are hence assumed to have been translated into a numeric scale).

We will use p as the index of an arbitrary person using the system, i is the index of an item that can be rated, and $\boldsymbol{R}(p,i)$ is therefore the rating that person p gives item i. We will use the indicator function $\delta(p,i)$ to show whether or not person p has rated item i: $\delta(p,i) = 1$ if the rating exists, otherwise $\delta(p,i) = 0$. Furthermore, $\mathcal{I}(p)$ is the set of items that person p has rated, i.e., $\mathcal{I}(p) = \cup_{i:\delta(p,i)\neq 0}\{i\}$, and similarly we let $\mathcal{P}(i) = \cup_{p:\delta(p,i)\neq 0}\{p\}$ be the persons who have rated item i. As usual, lowercase letters are used to signify that a random variable is observed, so $\boldsymbol{r}(p,i)$ is the rating that p has given item i (that is, $\delta(p,i) = 1$ in this case). We abuse notation slightly and let $\boldsymbol{r}(p,\mathcal{I}(p))$ and $\boldsymbol{r}(\mathcal{P}(i),i)$ denote all the ratings given by person p and to item i, respectively. Finally, we let $\boldsymbol{r}$ denote all observed ratings (the part of $\boldsymbol{R}$ that is not missing).

When working in model-based CF, we search for a representation of $\boldsymbol{r}$ based on model parameters $\boldsymbol{\theta}_{\boldsymbol{r}}$, i.e., we assume the existence of a function $g(\cdot)$ s.t. $\boldsymbol{r}(p,i) = g(\boldsymbol{\theta}_{\boldsymbol{r}}, p, i)$ for all the observed ratings. By the inductive learning principle we will predict the rating a person p' gives to item i', $\boldsymbol{R}(p',i')$, as $g(\boldsymbol{\theta}_{\boldsymbol{r}}, p', i')$. This process is called *single-rating predictions*. Often, $g(\cdot)$ will be based on a statistical model of the conditional distribution of $\boldsymbol{R}(p,i)\,|\{\boldsymbol{r}, \boldsymbol{\theta}_{\boldsymbol{r}}\}$, and the prediction is then either the expected value or the median value of that conditional distribution.[2] A more complicated problem is *multi-rating predictions* (see, e.g. [23] for an overview): One may, for instance, want to find items that a group of users (persons p_1 and p_2, say) will enjoy together. A naive solution to the current example is to consider the multi-rating problem as a collection of single-rating problems, and then use $g(\boldsymbol{\theta}_{\boldsymbol{r}}, p_1, i) + g(\boldsymbol{\theta}_{\boldsymbol{r}}, p_2, i)$ to score item i. In practice, one would, however, often need to rank items in a more sophisticated way, i.e., by using a non-linear function of $\boldsymbol{R}(p_1,i)$ and $\boldsymbol{R}(p_2,i)$ (e.g. $\min(\boldsymbol{R}(p_1,i), \boldsymbol{R}(p_2,i))$). Doing so imposes further requirements on the model $g(\cdot)$ as the evaluation must take the correlation between the different predictions into consideration.

[1] For a comparison and discussion on alternative models, including the aspect model and the flexible mixture model [49], see [25].

[2] See [35] for a discussion of the relative merits of these estimators.

4.1. A data compression model

One of the more popular approaches for building CF systems is *data compression*, i.e., to find a representation $g(\boldsymbol{\theta}_{\boldsymbol{r}}, \cdot, \cdot)$ that is more compact than representing the original $\#U \times \#M$-matrix $\boldsymbol{R}$. Data compression techniques were pioneered in the late 1990s [2,43,16,46], and is still a major component of most state-of-the-art CF systems (see, e.g. [44,45,29]).

The first data compression approach we will describe assumes the existence of two matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ of size $q \times \#U$ and $q \times \#M$, respectively for some fixed q (i.e. $\boldsymbol{\theta}_{\boldsymbol{r}} = \{\boldsymbol{V}, \boldsymbol{W}\}$), and chooses $\boldsymbol{\theta}_{\boldsymbol{r}}$ s.t. $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{W}$ is the best rank-q approximation of $\boldsymbol{R}$. Here $q \leq \min(\#U, \#M)$ defines the granularity of the approximation. If we choose $q = \min(\#U, \#M)$ we will be able to recover the matrix $\boldsymbol{R}$, but typically $q \ll \min(\#U, \#M)$ is chosen in applications. For ease of later notation, we will consider $\boldsymbol{V}$ as consisting of $\#U$ column-vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{\#U}$ (each of length q), and similarly $\boldsymbol{W}$ as consisting of $\#M$ column-vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{\#M}$, again each vector is of length q. With this notation we have $g(\boldsymbol{\theta}_{\boldsymbol{r}}, p, i) = \boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{w}_i$. Note that we have one vector $\boldsymbol{w}_i$ per item i and one vector $\boldsymbol{v}_p$ per person p. The entries of $\boldsymbol{w}_i$ can be interpreted as describing item i in some abstract way (as a point in $\mathbb{R}^q$), and we can choose to look at each dimension of $\boldsymbol{w}_i$ as describing a unique feature of item i. The same features are used to describe all items (as the representation – a vector in $\mathbb{R}^q$ – is fixed for all items), but the presence of each feature can differ between the items (as numerical values of the vectors $\boldsymbol{w}_i$ may differ). In the movie-domain, one may for instance find that the first dimension of $\boldsymbol{w}_i$ is used to describe the amount of explicit violence in a movie, the second measuring the scale of the production, the third describing the age of the typical viewer (i.e., kids, teenager, youth, or adult audience), and so on. Similarly, each user is represented by a vector in q-dimensional space describing his or her liking for each of the features used to describe the items (so, in the example above, the first entry may say something about tolerance for explicit violence, the second say something about preference for smaller vs. larger productions, and so on).

To learn this representation, we need to find the pair $(\boldsymbol{V}, \boldsymbol{W})$ that minimizes the observed error over the ratings. It is common to consider the squared error, i.e., the Frobenius norm denoted by $\|\cdot\|_F$. Thus, the learning task can be stated as the following minimization problem:

$$\{\boldsymbol{V}, \boldsymbol{W}\} = \arg \min_{\{\tilde{\boldsymbol{V}}, \tilde{\boldsymbol{W}}\}} \left\| \boldsymbol{R} - \tilde{\boldsymbol{V}}^{\mathrm{T}}\tilde{\boldsymbol{W}} \right\|_F . \tag{1}$$

We know how to solve Eq. (1) when $\boldsymbol{R}$ contains no missing values; in this case $\boldsymbol{V}$ and $\boldsymbol{W}$ find their interpretation via the singular value decomposition (SVD) representation of $\boldsymbol{R}$. However, the rating matrix is sparsely filled, so we need to find an analogue to SVD, which is well-defined also when $\boldsymbol{R}$ contains missing values [50,44]. This is an idea eagerly explored in the CF community [52], where one of the leading approaches is to numerically minimize the objective function

$$\begin{aligned} \|\boldsymbol{r} - \boldsymbol{V}^{\mathrm{T}}\boldsymbol{W}\|_F &= \sum_{p=1}^{\#U} \sum_{i=1}^{\#M} \delta(p, i) \left(\boldsymbol{r}(p, i) - g(\boldsymbol{\theta}_{\boldsymbol{r}}, p, i)\right)^2 \\ &= \sum_{p=1}^{\#U} \sum_{i \in \mathcal{I}(p)} \left(\boldsymbol{r}(p, i) - \boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{w}_i\right)^2 . \end{aligned} \tag{2}$$

This can, e.g. be done using gradient descent learning, which leads to the updating rules

$$\boldsymbol{v}_p \leftarrow \boldsymbol{v}_p + \eta \sum_{i \in \mathcal{I}(p)} \left(\boldsymbol{r}(p, i) - \boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{w}_i\right) \boldsymbol{w}_i, \qquad \boldsymbol{w}_i \leftarrow \boldsymbol{w}_i + \eta \sum_{p \in \mathcal{P}(i)} \left(\boldsymbol{r}(p, i) - \boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{w}_i\right) \boldsymbol{v}_p,$$

where η is the learning rate.

One apparent problem with Eq. (2) is that the model is not regularized, meaning that the parameters $\boldsymbol{V}$ and $\boldsymbol{W}$ can grow without bounds (with over-fitting as the probable result). This is particularly problematic when a user p has rated only a few items (leading to an unstable estimate for $\boldsymbol{v}_p$) or an item i has been rated by only a few users (in this case leading to an unstable estimate of $\boldsymbol{w}_i$). The typical way of handling this is by adding a term that penalizes large parameters, e.g. by looking at the objective function [44]

$$\sum_{p=1}^{\#U} \sum_{i \in \mathcal{I}(p)} \left(\boldsymbol{r}(p, i) - \boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{w}_i\right)^2 + \lambda \sum_{p=1}^{\#U} \sum_{i \in \mathcal{I}(p)} \left(\boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{v}_p + \boldsymbol{w}_i^{T}\boldsymbol{w}_i\right), \tag{3}$$

where λ is a parameter that balances parameter regularization and model fit.

4.2. A simple generative model

A shortcoming with the present model is that it is not probabilistic, hence we cannot calculate the uncertainty associated with the different predictions (this is a feature we will find useful when performing multi-rating predictions). To avoid this problem, one solution is to embed the optimization problem in a statistical model. Since we are aiming at reducing the Frobenius norm, we can equivalently regard the ratings as coming from a Gaussian model with known variance σ^2 (see, e.g. [10]),

$$\boldsymbol{R}(p,i)\,|\{\boldsymbol{v}_p,\boldsymbol{w}_i,\sigma^2\}\sim\mathcal{N}\left(\boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{w}_i,\sigma^2\right), \tag{4}$$

and chose $\boldsymbol{v}_p$ and $\boldsymbol{w}_i$ to maximize the likelihood of the observed entries $\boldsymbol{r}$.

Next, we convert the probabilistic model of Eq. (4) into a *latent variable* model by considering $\{\boldsymbol{v}_p\}_{p=1}^{\#U}$ as being *i.i.d.* realizations of a random variable $\boldsymbol{U}$ rather than parameters in the model. With this perspective Eq. (4) corresponds to assuming that $\boldsymbol{R}(p,i)\,|\{\boldsymbol{U}=\boldsymbol{u}_p\}\sim\mathcal{N}\left(\boldsymbol{u}_p^{\mathrm{T}}\boldsymbol{w}_i,\sigma^2\right)$. For mathematical convenience we will assume that $\boldsymbol{U}\sim\mathcal{N}(\boldsymbol{\mu}_U,\boldsymbol{I})$ a priori, where $\boldsymbol{\mu}_U$ is the q-dimensional vector of expected values for $\boldsymbol{U}$ and $\boldsymbol{I}$ is the $q\times q$ identity matrix. The parameters $\boldsymbol{w}_i$ are shared among users, so this model is related to the traditional *factor analysis model*, see, e.g., [26]. The model is illustrated as a Bayesian network in Fig. 1.

The latent variable model gives us modeling control over $\boldsymbol{U}$, as it is assumed to follow a Gaussian distribution with rather small variation a priori. By utilizing that the distribution of $\boldsymbol{R}(p,\cdot)$ can be written as

$$f(\boldsymbol{r}(p,\cdot))=\int_{\boldsymbol{u}} f(\boldsymbol{r}(p,\cdot)\,|\boldsymbol{U}=\boldsymbol{u})\cdot f(\boldsymbol{u})\,d\boldsymbol{u},$$

it follows that the model is valid under the assumption that rating vectors are *i.i.d.* realizations from the distribution

$$[\boldsymbol{R}(p,1)\ \boldsymbol{R}(p,2)\ldots\boldsymbol{R}(p,\#M)]^{\mathrm{T}}\sim\mathcal{N}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\mu}_U,\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}+\sigma^2\boldsymbol{I}\right); \tag{5}$$

recall that $\boldsymbol{W}=[\boldsymbol{w}_1,\ldots,\boldsymbol{w}_{\#M}]$ is the matrix containing all "movie-representations" $\boldsymbol{w}_i$. Maximum likelihood parameters for the model can be learned using the EM algorithm [11,26].

This model is focused on a single user p, and uses the ratings of a single user to predict the ratings of the items currently not rated by the user.

Alternatively, we can focus on the items instead, giving us the *item-based* perspective, where a model is developed for all the ratings given to a particular item. Again, we take Eq. (4) as our starting-point, but this time we assume that $\{\boldsymbol{w}_i\}_{i=1}^{\#M}$ are *i.i.d.* realization of a random variable that we will denote $\boldsymbol{M}$. By assuming that $\boldsymbol{M}\sim\mathcal{N}(\boldsymbol{\mu}_M,\boldsymbol{I})$ apriori, we get the model

$$\boldsymbol{R}(\cdot,i)\sim\mathcal{N}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{\mu}_M,\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}+\sigma^2\boldsymbol{I}\right),$$

which can be used for making joint predictions of how several users will rate an item i.

A potential problem with the above models is that during inference the model will either focus on the ratings of the active user (user-based model) or the active item (item-based model). Although these models can, in principle, be used for multi-rating predictions (e.g. the item-based model can be used to find an item several users like), the quality of the predictions is usually poor, since correlations (especially negative) in the users' ratings are not taken into account (see also Section 6.3). To alleviate this, we propose a combined model where the user-view and the item-view are merged.

4.3. The proposed generative model

4.3.1. A dual perspective

As for the previous models, we will use latent variables to describe users and items abstractly as real vectors. We will, however, extend the model by considering all users and all items *simultaneously*. Let $\boldsymbol{M}_i$ be the latent variables representing item i, and assume a priori that $\boldsymbol{M}_i\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})$, for $1\leq i\leq\#M$. Similarly, for users we assume the existence of the latent variables $\boldsymbol{U}_p$ representing user p, and choose $\boldsymbol{U}_p\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})$, for $1\leq p\leq\#U$. The final model is now built by assuming that there exists a linear mapping from the space describing users and items to the numerical rating scale:

$$\boldsymbol{R}(p,i)\,|\{\boldsymbol{M}_i=\boldsymbol{m}_i,\boldsymbol{U}_p=\boldsymbol{u}_p\}=\boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{m}_i+\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{u}_p+\phi_p+\psi_i+\epsilon. \tag{6}$$

In Eq. (6), $\boldsymbol{m}_i$ and $\boldsymbol{u}_p$ are abstract representations of item i and user p (possibly of different dimensionality). For example, one may interpret the different dimensions of $\boldsymbol{m}_i$ as representing different features of movie i (discussed further in Section 4.4) and the dimensions of $\boldsymbol{u}_p$ as corresponding to different user characteristics. Since the variables are continuous, the value $\boldsymbol{u}_p^j$ of the jth variable $\boldsymbol{U}_p^j$ can be interpreted as representing to what extent user p has the characteristics modeled by variable j. This also means that rather than assigning users to single "user classes", the continuous variables $\boldsymbol{U}_p^j$ encode *to what extent* a user belongs to a certain class. The final rating in Eq. (6) is now determined as an additive combination of user p's preferences $\boldsymbol{v}_p$ for (or attitude towards) the features describing item i and item i's disposition $\boldsymbol{w}_i$ towards the different user classes.[3] The constants ϕ_p and ψ_i in Eq. (6) can be interpreted as representing the average rating of user p and the average rating of item i (after compensating for the user average), respectively. Furthermore, ϵ represents "sensor noise", i.e., the variation in the ratings the model cannot explain, and we will assume that $\epsilon\sim\mathcal{N}(0,\theta)$. By examining the model more closely, the

[3] Note that the relative importance of the movie features and the user class can be encoded in the weight vectors $\boldsymbol{v}_p$ and $\boldsymbol{w}_i$.

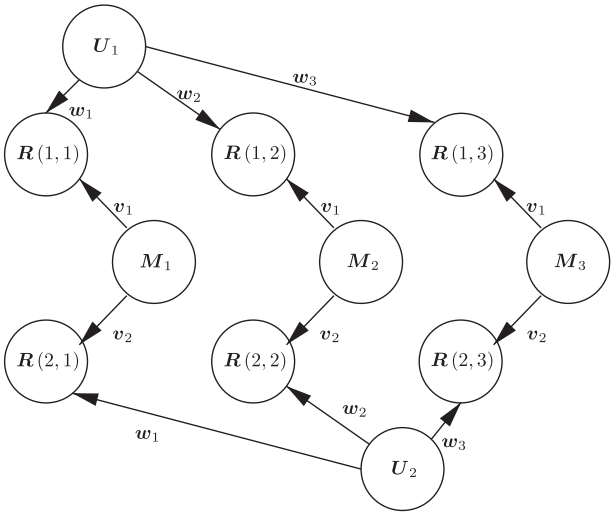


Fig. 2. The full statistical model for collaborative filtering; this model has $\#M = 3$ and $\#U = 2$.

marginal distribution for $\boldsymbol{R}(p, i)$ can be written as

$$\boldsymbol{R}(p, i) \sim \mathcal{N}\left(\phi_p + \psi_i, \boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{v}_p + \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_i + \theta\right).$$

Finally it should be emphasized that we have the same number of latent variables for all users (i.e., $|\boldsymbol{U}_o| = |\boldsymbol{U}_p|$) and for all movies (i.e., $|\boldsymbol{M}_r| = |\boldsymbol{M}_i|$).

The main motivation for using the model is how correlations between *arbitrary* ratings are efficiently taken into account when making recommendations. Consider Fig. 2, which shows a full BN representation of the proposed model for a domain with two users and three items ($\#U = 2$ and $\#M = 3$ in this example). For the sake of the argument, let us assume that both users have rated Item 1, and that User 1 has rated Item 2 also. Consider now how this last rating, $\boldsymbol{r}(1, 2)$, influences the predictions the system will make:

User-based perspective: Entering the evidence $\boldsymbol{r}(1, 2)$ will tell the model something about User 1 (represented by $\boldsymbol{U}_1$). This new information is incorporated in the updated posterior distribution over $\boldsymbol{U}_1$, which will influence the prediction for all ratings User 1 have not yet made (in this case only $\boldsymbol{R}(1, 3)$ is affected).

Item-based perspective: The evidence $\boldsymbol{r}(1, 2)$ also tells the model something about the active item, resulting in an updated posterior for $\boldsymbol{M}_2$. This influences the distribution over all remaining ratings for Item 2 ($\boldsymbol{R}(2, 2)$ in this case).

Global perspective: The model also offers a global view towards the recommendation task. To see this, let us follow a slightly more intricate chain of reasoning: When evidence about $\boldsymbol{r}(1, 2)$ is entered, one immediate effect is that the posterior distribution over $\boldsymbol{U}_1$ is updated to take the new information into account. Changing $\boldsymbol{U}_1$ gives the model a new perspective towards all ratings User 1 has given, and in particular the observation $\boldsymbol{r}(1, 1)$ can be re-considered: If $\boldsymbol{U}_1$ is changed we get a new understanding of how that particular rating came to be, and this may shed new light on Item 1. Thus, the system-internal encoding of Item 1, represented by the distribution over $\boldsymbol{M}_1$, should be altered. Next, the new posterior over $\boldsymbol{M}_1$ makes the model reconsider its representation of all users who have already rated Item 1, and thus the internal representation of $\boldsymbol{U}_2$ must also be updated. This will again change the model's belief in all ratings that User 2 will give, in particular the expectation regarding Item 3, i.e., the rating $\boldsymbol{R}(2, 3)$ is also affected. Thus, $\boldsymbol{R}(2, 3)$ and $\boldsymbol{R}(1, 2)$ are *dependent* given the evidence, written $\boldsymbol{R}(2, 3) \not\perp \boldsymbol{R}(1, 2) \,|\{\boldsymbol{R}(1, 1), \boldsymbol{R}(2, 1)\}$. This exemplifies the global view of the present model.

To summarize, contrary to standard (non-relational) probabilistic models, we treat the entire database as a single case. This also implies that we no longer have to explicitly assume that the different ratings are independent and identical distributed (the underlying distribution still has to respect the independence assumptions in the model, though). Comparing the proposed model to the SVD-based techniques described in Section 4.1, the model in Eq. (6) is probabilistic, and therefore gives uncertainty estimates in its ratings. In contrast to the models presented in Section 4.2, Eq. (6) maintains the user perspective and the item perspective simultaneously, something we will later show improves the predictive ability (see Section 6). Finally, one could envision building a model from Eq. (4) by simply replacing both $\boldsymbol{v}_p$ and $\boldsymbol{w}_i$ by random variables. In this case, the proposed model distinguishes itself by relying on an additive combination function, which ensures that during inference we will always stay within the class of linear Gaussian models for which there are known closed-form updating rules, and not be forced to consider product distributions.

4.3.2. Generating multi-ratings

The proposed model generates a statistical distribution over all ratings simultaneously, and we can utilize this to generate multi-ratings (i.e., combined ratings over several items and/or users); see [23] for an overview. To exemplify, let us consider the problem of finding an item that persons p_1 and p_2 will enjoy together, that is, we will use the joint distribution over $[\boldsymbol{R}(p_1,i)\ \ \boldsymbol{R}(p_2,i)]^{\mathrm{T}}$ to evaluate item i. After establishing this joint distribution (see below), we define a utility function $V(\boldsymbol{r}(p_1,i),\boldsymbol{r}(p_2,i))$ encoding how different combinations of ratings are evaluated. We then choose the item that maximizes the expected utility wrt. the joint distribution over the ratings.

Different strategies for selecting an "appropriate" item for users p_1 and p_2 can be envisioned, each leading to a different formulation of the utility function [8,36]:

Independence: Choose the value function $V(\boldsymbol{r}(p_1,i),\boldsymbol{r}(p_2,i)) = \boldsymbol{r}(p_1,i) + \boldsymbol{r}(p_2,i)$ to produce a preference for an item that is enjoyed the best *on average*.

Maximin: Use the value function $V(\boldsymbol{r}(p_1,i),\boldsymbol{r}(p_2,i)) = \min(\boldsymbol{r}(p_1,i),\boldsymbol{r}(p_2,i))$ to introduce preference for items that both users will find acceptable. A recommendation based on the maximin principle will typically be more "safe" than one based on independence, as high predictive variance will be regarded as a disadvantage.

General formulations: Finally, value-functions can be hand-crafted to produce particular results, for example preferring items that both users dislike over an item that splits opinions.

We end this discussion by detailing how the required joint distribution function can be found. Firstly, we use the conditional independence statements embedded in the model representation to realize that all ratings are conditionally independent (written using the "$\perp\!\!\!\perp$" symbol) given the latent variables:

$$\{\boldsymbol{R}(p_1,i),\boldsymbol{R}(p_2,i)\} \perp\!\!\!\perp \boldsymbol{r} \mid \{\boldsymbol{M}_i,\boldsymbol{U}_{p_1},\boldsymbol{U}_{p_2}\}.$$

Thus, to calculate the posterior distribution over $[\boldsymbol{R}(p_1,i)\ \ \boldsymbol{R}(p_2,i)]^{\mathrm{T}}$ given $\boldsymbol{r}$, we should first calculate the effect $\boldsymbol{r}$ has on the latent variables, then project this information into updated beliefs about the queried ratings. From the basic properties of the multivariate Gaussian distribution (see any standard textbook on statistics or machine learning, e.g. [3]), we obtain that the joint distribution over the latent variables conditioned on the observed ratings is given by

$$\left[\boldsymbol{M}^{\mathrm{T}}\ \boldsymbol{U}^{\mathrm{T}}\right]^{\mathrm{T}} \Big| \boldsymbol{r} \sim \mathcal{N}(\boldsymbol{\nu},\boldsymbol{\Sigma}),$$

where $\boldsymbol{M} = (\boldsymbol{M}_1^{\mathrm{T}},\ldots,\boldsymbol{M}_{\#M}^{\mathrm{T}})^{\mathrm{T}}$ and $\boldsymbol{U} = (\boldsymbol{U}_1^{\mathrm{T}},\ldots,\boldsymbol{U}_{\#U}^{\mathrm{T}})^{\mathrm{T}}$ are the latent variables for the items and users, respectively. Here, the covariance matrix is given by (see also Appendix Appendix A)

$$\boldsymbol{\Sigma} = (\boldsymbol{I} + \boldsymbol{L}^{\mathrm{T}}\theta^{-1}\boldsymbol{L})^{-1},$$

where $\boldsymbol{L}$ is the sparse regression matrix (of size $|\boldsymbol{r}| \times (|\boldsymbol{M}| + |\boldsymbol{U}|)$) for the ratings given $\boldsymbol{M}$ and $\boldsymbol{U}$ (i.e., consisting of the $\boldsymbol{v}_p$s and $\boldsymbol{w}_i$s), and

$$\boldsymbol{\nu} = \boldsymbol{\Sigma}(\boldsymbol{L}^{\mathrm{T}}\theta^{-1}(\boldsymbol{r} - (\phi + \psi))).$$

Next, we define the matrix $\boldsymbol{A} = [\boldsymbol{a}_1\ \boldsymbol{a}_2]$, where the column-vector $\boldsymbol{a}_j$ is such that it contains zero-elements except for two parts containing $\boldsymbol{w}_i$ and $\boldsymbol{v}_{p_j}$, and designed s.t. $\boldsymbol{a}_j^{\mathrm{T}}\begin{bmatrix}\boldsymbol{m}\\ \boldsymbol{u}\end{bmatrix} = \boldsymbol{v}_{p_j}^{\mathrm{T}}\boldsymbol{m}_i + \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{u}_{p_j}$. Thus,

$$\begin{bmatrix}\boldsymbol{R}(p_1,i)\\ \boldsymbol{R}(p_2,i)\end{bmatrix} \Bigg| \begin{bmatrix}\boldsymbol{M}\\ \boldsymbol{U}\end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{A}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{m}\\ \boldsymbol{u}\end{bmatrix} + \begin{bmatrix}\phi_{p_1}+\psi_i\\ \phi_{p_2}+\psi_i\end{bmatrix}, \theta\boldsymbol{I}\right)$$

and it follows that the joint distribution over the queried ratings are

$$\begin{bmatrix}\boldsymbol{R}(p_1,i)\\ \boldsymbol{R}(p_2,i)\end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\nu} + \begin{bmatrix}\phi_{p_1}+\psi_i\\ \phi_{p_2}+\psi_i\end{bmatrix}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{A} + \theta\boldsymbol{I}\right).$$

4.4. Model interpretation

To get additional insight into the model, it may be informative to analyze a model learned for a particular dataset. To this end, we learned a model (detailed in Section 5) for the MovieLens dataset [19] with two latent variables for each movie and one latent variable for each user, i.e., ($|\boldsymbol{M}_i| = 2$ and $|\boldsymbol{U}_p| = 1$).

If we start off by considering the latent variables for the movies, then these variables can be interpreted as abstract representations of the movies. That is, for movie i we have a Gaussian distribution over $\mathbb{R}^q$ (assuming $|\boldsymbol{M}_i| = q$), and $\hat{\boldsymbol{m}}_i =$

Table 1
The 10 movies closest to *Star Wars* and *Three Colors: Blue*, respectively.

1.	Star Trek IV	1.	The Apostle
2.	Indiana Jones and the Last Crusade	2.	Three Colors: White
3.	The Empire Strikes Back	3.	Heavenly Creatures
4.	Independence Day	4.	Stealing Beauty
5.	Home Alone	5.	Three Colors: Red
6.	Back to the Future	6.	Hoodlum
7.	Jaws 2	7.	In the company of men
8.	Star Trek VI	8.	Big night
9.	Return of the Jedi	9.	Wings of Desire
10.	Twister	10.	Boogie Nights

Table 2
The 10 movies furthest away from *Star Wars* and *Three Colors: Blue*, respectively.

1.	Angels and Insects	1.	Die Hard
2.	Three Colors: Blue	2.	Raiders of the Lost Ark
3.	The Unbearable Lightness of Being	3.	Jurassic Park
4.	Stealing Beauty	4.	Ace Ventura: Pet Detective
5.	The Apostle	5.	Home Alone
6.	The Postman	6.	The Empire Strikes Back
7.	Breakfast at Tiffany's	7.	The Terminator
8.	Il Postino	8.	Field of Dreams
9.	Breaking the Waves	9.	Terminator 2: Judgment Day
10.	Big night	10.	Star Trek II

$\mathbb{E}(\boldsymbol{M}_i|\boldsymbol{r})$ can therefore be considered a point estimate representation of movie i. With this interpretation we hypothesize that if the point estimates of two movies are close in latent space, then they have the same abstract representation, and they should therefore be similar (i.e., have similar rating patterns). To test this hypothesis we determined the movies that are close to *Star Wars* (1977) and *Three Colors: Blue* (1993).[4] As distance measure for two movies $\hat{\boldsymbol{m}}_i$ and $\hat{\boldsymbol{m}}_j$ we used the Mahalanobis distance to account for the correlation between the latent variables:

$$\text{dist}_M\left(\hat{\boldsymbol{m}}_i, \hat{\boldsymbol{m}}_j\right) = \left(\hat{\boldsymbol{m}}_i - \hat{\boldsymbol{m}}_j\right)^{\mathrm{T}} \hat{\boldsymbol{Q}} \left(\hat{\boldsymbol{m}}_i - \hat{\boldsymbol{m}}_j\right),$$

where $\hat{\boldsymbol{Q}}$ is the empirical precision matrix (the inverse of the empirical covariance matrix) for the latent variables calculated from the point estimates of the movies in the dataset.

Star Wars is a sci.-fi./action movie with sequels *The Empire Strikes Back* and *Return of the Jedi*, so we would hope to see these movies, as well as other sci.-fi. movies, to be named "close" to *Star Wars*. On the other hand, *Three Colors: Blue* is a drama, and is the first in a trilogy of movies that also includes *Three Colors: Red* and *Three Colors: White*. The results are shown in Table 1. Out of the 10 movies closest to *Star Wars*, 6 are movies that we (the authors) believe are well classified as "similar to Star Wars". *Indiana Jones and the Last Crusade* is somewhat related in the sense that it is an adventure movie, but, e.g. *Home Alone* does not seem to fit in that well. We see a similar pattern for the movies closest to *Three Colors: Blue*. Observe that for both trilogies, the two other movies in the trilogies appear on the lists. Considering that there are 1682 movies in the database we find this quite satisfactory.

With the specified distance measure we are also able to find the movies furthest away from *Star Wars* and *Three Colors: Blue*. The results are shown in Table 2, where we find that the movies furthest away from *Star Wars* are primarily dramas and the movies furthest away *Three Colors: Blue* are mainly sci.-fi. movies and comedies.

One may also attempt to investigate whether the latent variables have a semantic interpretation. For this analysis we selected the movies with smallest and highest values along each of the two dimensions in the latent space. The results can be seen in Tables 3 and 4. Based on the listed movies, one possible semantic interpretation might be that the first dimension encodes to what extent the movie would appeal to a male/female audience and the second dimension represent whether the movie appeals to a teenage audience.

Next, we consider the parameter ψ_i. Recall that this parameter is intended to represent the average rating of item i (after adjusting for the user types that have rated the movie), and ψ_i may therefore be thought of as representing the *quality* of an item. For illustration, we ordered the movies based on the estimated ψ-values. The result is shown in Table 5, where each movie's position on the Internet Movie Database's (IMDB's) list of top 250 movies are given as reference.[5] Note that our model has picked out 3 "Wallace and Gromit" movies (marked with a * in the table). These movies are either short-movies ("A close shave" and "The Wrong Trousers") or a compilation of such ("The Best of Aardman Animation"), and do therefore not qualify for the IMDB top 250-list. However, the movies' IMDB ratings make all three of them comparable to IMDB movies around Top 50–80: The Wrong Trousers is rated 8.5 (place 37–55), A Close Shave is rated 8.3 (place 76-112), and The best of Aardman Animation is rated 8.4 (place 56-75). Note also that our dataset only contains movies released in 1998 or before,

[4] In the analyzes below, we only considered movies with at least 50 ratings.
[5] http://www.imdb.com, retrieved August 5th, 2011.

Table 3
The 10 movies with lowest and highest values in the first dimension in the latent space. Semantically, this dimension may be interpreted as to what extent the movie appeals to a male/female audience.

1.	Three Colors: Blue	1.	Die Hard
2.	Apostle, The	2.	Raiders of the Lost Ark
3.	Stealing Beauty	3.	Jurassic Park
4.	The Unbearable Lightness of Being	4.	Home Alone
5.	Angels and Insects	5.	Empire Strikes Back, The
6.	Three Colors: White	6.	Star Trek: The Wrath of Khan
7.	Boogie Nights	7.	Star Wars
8.	Heavenly Creatures	8.	Return of the Jedi
9.	Big Night	9.	Ace Ventura: Pet Detective
10.	Cold Comfort Farm	10,	Field of Dreams

Table 4
The 10 movies with lowest and highest values in the second dimension in the latent space. Semantically, this dimension may be interpreted as to what extent the movie appeals to a teenage audience.

1.	Beavis and Butt-head Do America	1.	Breakfast at Tiffany's
2.	Event Horizon	2.	Bridges of Madison County, The
3.	Army of Darkness	3.	On Golden Pond
4.	Spawn	4.	Angels and Insects
5.	Starship Troopers	5.	English Patient, The
6.	From Dusk Till Dawn	6.	Room with a View, A
7.	Crow, The	7.	It's a Wonderful Life
8.	Evil Dead II	8.	Crying Game, The
9.	Supercop	9.	Old Yeller
10.	Fifth Element, The	10.	My Fair Lady

Table 5
The 10 "best" movies, i.e., the movies with the highest ψ_i value.

1.	The Shawshank Redemption	IMDB: 1
2.	Schindler's List	IMDB: 7
3.	A Close Shave*	IMDB: NA
4.	The Wrong Trousers*	IMDB: NA
5.	Casablanca	IMDB: 19
6.	Wallace and Gromit: The Best of Aardman Animation*	IMDB: NA
7.	Star Wars	IMDB: 16
8.	The Usual Suspects	IMDB: 24
9.	Rear Window	IMDB: 21
10.	Raiders of the Lost Ark	IMDB: 23

which explains why, e.g. "The Dark Knight" (IMDB 10) and the "The Lord of the Rings" series (IMDB 11, 17, and 29) are not on our list.[6]

The IMDB Top 250 list is obviously not an objective truth, but we compare our results to it because the IMDB has a much higher number of ratings than the MovieLens dataset, and may therefore offer a more robust ranking. For comparison, we found that simply ordering the movies by their average rating did not give convincing results; none of the 10 movies that are top-ranked following this scheme are in the IMDB Top 250. We believe the reason for this is twofold: (*i*) the sparsity of the data; items with few ratings may get "extreme" averages, (*ii*) simply talking averages disregards the underlying differences between users: Some are "happy" and others are "grumpy". The fact that a "happy" user has seen movie i_1 and a "grumpy" one has seen i_2 does not mean that movie i_1 is better than i_2 (even though it may get a better rating).

5. Learning

5.1. The EM algorithm

When learning the model, we need to find the number of latent variables to describe both users and items (the model structure) as well as learning the parameters for the chosen model structure. The model structure is learned based on a greedy search (detailed in Section 6) and the parameters in the model are learned using the EM algorithm [11]. However, contrary to standard (non-relational) applications of the EM algorithm, we treat the entire database as a single case.

Learning the parameters of the model amounts to estimating the parameters for the regression model

$$\boldsymbol{R}(p,i)\,|\{\boldsymbol{m}_i,\boldsymbol{u}_p\} \sim \mathcal{N}(\boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{m}_i + \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{u}_p + \phi_p + \psi_i, \theta),$$

since we assume a standard Gaussian distribution associated with the latent variables.

[6] 21 of the 75 highest ranked movies in IMDB 250 appeared after 1998.

When applying the EM algorithm in this setting, we get the following updating rules for the parameters (see Appendix Appendix A for the derivations):

$$
\begin{aligned}
\hat{\theta} &\leftarrow \frac{1}{d}\sum_{p=1}^{\#U}\sum_{i\in\mathcal{I}(p)}\mathbb{E}[(\boldsymbol{r}(p,i)-(\boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{M}_i+\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{U}_p+\phi_p+\psi_i))^2];\\
\hat{\boldsymbol{v}_p} &\leftarrow \left[\sum_{i\in\mathcal{I}(p)}\mathbb{E}(\boldsymbol{M}_i\boldsymbol{M}_i^{\mathrm{T}})\right]^{-1}\left[\sum_{i\in\mathcal{I}(p)}(\mathbb{E}(\boldsymbol{M}_i)\boldsymbol{r}(p,i)-\mathbb{E}(\boldsymbol{M}_i\boldsymbol{U}_p^{\mathrm{T}})\boldsymbol{w}_i-\mathbb{E}(\boldsymbol{M}_i)(\phi_p+\psi_i))\right];\\
\hat{\phi_p} &\leftarrow \frac{1}{|\mathcal{I}(p)|}\sum_{i\in\mathcal{I}(p)}\left(\boldsymbol{r}(p,i)-(\boldsymbol{v}_p^{\mathrm{T}}\mathbb{E}(\boldsymbol{M}_i)+\boldsymbol{w}_i^{\mathrm{T}}\mathbb{E}(\boldsymbol{U}_p)+\psi_i)\right);\\
\hat{\boldsymbol{w}_i} &\leftarrow \left[\sum_{p\in\mathcal{P}(i)}\mathbb{E}(\boldsymbol{U}_p\boldsymbol{U}_p^{\mathrm{T}})\right]^{-1}\left[\sum_{p\in\mathcal{P}(i)}\mathbb{E}(\boldsymbol{U}_p)\boldsymbol{r}(p,i)-\mathbb{E}(\boldsymbol{U}_p\boldsymbol{M}_i^{\mathrm{T}})\boldsymbol{v}_p-\mathbb{E}(\boldsymbol{U}_p)(\phi_p+\psi_i)\right];\\
\hat{\psi}_i &\leftarrow \frac{1}{|\mathcal{P}(i)|}\sum_{p\in\mathcal{P}(i)}(\boldsymbol{r}(p,i)-(\boldsymbol{v}_p^{\mathrm{T}}\mathbb{E}(\boldsymbol{M}_i)+\boldsymbol{w}_i^{\mathrm{T}}\mathbb{E}(\boldsymbol{U}_p)+\phi_p)).
\end{aligned}
\tag{7}
$$

Since the number of latent variables used to described both users and items (i.e., $|\boldsymbol{U}_p|$ and $|\boldsymbol{M}_i|$) is typically small (in our experiments we have considered $|\boldsymbol{M}_i|, |\boldsymbol{U}_p| \leq 5$), it is clear from the above expressions that the complexity of performing the M-step is relatively low. Unfortunately, the calculations of the expectations used in the M-step requires the calculation of the full covariance matrix for all the latent variables; in the calculation of, e.g., $\mathbb{E}(\boldsymbol{M}_i\boldsymbol{U}_p^{\mathrm{T}})$ we exploit that $\mathrm{Cov}(\boldsymbol{M}_i\boldsymbol{U}_p^{\mathrm{T}})$ can be extracted directly from the posterior covariance matrix for all the latent variables. Note that although the corresponding precision matrix might be sparse, this is not the case for the covariance matrix (which is also evident when one analyzes the independence properties in the model).[7] The derivations of the expectations are detailed in Appendix A.

Finally, when learning the collaborative filtering model we also need to select the number of latent variables representing the users and movies, respectively. Recall that all users are described using the same number of latent variables; the same holds for the movies. In the experiments we have run, these parameters were found using a greedy approach that will be described in Section 6; alternatively one could also consider the wrapper approach [28].

5.2. Regularization

In our preliminary experiments we frequently observed that some regression vectors (primarily for users and items with few ratings) contained unexpectedly large values, suggesting that the model might be over-fitted for these parts of the data. When analyzing the updating rule for, e.g. $\boldsymbol{v}_p$ (see Eq. (7)) we find a possible explanation for this behavior: the updating rule for $\boldsymbol{v}_p$ requires the inversion of $\boldsymbol{A} = \sum_{i\in\mathcal{I}(p)}\mathbb{E}(\boldsymbol{M}_i\boldsymbol{M}_i^{\mathrm{T}})$, which is a sum of $|\mathcal{I}(p)|$ rank-one matrices. $\boldsymbol{A}$ is thus at most rank-$|\mathcal{I}(p)|$, but as the elements in the sum may be close to being linearly dependent (movies rated by the same user may be similar [35]), the actual rank of $\boldsymbol{A}$ may be less than $|\boldsymbol{M}_i|$, and the results for $\boldsymbol{v}$ and $\boldsymbol{w}$ will therefore be numerically unstable. In our preliminary experiments with $|\boldsymbol{M}_i| = |\boldsymbol{U}_p| = 2$ we, e.g., found that the regression vectors contain components having values larger than 20 when learned from the MovieLens database. This database has ratings ranging from one to five, and intuitively, one would not expect to see a large part of the estimated parameters to have absolute values greater than the spread of the ratings. One approach to this problem is to consider the estimation of, e.g., $\boldsymbol{v}_p$ as a linear regression problem

$$\boldsymbol{R}(p,i) = \boldsymbol{M}_i^{\mathrm{T}}\boldsymbol{v}_p + \boldsymbol{U}_p^{\mathrm{T}}\boldsymbol{w}_i + \phi_p + \psi_i + \epsilon,$$

where $\epsilon \sim \mathcal{N}(0,\theta)$. Since $\boldsymbol{M}_i$ and $\boldsymbol{U}_p$ are unobserved we attempt to minimize the expected least squares solution, and it is now easy to see that Eq. (7) is also the solution that minimizes the expected least squared error.[8] A standard approach for handling the situation where $\boldsymbol{A} = \sum_{i\in\mathcal{I}(p)}\mathbb{E}(\boldsymbol{M}_i\boldsymbol{M}_i^{\mathrm{T}})$ is close to being singular (or with correlated variables), is to employ regularization. A possibility is Tikhonov regularization (also known as ridge regression), giving the modified updating rule [17]:

$$\hat{\boldsymbol{v}_p} \leftarrow \left[\sum_{i\in\mathcal{I}(p)}\mathbb{E}(\boldsymbol{M}_i\boldsymbol{M}_i^{\mathrm{T}}) + \alpha\boldsymbol{I}\right]^{-1}\left[\sum_{i\in\mathcal{I}(p)}(\mathbb{E}(\boldsymbol{M}_i)\boldsymbol{r}(p,i)-\mathbb{E}(\boldsymbol{M}_i\boldsymbol{U}_p^{\mathrm{T}})\boldsymbol{w}_i-\mathbb{E}(\boldsymbol{M}_i)(\phi_p+\psi_i))\right],$$

[7] In our experiments, we have observed that the covariance matrix typically contains a large number of small entries, which may be exploited in an approximate inference scheme. This is a topic for future research and outside the scope of the present paper.

[8] For the standard matrix formulation of the solution, note that, e.g. $\sum_{i\in\mathcal{I}(p)}\mathbb{E}(\boldsymbol{M}_i\boldsymbol{M}_i^{\mathrm{T}}) = \mathbb{E}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})$, where $\boldsymbol{X}_{i,:} = \boldsymbol{M}_i^{\mathrm{T}}$.

where $\alpha = 0$ gives the standard least square solution. This regularized updating rule can be derived by assigning a suitable prior distribution to the regression parameters. Specifically, by letting $\boldsymbol{v}_p \sim \mathcal{N}(\boldsymbol{0}, \tau \boldsymbol{I})$, then the estimate above maximizes the expected (w.r.t. $\boldsymbol{M}$ and $\boldsymbol{U}$) log-posterior density for $\boldsymbol{v}_p$ given $\boldsymbol{r}$, with $\alpha = \theta/\tau$. A similar result is obtained for $\boldsymbol{w}_i$:

$$\hat{\boldsymbol{w}}_i \leftarrow \left[\sum_{p \in \mathcal{P}(i)} \mathbb{E}(\boldsymbol{U}_p \boldsymbol{U}_p^{\mathrm{T}}) + \alpha \boldsymbol{I} \right]^{-1} \left[\sum_{p \in \mathcal{P}(i)} \mathbb{E}(\boldsymbol{U}_p) \boldsymbol{r}(p, i) - \mathbb{E}(\boldsymbol{U}_p \boldsymbol{M}_i^{\mathrm{T}}) \boldsymbol{v}_p - \mathbb{E}(\boldsymbol{U}_p)(\phi_p + \psi_i) \right].$$

Following general practice [1] we use the estimators for ϕ_p and ψ_i that were found *without* regularization.

6. Results

6.1. Introduction to the datasets

In this section we investigate the predictive performance of the proposed system. Specifically, we evaluate the system using two different datasets: MovieLens [19] and Jester [14].

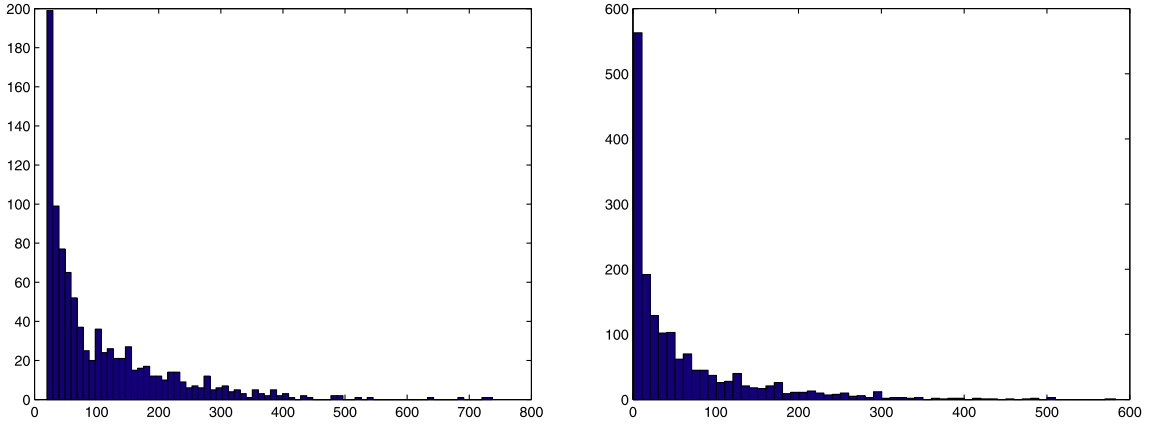
The MovieLens dataset consists of 100,000 integer ratings (values from 1 to 5), collected from 943 users on 1682 movies. The mean rating is 3.53, and the standard deviation is 1.13. These numbers are fairly constant between users, although some users tend to rate mostly their favorites (160 users have a mean rating of 4.0 or above). The MovieLens dataset is supplied with five pre-defined folds for cross validation, and these were also used during the actual testing (see below). The variability between the cross validation folds appears negligible.

There is a large hetrogenity in the rating frequency of both users and items, see Fig. 3. Part (a) presents the number of ratings per user: The mean number of rated items is 106, the median is 65, and the range is from 20 to 737. Similarly, Fig. 3 (b) shows the histogram over the number of ratings per item, in which case the mean is 59, median is 27 and the range is from 1 to 583.

Fig. 4 gives the co-rating matrix, showing which item (x-axis) has been rated by which user (y-axis). An interesting observation is that items apparently have been introduced into the dataset after the rating started; the first user has for instance rated the 272 first movies in the database, but none after that. Similarly, the last movie in the dataset was not rated before user 916 came along. It is also worth noticing that the rating matrix is sparse; only 6.3% of the possible (user, item)-combinations have resulted in a rating in the dataset.

As a final comment on the MovieLens data, we have found that a total of 18 movies are reported twice in the dataset (e.g. the 1993 movie "Body Snatchers" is reported both using ID 573 and ID 670). We could easily have removed these double-entries during pre-processing of the data, but to make sure that our results are comparable to those already reported in the literature, we have chosen to disregard this problem. Looking further into the associated data can also help us understand the fundamental variability we are confronted with in this dataset: Seventeen users have rated both "Body Snatchers (ID 573)" and "Body Snatchers (ID 670)". Out of these, five rated the two items differently, and one user (User ID 617) gave the first item 4, whereas the second item was rated only 1! Similar variability was observed also for the other doubled-registered movies.

The Jester data [14] consists of 4,136,360 ratings using real numbers between -10 and 10 from 73,421 users on 100 jokes. This data is not as sparse as the MovieLens data; 19.2% of the users have rated all the jokes, approximately 17% of the items have been rated by more than 90% of the users, and in total 56.3% of the (user, joke) combinations are given as ratings in the database. The mean rating is 0.74, and the standard deviation is as large as 5.3. The dataset is not supplied with a specific



(a) No. ratings per user (b) No. ratings per movie

Fig. 3. Rating-patterns in the MovieLens dataset.

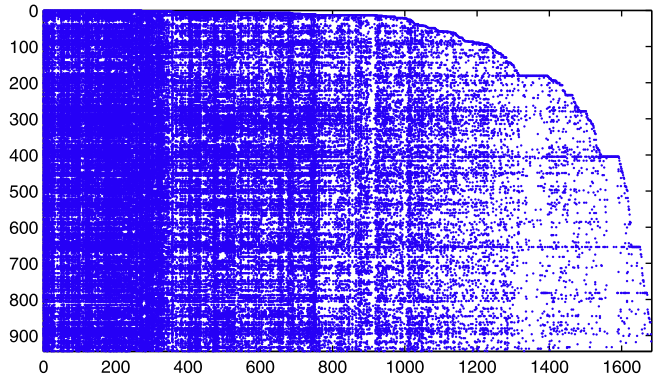


Fig. 4. Co-rating: user vs. movie.

division into cross validation folds; hence the research groups that have reported results on this dataset have used their own privately generated training and test-sets.

6.2. Experimental setup and results

When learning the collaborative filtering model, we used the regularized EM algorithm described in Section 5.2, and for the actual learning we used standard parameter settings: the algorithm terminates when the increase in log-likelihood falls below 10^{-5} or after a maximum of 100 iterations. To decide upon the number of latent variables to describe both users and items (the model structure) and the values for the prior precision of the regression parameters, we used a greedy strategy. The results in Fig. 5 illustrates the procedure; the figure shows the mean absolute error (MAE) as a function of the prior precision α for the regression parameters. The plots are generated for different combinations of latent variables s.t. the plot at position $(|\boldsymbol{U}_p|, |\boldsymbol{M}_i|)$ correspond to a model with $|\boldsymbol{U}_p|$ latent user variables and $|\boldsymbol{M}_i|$ latent movie variables. For example, the bottom-left plot is for a model with 3 latent user variables and 1 latent movie variable. The results shown in these plots are the basis for the greedy learning. We start by choosing $|\boldsymbol{U}_p| = 1$, $|\boldsymbol{M}_i| = 1$, and by setting the prior precision to zero (i.e., no regularization). We then increase the regularization parameter until this harms the MAE; this can, e.g. be calculated using the wrapper approach [28]. Next, we considered non-visited neighboring candidate models that can be reached by either increasing $|\boldsymbol{U}_p|$ or $|\boldsymbol{M}_i|$. This gives the candidate structures $(|\boldsymbol{U}_p| = 1, |\boldsymbol{M}_i| = 2)$ and $(|\boldsymbol{U}_p| = 2, |\boldsymbol{M}_i| = 1)$; both evaluated as above. The best of these two candidate models is chosen (in this case, $(|\boldsymbol{U}_p| = 1, |\boldsymbol{M}_i| = 2)$ was the better option), and we again proceeded by attempting to extend the model in either of the two possible directions. This time, increasing the model size did not pay off in terms of estimated MAE, and we chose to use the candidate model $(|\boldsymbol{U}_p| = 1, |\boldsymbol{M}_i| = 2)$ as our final model. The greedy approach is time saving to the extent that not all structures need to be examined; in our model search only five of the smallest structures were inspected. Furthermore, Fig. 5 indicate that the predictive quality of our model is fairly robust wrt. both structure and reasonable values of the prior precision for the parameters.

An alternative view of this information is given in Fig. 6. Here, the relation between the number of latent variables representing users and movies and the estimated MAE is shown. The minimum MAE is found at $|\boldsymbol{U}_p| = 1$ and $|\boldsymbol{M}_i| = 2$ with an MAE of 0.685 (calculated using a prior precision of 25 for the regression parameters).

Finally, to evaluate the predictive properties of the proposed model, we have empirically compared it with other collaborative filtering algorithms on the same dataset and with the cross-validation folds specified previously. Specifically, we have considered the following straw-men:

Pearson(k) denotes a memory-based approach, where the predicted rating of the active item is calculated as a weighted sum of the ratings given to the k items deemed most important (measured using Pearson correlation) wrt. the active item [19].

Euclidean(k) is the k-nearest neighbors algorithm, where the distance is calculated using Euclidean norm [35].

DM is the decoupled model for rating patterns and intrinsic preferences. This model uses two separate latent variables to explicitly model a user's rating patterns and the intrinsic preference of the users [25].

ML+IMDB(I_1; EQ) is a model combining a collaborative filtering model with content information (from the Internet Movie Database). de Campos et al. [9] investigate several ways of merging the collaborative information with the content information, and the results reproduced here are the best results they obtain.

Triadic uses a latent variable relating the triplet (user, item, rating) to enable a user to have a set of different "reasons" to give an item a specific rating [20]; the results have been reproduced from [9].

FA-U(q) corresponds to the user-centered factor analysis model, where q denotes the number of latent variables [26], see Eq. (5). The model was learned using the EM algorithm with standard parameter settings. The value for q was chosen as the number of latent variables yielding the lowest MAE in the range $[1, 25]$.

FA-I(q) is as for FA-U(q), but with the *item*-centric view.

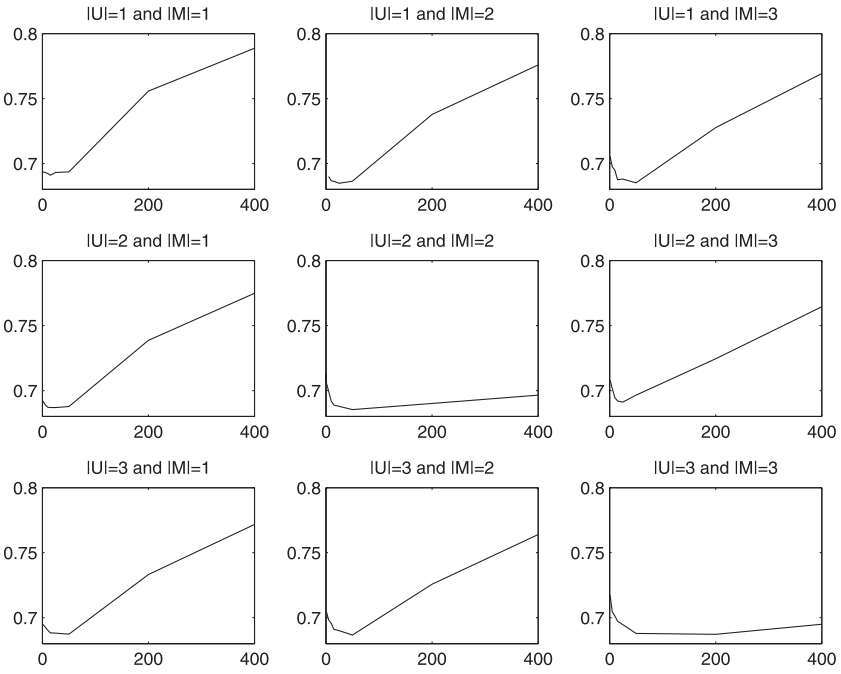


Fig. 5. The figure shows the MAE as a function of the prior precision α for the regression vectors. Each plot corresponds to a certain configuration of the number of latent variables.

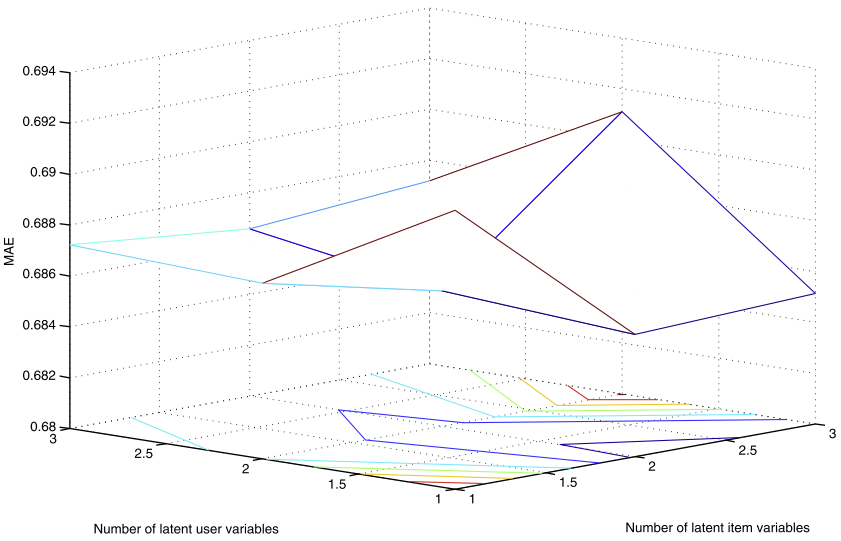


Fig. 6. The figure shows the MAE as a function of the number of latent variables. A minimum (0.685) is found at $|\boldsymbol{U}_p| = 1$ and $|\boldsymbol{M}_i| = 2$.

SVD(q, λ) performs a singular value decomposition in q dimensions. λ is the regularization weight (see Eq. (3)). For each setting of λ we ran experiments with values for q ranging from one to twenty-five, and we represent the best of these results here. Note that when choosing the q-parameter based on the obtained results, we slightly favor the SVD algorithm over the other algorithms. Two options were considered for λ: $\lambda = 0$ resulting in a non-regularized model, and $\lambda = 0.01$ (as done by Salakhutdinov et al. [44]).

The results are shown in Table 6, where we see that the proposed model outperforms the straw-men models on all the folds in the data set; before calculating the MAE we rounded off the predicted ratings to the nearest integer value between one and five as this slightly improved the results (this was done for all models except for DM, ML+IMDB, and Triadic where the originally reported results have been reproduced). Note also that the user-centered factor analysis method selects a single latent variable to encode the correlation among the ratings. This is consistent with the proposed model, where $|\boldsymbol{U}_p| = 1$ is

Table 6
The mean absolute error (MAE) for the MovieLens dataset using the proposed method as well as different straw-men. The MAE is given for each of the five folds together with the average MAE for all the folds. The adjusted t-test [39] was used to compare the classifiers: Results that are significantly poorer than the proposed method at the 10%-level are marked with "*", results significant at the 5%-level are marked with "$^\bullet$", and 1%-level with "†".

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean
Pearson(all)†	0.7225	0.7133	0.7062	0.7063	0.7130	0.7122
Euclidean(all)†	0.7306	0.7195	0.7181	0.7210	0.7211	0.7220
Pearson(10)†	0.7367	0.7297	0.7230	0.7270	0.7311	0.7295
Euclid(10)†	0.7532	0.7354	0.7410	0.7448	0.7488	0.7446
Pearson(25)†	0.7185	0.7071	0.7065	0.6998	0.7082	0.7080
Euclidean(25)†	0.7306	0.7192	0.7237	0.7213	0.7272	0.7244
Pearson(50)†	0.7157	0.7049	0.7133	0.7107	0.7102	0.7110
Euclidean(50)†	0.7373	0.7314	0.7315	0.7335	0.7305	0.7328
Pearson(75)†	0.7140	0.7002	0.7027	0.6982	0.7043	0.7039
Euclidean(75)†	0.7260	0.7147	0.7157	0.7160	0.7205	0.7185
DM†	0.7580	0.7418	0.7284	0.7509	0.7497	0.7458
ML+IMDB(EQ)†	0.7304	0.7206	0.7069	0.7201	0.7209	0.7198
Triadic†	0.7500	0.7369	0.7306	0.7328	0.7324	0.7365
FA/U($q = 1$)†	0.7324	0.7280	0.7257	0.7279	0.7208	0.7269
FA/I($q = 1$)†	0.8048	0.8051	0.8039	0.8000	0.8067	0.8041
SVD($q = 5, \lambda = 0$)$^\bullet$	0.7005	0.6909	0.6971	0.6918	0.6992	0.6959
SVD($q = 4, \lambda = 0.01$)*	0.6987	0.6876	0.6899	0.6893	0.6926	0.6916
CF($\lvert\boldsymbol{U}_p\rvert = 1, \lvert\boldsymbol{M}_i\rvert = 2, \tau = 1/25$)	0.6837	0.6869	0.6846	0.6861	0.6826	0.6848

Table 7
The mean absolute error and the mean squared error for four different subsets of the Jester dataset. The subsets contain 100, 500, 1000, and 2000 users respectively, and the results are given for the proposed model as well as different straw-men models.

	100	500	1000	2000	Mean
Pearson(all)	3.6357/20.9568	3.5661/20.1514	3.6036/20.3130	3.5998/20.5749	3.6013/20.4990
Euclidean(all)	3.6061/21.2762	3.5630/20.3065	3.6249/20.6502	3.6232/20.8279	3.6043/20.7652
Pearson(10)	3.5986/21.1074	3.5748/20.7775	3.6393/21.1773	3.6903/21.7777	3.6258/21.2099
Euclidean(10)	3.6312/21.3128	3.5949/20.8643	3.6764/21.3474	3.7043/21.9191	3.6517/21.3609
Pearson(25)	3.5883/20.9083	3.4995/19.9487	3.5554/20.1860	3.5825/20.5863	3.5564/20.4073
Euclidean(25)	3.6129/21.0946	3.5475/20.1318	3.6151/20.5942	3.6341/20.9933	3.6024/20.7035
Pearson(50)	3.5666/20.4993	3.4848/19.7149	3.5282/19.8560	3.5473/20.1967	3.5317/20.0667
Euclidean(50)	3.6147/21.2668	3.5376/19.9798	3.5966/20.3322	3.6071/20.6362	3.5890/20.5538
Pearson(75)	3.6371/20.9768	3.4851/19.6928	3.5263/19.8025	3.5355/20.0705	3.5460/20.1357
Euclidean(75)	3.6057/21.2597	3.5366/19.9969	3.5972/20.3051	3.6004/20.5383	3.5849/20.5250
FA/U	3.7304/22.9915	3.5840/20.5061	3.6307/20.6807	3.6125/20.6852	3.6394/21.2159
SVD($\lambda = 0$)	3.6071/21.0929	3.5042/20.6272	3.6242/22.3482	3.5433/21.5128	3.5697/21.3953
SVD($\lambda = 0.01$)	3.4768/20.2950	3.5313/21.0954	3.6612/23.2504	3.5591/21.4473	3.5571/21.5220
CF	3.5646/20.6831	3.4934/20.2447	3.4590/19.2550	3.4435/19.4577	3.4901/19.9101

chosen. For the item-centered factor analysis model, results were best for small number of factors, and with $q = 1$ marginally better than $q = 2$ overall. Also this result is related with the results of the proposed model, where $|\boldsymbol{M}_i| = 2$ is selected.[9]

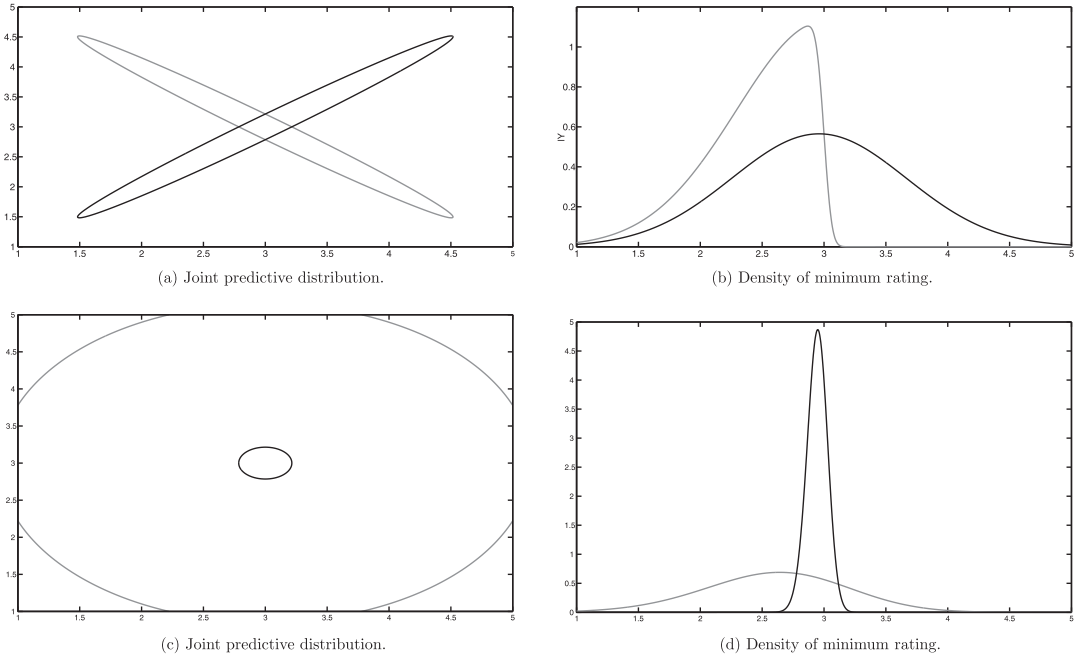
It is difficult to find results in the scientific literature that are directly comparable to ours, mainly because the experimental setting is different. Many researchers using the MovieLens dataset have made their own training and test sets without further documentation. However, the reported MAE values are typically about 0.73–0.74 or poorer [19,48,37,32,38,27,7,40,31,55].

For the Jester dataset, no predefined training/test-set division of the data is given and in our test setup we have therefore randomly selected 80% of the data for training and 20% for testing. The size of the original dataset does, however, cause complexity problems for the current learning algorithm: recall that we need the full covariance matrix over the latent variables for all users and items, hence with, e.g. $|\boldsymbol{U}_p| = |\boldsymbol{M}_i| = 2$ we would be working with a covariance matrix of size 147,042$\times$ 147,042. Instead we have randomly selected four subsets from the database containing 100, 500, 1000, and 2000 users, respectively. For the actual learning we fixed the precision on the regression vectors to 25 based on preliminary experiments, and for finding an appropriate model structure we used the greedy search method described above. The results of the experiments can be see in Table 7.

6.3. Group recommendations

Next, let us turn to the multi-rating aspect of our model, as outlined in Section 4.3.2. To exemplify, we will again focus on the MovieLens dataset, and we have initially chosen to restrict our attention to the first cross-validation split of the dataset. This gives us a dataset of 80,000 ratings from which we have learned a model with $|\boldsymbol{M}_i| = 2$ and $|\boldsymbol{U}_p| = 1$. We have somewhat arbitrarily chosen the two users "User ID 49" and "User ID 279"; let us call them One and Two in the following. The two were selected mainly because both users have rated a significant number of movies in both the training set and the

[9] The number of latent variables in this model equals $\#M \cdot |\boldsymbol{M}_i| + \#U \cdot |\boldsymbol{U}_p| = 4307$.



(a) Joint predictive distribution. (b) Density of minimum rating.

(c) Joint predictive distribution. (d) Density of minimum rating.

Fig. 7. The effect the predictive covariance matrix has on group recommendations.

test set. One has rated 107 movies in the training data with an average rating of 2.73. Among One's favourites are "classic" comedies, like "Monty Python's Life of Brian", "In the Company of Men", and "This is Spinal Tap". Two has rated 242 movies in the training-set with an average score of 3.17 stars. Two is also fond of comedies, and in particular action-comedies, like "Men in Black", "Blues Brothers", and "Bad Boys". There are 18 movies that are rated by both users in the training set, and even though the users apparently share an interest in comedies, their empirical Pearson correlation coefficient is -0.59. The movie "Harold and Maude", a romantic comedy from 1971, is for instance given the maximum score by One, and the minimum score by Two. Similarly, "Jackie Chan's First Strike", an action/comedy from 1996, is loved by Two but hated by One.

For a given utility function (like the ones proposed in Section 4.3.2), the system must look at all movies that are unrated by both users, and find the one that maximises the expected utility. However, in order to be able to evaluate the recommendation process, we here restrict our attention to the 21 movies that both users have rated in the test-set. These movies mostly include comedies of different varieties, like "Cold Comfort Farm", "Addams Family Values", "The Cable Guy", "Monty Python and the Holy Grail", and "Dave", but also some thriller movies and a drama. To find a movie that both users can enjoy, the system needs to understand the more subtle reasons why a user finds some comedies funnier than others, or alternatively to find a quality movie outside that genre.

If we try to find a movie that fits One (disregarding Two's preferences), the system recommends "Cold Comfort Farm", a romantic comedy released directly for TV in 1995. This turns out to be a reasonable guess, as One gave this movie 4 stars. However, Two only gave this movie one star, and the movie is thus not a good choice for the pair to watch together. On the other hand, the system would suggest the thriller "The Crow" if it only considers the preferences of Two. This is also a reasonable suggestion, as Two gave this movie 4 stars. However, One gives it only a single star, and again the system has not found a good movie for the two users together. Using the independence-definition makes the system choose "Cold Comfort Farm", as it did when disregarding Two, and which is not really a choice that fits the pair well (recall that Two gave this movie a single star). The Maximin utility function of Section 4.3.2 is, on the other hand, specially defined to select a movie that both users will enjoy. This is obtained by looking for a movie that one can be pretty sure neither will dis-like. To this end, the system ends up suggesting the movie "Brazil" (1985) by Terry Gilliam from Monty Python. This suggestion is perfect, as it was given five stars by both One and Two.

Let us examine further the underlying mathematics of the Maximin predictions. Assume that we calculate the joint predicted ratings of two users for a given movie. These predictions will be in terms of a bivariate Normal distribution, and two possible examples are shown in Fig. 7 (a). Both distributions share the same mean and marginal variances, but where the black ellipsoid shows the distribution for positively correlated predictions (Pearson correlation coefficient $\rho = +0.99$), the grey ellipsoid depicts a distribution with negative correlation ($\rho = -0.99$); each ellipsoid contain 90% of the associated probability mass. As the two predictive distributions have the same means and marginal variances, they will lead to identical recommendations when the Independence utility function is employed. On the other hand, this is not the case for the Maximin utility function. For a strongly positively correlated predictive distribution (black ellipsoid), the two ratings will be close to identical, and the minimum rating is thus almost equal to either rating. The distribution of the minimum rating, which in this case appears to be "almost Gaussian" (it is identical to a Gaussian when the Pearson correlation coefficient equals one),

Table 8. Users pairs were Maximin recommendation differs from the Independence recommendation.

User	IDs	Independence recommendation	Ratings			Maximin recommendation	Ratings			Diff.
			R1	R2	min		R1	R2	min	
175	363	Field of Dreams (1989)	5	3	3	Alien (1979)	4	4	4	1
128	409	Hoop Dreams (1994)	4	2	2	Star Wars (1977)	4	5	4	2
14	296	Pulp Fiction (1994)	5	5	5	The Silence of the Lambs (1991)	3	5	3	-2
217	328	Psycho (1960)	3	4	3	Braveheart (1995)	5	5	5	2
299	303	Citizen Kane (1941)	4	5	4	Schindler's List (1993)	4	5	4	0
269	321	Dr. Strangelove or: How I Learned ...(1963)	4	4	4	Casablanca (1942)	4	5	4	0
57	250	Pulp Fiction (1994)	3	4	3	Back to the Future (1985)	4	2	2	-1
24	269	Dead Man Walking (1995)	5	4	4	Fargo (1996)	5	5	5	1
151	426	The Big Sleep (1946)	4	4	4	The Silence of the Lambs (1991)	4	4	4	0
437	608	Leaving Las Vegas (1995)	5	2	2	On Golden Pond (1981)	4	3	3	1
433	435	Dr. Strangelove or: How I Learned ...(1963)	3	3	3	The Usual Suspects (1995)	5	5	5	2
62	326	Casablanca (1942)	4	5	4	Raiders of the Lost Ark (1981)	4	4	4	0
472	487	The Terminator (1984)	5	4	4	Return of the Jedi (1983)	5	4	4	0
59	354	Dead Man Walking (1995)	4	3	3	Three Colors: Red (1994)	5	5	5	2
56	371	The Rock (1996)	5	3	3	Indiana Jones and the Last Crusade (1989)	5	4	4	1
271	450	Indiana Jones and the Last Crusade (1989)	4	3	3	Groundhog Day (1993)	4	4	4	1
524	606	The Terminator (1984)	2	5	2	The African Queen (1951)	5	4	4	2
314	504	Corrina, Corrina (1994)	4	3	3	Four Weddings and a Funeral (1994)	1	3	1	-2
543	661	Good Will Hunting (1997)	3	4	3	North by Northwest (1959)	4	5	4	1
655	667	Taxi Driver (1976)	3	3	3	Good Will Hunting (1997)	3	5	3	0
881	942	E.T. the Extra-Terrestrial (1982)	4	5	4	Star Wars (1977)	3	5	3	-1
409	881	The Godfather (1972)	4	4	4	One Flew Over the Cuckoo's Nest (1975)	5	5	5	1
764	805	Pulp Fiction (1994)	4	4	4	Raiders of the Lost Ark (1981)	5	3	3	-1
514	645	Apocalypse Now (1979)	3	4	3	Amadeus (1984)	5	5	5	2
524	781	Pulp Fiction (1994)	4	3	3	L.A. Confidential (1997)	5	5	5	2
650	897	Raiders of the Lost Ark (1981)	4	5	4	The Princess Bride (1987)	5	3	3	-1
267	889	2001: A Space Odyssey (1968)	5	2	2	The Terminator (1984)	4	4	4	2
548	592	The Godfather (1972)	5	5	5	Alien (1979)	5	5	5	0
661	882	Terminator 2: Judgment Day (1991)	4	4	4	The Bridge on the River Kwai (1957)	5	5	5	1

is depicted with the black line in Fig. 7 (b). When the predictive distribution is negatively correlated, the distribution of the minimum is distinctly non-Gaussian (grey line in Fig. 7 (b)). Similar results are obtained when considering the effect of the predictive variances: The two predictive distributions in Fig. 7 (c) differ only by their variance, one having a variance of 0.01 (black circle), the other a variance of 1.0 (grey circle). The corresponding distributions of the minimum ratings are shown in Fig. 7 (d), where we can see that lower predictive variance will be preferred when using the Maximin utility function. The lesson learned from this analysis is that generating group recommendations is potentially far more difficult than producing recommendations for a single user, as the predictive covariance can play a key role in the group recommendation process. This is in stark contrast to single user recommendations, where it is sufficient to use the predictive mean.

We end this discussion by evaluating the importance of these effects in the MovieLens data.[10] For each of the five predefined cross validation folds, we randomly selected 50 user pairs making sure that at least five movies were rated by both users in the test-set. For each pair, we looked at the items both users have rated in the test-set, and considered for each type of utility function which of these movies to recommend for that user pair. Independence and Maximin behaved differently in the 29 cases listed in Table 8. The table gives the user IDs of the two randomly selected users, the recommendation based on the independence utility function, and the observed ratings the two users gave the recommended item together with the anticipated group-evaluation (calculated as the minimum of the two users' ratings). This is followed by the same information for the Maximin utility function, and finally we calculate the difference in the group ratings for the two suggested items. A positive difference means that the Maximin approach gives the better recommendation, a negative value means that the independence-approach was superior. Results from different folds are separated by a dashed line. Overall, the Maximin approach seems to be slightly better than the independence assumption, with improvement shown in 18 of the 29 cases. The independence approach is better in 6 cases, and 5 cases are drawn. By further examination, we see that the independence approach often recommends "Pulp Fiction" from 1994, a movie that is highly regarded albeit controversial due to its level of violence. These characteristics lead to a high predictive mean rating for the movie, but also a large predictive variance, thus making it less attractive seen from the Maximin utility function's point of view. Correspondingly, the Maximin approach seems to have a bias towards less debatable "classics", again in correspondence with our mathematical intuition.

7. Conclusions

In this paper we have proposed a new model for collaborative filtering, where the traditional user and item perspectives are combined into a single (relational) model. We have shown how to learn these models from rating-data using the EM-algorithm, and we have demonstrated that the framework offers very good predictive abilities. Furthermore, through examples we have shown that our model also carries implicit information about the domain captured in its latent variables. We anticipate that this information can be utilized to explain model predictions for a user and thereby increase the user's trust in the recommendations, and we are currently in the process of considering how this information can be used to generate explanations automatically.

The main contribution of the paper is the proposed model class together with the model learning algorithms and the analysis of the properties of the learned models. In particular, for the empirical experiments we have relied on exact inference algorithms when learning and analyzing the models, thus putting less emphasis on computational complexity. Using exact inference algorithms when learning models for large data sets will, however, be prohibitive in general. An immediate direction for future research is therefore the design of efficient approximate inference algorithms (e.g. based on variational approximations) tailored specifically to the proposed model class.

Other directions for future research include extending the model to allow a flexible and seamless integration of content information. We anticipate that content information will mainly be represented by discrete variables, and a particular challenge will therefore be the complexity of the model.

Appendix A. The EM algorithm

In this section we specify the EM algorithm for the proposed model. First of all, we note that the joint probability distribution over $(\boldsymbol{R}, \boldsymbol{U}, \boldsymbol{M})$ can be expressed as

$$f(\boldsymbol{r}, \boldsymbol{u}, \boldsymbol{m}) = f(\boldsymbol{r}|\boldsymbol{m}, \boldsymbol{u}) f(\boldsymbol{m}) f(\boldsymbol{u}),$$

where

$$\begin{aligned} f(\boldsymbol{r}|\boldsymbol{m}, \boldsymbol{u}) &= \prod_{p=1}^{N} \prod_{i \in \mathcal{I}(p)} (2\pi\theta)^{-1/2} \exp\left(-\frac{1}{2\theta}(\boldsymbol{r}(p,i) - (\boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{m}_i + \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{u}_p + \phi_p + \psi_i))^2\right) \\ f(\boldsymbol{m}_i) &= \mathcal{N}(\boldsymbol{0}_s, \boldsymbol{I}_{s\times s}); \\ f(\boldsymbol{u}_p) &= \mathcal{N}(\boldsymbol{0}_t, \boldsymbol{I}_{t\times t}). \end{aligned}$$

[10] We would have liked to be able to perform a more systematic analysis of the multi-rating prediction problem, but we are unfortunately not aware of any databases supporting this kind of analysis.

The M-step for the EM algorithm can now be derived by considering the partial derivatives of the expected data-complete log-likelihood of the model:

$$\begin{aligned}\mathcal{Q} = &-\frac{\#M \cdot s}{2}\log(2\pi) - \frac{\#M}{2}\mathbb{E}(\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M}) - \frac{\#U \cdot t}{2}\log(2\pi) - \frac{\#U}{2}\mathbb{E}(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}) \\ &- \frac{d}{2}\log(2\pi) - \frac{d}{2}\log(\theta) \\ &- \frac{1}{2\theta}\sum_{p=1}^{\#U}\sum_{i\in\mathcal{I}(p)}\mathbb{E}((\boldsymbol{r}(p,i) - (\boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{M}_i + \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{U}_p + \phi_p + \psi_i))^2),\end{aligned}$$

where $d = \sum_{p=1}^{\#U}|\mathcal{I}(p)|$, $\#M$ is the number of movies, and $\#U$ is the number of users. Note that the expectations are implicitly conditioned on the observed ratings.

For the standard deviation θ we now get

$$\frac{\partial\mathcal{Q}}{\partial\theta} = \frac{-d}{2\theta} + \frac{1}{2\theta^2}\sum_{p=1}^{\#U}\sum_{i\in\mathcal{I}(p)}\mathbb{E}[(\boldsymbol{r}(p,i) - (\boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{M}_i + \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{U}_p + \phi_p + \psi_i))^2]$$

and the updating rule for θ therefore becomes

$$\hat{\theta} \leftarrow \frac{1}{d}\sum_{p=1}^{\#U}\sum_{i\in\mathcal{I}(p)}\mathbb{E}[(\boldsymbol{r}(p,i) - (\boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{M}_i + \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{U}_p + \phi_p + \psi_i))^2],$$

which involves the expectations $\mathbb{E}(\boldsymbol{U}_p)$, $\mathbb{E}(\boldsymbol{M}_i)$, $\mathbb{E}(\boldsymbol{M}_i\boldsymbol{M}_i^{\mathrm{T}})$, $\mathbb{E}(\boldsymbol{M}_i\boldsymbol{U}_p^{\mathrm{T}})$, and $\mathbb{E}(\boldsymbol{U}_p\boldsymbol{U}_p^{\mathrm{T}})$.

For $\boldsymbol{v}_p$ we get

$$\frac{\partial\mathcal{Q}}{\partial\boldsymbol{v}_p} = \frac{1}{\theta}\sum_{i\in\mathcal{I}(p)}(\mathbb{E}(\boldsymbol{M}_i\boldsymbol{M}_i^{\mathrm{T}})\boldsymbol{v}_p - \mathbb{E}(\boldsymbol{M}_i)r_{p.i} + \mathbb{E}(\boldsymbol{M}_i\boldsymbol{U}_p^{\mathrm{T}})\boldsymbol{w}_i + \mathbb{E}(\boldsymbol{M}_i)(\phi_p + \psi_i))$$

and therefore

$$\hat{\boldsymbol{v}_p} \leftarrow \left[\sum_{i\in\mathcal{I}(p)}\mathbb{E}(\boldsymbol{M}_i\boldsymbol{M}_i^{\mathrm{T}})\right]^{-1}\left[\sum_{i\in\mathcal{I}(p)}(\mathbb{E}(\boldsymbol{M}_i)\boldsymbol{r}(p,i) - \mathbb{E}(\boldsymbol{M}_i\boldsymbol{U}_p^{\mathrm{T}})\boldsymbol{w}_i - \mathbb{E}(\boldsymbol{M}_i)(\phi_p + \psi_i))\right].$$

The updating rule for ϕ_p follows from

$$\frac{\partial\mathcal{Q}}{\partial\phi_p} = \frac{1}{\theta}\sum_{i\in\mathcal{I}(p)}(\boldsymbol{r}(p,i) - (\boldsymbol{v}_p^{\mathrm{T}}\mathbb{E}(\boldsymbol{M}_i) + \boldsymbol{w}_i^{\mathrm{T}}\mathbb{E}(\boldsymbol{U}_p) + \phi_p + \psi_i)),$$

and is given by

$$\hat{\phi}_p \leftarrow \frac{1}{|\mathcal{I}(p)|}\sum_{i\in\mathcal{I}(p)}(\boldsymbol{r}(p,i) - (\boldsymbol{v}_p^{\mathrm{T}}\mathbb{E}(\boldsymbol{M}_i) + \boldsymbol{w}_i^{\mathrm{T}}\mathbb{E}(\boldsymbol{U}_p) + \psi_i)).$$

Finally, analogously to the updating rules for $\boldsymbol{v}_p$ and ϕ_p, we have the following rules for $\boldsymbol{w}_i$ and ψ_i:

$$\hat{\boldsymbol{w}}_i \leftarrow \left[\sum_{p\in\mathcal{P}(i)}\mathbb{E}(\boldsymbol{U}_p\boldsymbol{U}_p^{\mathrm{T}})\right]^{-1}\left[\sum_{p\in\mathcal{P}(i)}\mathbb{E}(\boldsymbol{U}_p)\boldsymbol{r}(p,i) - \mathbb{E}(\boldsymbol{U}_p\boldsymbol{M}_i^{\mathrm{T}})\boldsymbol{v}_p - \mathbb{E}(\boldsymbol{U}_p)(\phi_p + \psi_i)\right]$$

$$\hat{\psi}_i \leftarrow \frac{1}{|\mathcal{P}(i)|}\sum_{p\in\mathcal{P}(i)}(\boldsymbol{r}(p,i) - (\boldsymbol{v}_p^{\mathrm{T}}\mathbb{E}(\boldsymbol{M}_i) + \boldsymbol{w}_i^{\mathrm{T}}\mathbb{E}(\boldsymbol{U}_p) + \phi_p)).$$

The required expectations can be calculated from the joint distribution over the latent variables conditioned on the observed ratings:

$$[\boldsymbol{U}^{\mathrm{T}}, \boldsymbol{M}^{\mathrm{T}}]^{\mathrm{T}}|\boldsymbol{r} \sim \mathcal{N}\left(\boldsymbol{\Sigma}(\boldsymbol{L}^{\mathrm{T}}\theta^{-1}(\boldsymbol{r} - (\phi + \psi))), \boldsymbol{\Sigma}\right),$$

where the covariance matrix is given by

$$\boldsymbol{\Sigma} = (\boldsymbol{I} + \boldsymbol{L}^{\mathrm{T}}\theta^{-1}\boldsymbol{L})^{-1}.$$

and $\boldsymbol{L}$ is the regression matrix for the ratings given $\boldsymbol{U}$ and $\boldsymbol{M}$ (i.e., consisting of the $\boldsymbol{v}_p$s and $\boldsymbol{w}_i$s).

Specifically, $\mathbb{E}(\boldsymbol{U}_p)$ and $\mathbb{E}(\boldsymbol{M}_i)$ can be extracted directly from the mean vector, and, e.g. $\mathbb{E}(\boldsymbol{M}_i\boldsymbol{U}_p^{\mathrm{T}})$ can be calculated as

$$\mathbb{E}(\boldsymbol{M}_i\boldsymbol{U}_p^{\mathrm{T}}) = \boldsymbol{\Sigma}_{i,p} - \mathbb{E}(\boldsymbol{M}_i)\mathbb{E}(\boldsymbol{U}_p)^{\mathrm{T}},$$

where $\boldsymbol{\Sigma}_{i,p}$ is the sub-matrix of $\boldsymbol{\Sigma}$ restricted to the variables $\boldsymbol{M}_i$ and $\boldsymbol{U}_p$.

References

[1] A.J. Bertie, G.W. Cran, Estimation of the constant term when using ridge regression, International Journal of Mathematical Education in Science and Technology 16 (1985) 63–65.
[2] D. Billsus, M.J. Pazzani, Learning collaborative information filters, in: Proceedings of the Fifteenth International Conference on Machine Learning, 1998, pp. 46–54.
[3] C.M. Bishop, Pattern Recognition and Machine Learning, Information Science and Statistics, Springer-Verlag, 2006.
[4] D.M. Blei, A.Y. Ng, M.I. Jordan, J. Lafferty, Latent Dirichlet allocation, Journal of Machine Learning Research 3 (2003) 2003.
[5] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, 1998, pp. 43–52.
[6] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (8) (2011) 1548–1560.
[7] J. Chen, J. Yin, Recommendation based on influence sets, in: Proceedings of the Workshop on Web Mining and Web Usage Analysis, 2006.
[8] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, M.A. Rueda-Morales, Managing uncertainty in group recommending processes, User Modeling and User-Adapted Interaction 19 (2009) 207–242.
[9] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, M.A. Rueda-Morales, Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks, International Journal of Approximate Reasoning 51 (2010) 785–799.
[10] N. Delannay, M. Verleysena, Collaborative filtering with interlaced generalized linear models, Neurocomputing 71 (2008) 1300–1310.
[11] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B 39 (1977) 1–38.
[12] P. Domingos, M. Pazzani, On the optimality of the simple Bayesian classifier under zero-one loss, Machine Learning 29 (2–3) (1997) 103–130.
[13] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, New York, 1973.
[14] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: a constant time collaborative filtering algorithm, Information Retrieval 4 (2002) 133–151.
[15] Q. Gu, J. Zhou, C.H.Q. Ding, Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs, in: The SIAM International Conference on Data Mining, 2010, pp. 199–210.
[16] D. Gupta, M. Digiovanni, H. Narita, K. Goldberg, Jester 2.0 (poster abstract): evaluation of an new linear time collaborative filtering algorithm, Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 1999, pp. 291–292.
[17] T. Hastie, R. Tibshirani, J. Friedman, Elements of Statistical Learning: Data Mining, Inference and Prediction, Springer-Verlag, New York, 2001.
[18] D. Heckerman, D. Chickering, C. Meek, R. Rounthwaite, C. Kadie, Dependency networks for inference, collaborative filtering and data visualization, Journal of Machine Learning Research 1 (2000) 49–75.
[19] J. Herlocker, J. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, in: Proceedings of the ACM 1999 Conference on Research and Development in Information Retrieval, 1999, pp. 230–237.
[20] T. Hofmann, Learning what people (don't) want, Proceedings of the Twelfth European Conference on Machine Learning, Springer-Verlag, London, UK, 2001, pp. 214–225.
[21] T. Hofmann, Latent semantic models for collaborative filtering, ACM Transactions on Information Systems 22 (1) (2004) 89–115.
[22] T. Hofmann, J. Puzicha, Latent class models for collaborative filtering, Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999, pp. 688–693.
[23] A. Jameson, B. Smyth, The Adaptive Web, Springer-Verlag, Berlin, Heidelberg, 2007., pp. 596–627 (Chapter Recommendation to groups).
[24] F.V. Jensen, T.D. Nielsen, Bayesian Networks and Decision Graphs, Springer-Verlag, Berlin, Germany, 2007.
[25] R. Jin, L. Si, C. Zhai, A study of mixture models for collaborative filtering, Information Retrieval 9 (3) (2006) 357–382.
[26] M. Kendall, Multivariate Analysis, second ed., Charles Griffin & Co., London, UK, 1980.
[27] D. Kim, B.-J. Yum, Collaborative filtering based on iterative principal component analysis, Expert Systems with Applications 28 (4) (2005) 823–830.
[28] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1–2) (1997) 273–324.
[29] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, IEEE Computer 42 (2009) 30–37.
[30] H. Langseth, T.D. Nielsen, Latent classification models, Machine Learning 59 (3) (2005) 237–265.
[31] G. Lekakos, P. Caravelas, A hybrid approach for movie recommendation, Multimedia Tools and Applications 36 (2008) 5570.
[32] Q. Li, B.M. Kim, Clustering approach for hybrid recommender system, WI'03: Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence, IEEE Computer Society, Washington, DC, USA, 2003, pp. 33–38.
[33] Y.J. Lim, Y.W. Teh, Variational Bayesian approach to movie rating prediction, in: Proceedings of KDD Cup and Workshop, 2007, pp. 15–21.
[34] B. Marlin, Modeling user rating profiles for collaborative filtering, Advances in Neural Information Processing Systems, vol. 15, The MIT Press, 2003, pp. 627–634.
[35] B. Marlin, Collaborative filtering: a machine learning perspective. Master of Science Thesis, Graduate Department of Computer Science, University of Toronto, 2004.
[36] J. Masthoff, Group recommender systems: combining individual models, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Recommender Systems Handbook, Springer, US, 2011, pp. 677–702.
[37] P. Melville, R. Mooney, R. Nagarajan, Content-boosted collaborative filtering for improved recommendations, Proceedings of the Eighteenth National Conference on Artificial Intelligence, The AAAI Press, 2002, pp. 178–192.
[38] B. Mobasher, X. Jin, Y. Zhou, Semantically enhanced collaborative filtering on the web, in: Web Mining: From Web to Semantic Web, First European Web Mining Forum, EMWF 2003, Lecture Notes in Computer Science, vol. 3209, 2003, pp. 57–76.
[39] C. Nadeau, Y. Bengio, Inference for the generalization error, Machine Learning 52 (3) (2003) 239–281.
[40] S.-T. Park, D. Pennock, O. Madani, N. Good, D. DeCoste, Naïve filterbots for robust cold-start recommendations, in: KDD+06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 699–705.
[41] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, San Mateo, CA, 1988.
[42] D.M. Pennock, E. Horvitz, S. Lawrence, C.L. Giles, Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach, Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, 2000, pp. 473–480.

[43] M.H. Pryor, The effects of singular value decomposition on collaborative filtering, Tech. Rep. PCS-TR98-338, Dartmouth Computer Science, 1998.
[44] R. Salakhutdinov, A. Mnih, G. Hinton, Restricted Boltzmann machines for collaborative filtering, in: Proceedings of the Twenty-fourth International Conference on Machine Learning, vol. 24. 2007, pp. 791–798.
[45] R. Salakhutdinow, A. Mnih, Bayesian probabilistic matrix factorization using Markov chain monte carlo, Proceedings of the Twenty-Fifth International Conference on Machine Learning, Omnipress, 2008, pp. 880–887.
[46] B.M. Sarwar, G. Karypis, J.A. Konstan, J.T. Riedl, Application of dimensionality reduction in recommender systems – a case study, Tech. Rep. CS-TR 00-043, Computer Science and Engineering Department, University of Minnesota, 2000.
[47] E. Savia, K. Puolamäki, J. Sinkkonen, S. Kaski, Two-way latent grouping model for user preference prediction, in: Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence, 2005, pp. 518–525.
[48] A. Schein, A. Popescul, L. Ungar, D. Pennock, Generative models for cold-start recommendations, in: Proceedings of the 2001 SIGIR Workshop on Recommender Systems, 2001.
[49] L. Si, R. Jin, Flexible mixture model for collaborative filtering, in: Proceedings of the Twentieth International Conference on Machine Learning, National Conference on Artificial Intelligence, 2003, pp. 704–711.
[50] N. Srebro, T. Jaakkola, Weighted low-rank approximations, in: Proceedings of the Twentieth International Conference on Machine Learning, National Conference on Artificial Intelligence, 2003, pp. 720–727.
[51] T.T. Truyen, D.Q. Phung, S. Venkatesh, Preference networks: Probabilistic models for recommendation systems, in: Sixth Australasian Data Mining Conference (AusDM 2007), CRPIT. ACS, Gold Coast, Australia, 2007, pp. 195–202.
[52] T.T. Truyen, D.Q. Phung, S. Venkatesh, Ordinal boltzmann machines for collaborative filtering, in: Proceedings of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence, 2009, pp. 548–556.
[53] J. Wang, A.P. de Vries, M.J.T. Reinders, Unifying user-based and item-based collaborative filtering approaches by similarity fusion, SIGIR'06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, 2006, pp. 501–508.
[54] Z. Xu, V. Tresp, K. Yu, H.-P. Kriegel, Infinite hidden relational models, in: Proceedings of the Twenty-second Conference on Uncertainty in Artificial Intelligence, 2006.
[55] K. Yoshii, M. Goto, K. Komatani, T. Ogata, H. Okuno, An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model, IEEE Transaction on Audio, Speech and Language Processing 16 (2008) 435–447.