AALBORG UNIVERSITY

TECHNICAL REPORT

HELGE LANGSETH AND THOMAS D. NIELSEN:

A latent model for collaborative filtering

Cite as: *Helge Langseth and Thomas D. Nielsen*:
A latent model for collaborative filtering
Technical Report 09-003,
Department of Computer Science,
Aalborg University

Date: 24/09 2009

A latent model for collaborative filtering

Helge Langseth
Department of Computer and Information Science,
The Norwegian University of Science and Technology,
NO-7491 Trondheim, Norway
helgel@idi.ntnu.no

Thomas D. Nielsen
Department of Computer Science,
Aalborg University,
DK-9220 Aalborg Ø, Denmark
tdn@cs.aau.dk

September 24, 2009

Abstract

Recommender systems based on collaborative filtering have received a great deal of interest over the last decade. Typically, these types of systems either take a user-centered or an item-centered approach when making recommendations, but by employing only one of these two perspectives we may unintentionally leave out important information that could otherwise have improved the recommendations. In this paper, we propose a collaborative filtering model that contains an explicit representation of all items and users. 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.

Keywords: Recommender systems; Collaborative filtering; Graphical models; Latent variables

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 (Breese et al, 1998), although hybrid systems have also been developed (Pennock et al, 2000). 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 (Duda and Hart, 1973)) for either grouping users into user-groups or items into item-categories. More elaborate model structures have also been developed (see e.g. Heckerman et al (2000)), but common for most of these approaches is that they rely on a single item-model and/or user-model for predicting preferences. As a consequence, the models may fail to exploit information involving other users and items. For example, a simple multinomial mixture model for user clustering does not directly exploit information about other users when clustering the active user.

In this paper we propose a 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 model combining both an item perspective and a user perspective. The generative properties of the model support a natural model interpretation, and empirical results demonstrate that the proposed model outperforms other memory-based and model-based approaches. Finally, by having all users represented in the same model, the proposed system supports joint recommendations for several users.

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 (Pearl, 1988; Jensen and Nielsen, 2007) 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 Figure 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:

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

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 (Breese et al, 1998) and dependency networks (Heckerman et al, 2000). 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 (Breese et al, 1998), 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 (Duda and Hart, 1973)). 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 (Domingos and Pazzani, 1997; Langseth and Nielsen, 2005). 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 (Jan and Puzicha, 1999; Hofmann, 2001, 2004) addresses some of the inherent limitations of the mixture model by allowing users to have different prior distributions over the user classes. This idea is further pursued by Marlin (2003) who 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 (Blei et al, 2003), and relies on approximate methods like variational methods or Gibbs sampling for inference and parameter learning. This model has been further explored by Savia et al (2005) who extend the latent model structure to cover both users and items. For a comparison and discussion on alternative models, including the aspect model and the flexible mixture model (Si and Jin, 2003), see Jin et al (2006).

There has also been investigations into so-called hybrid recommendation systems. For example, recommendation systems based on a unification of collaborative and content-based approaches have been considered (Popescul et al, 2001). Pennock et al (2000) propose 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. Also, Wang et al (2006) proposed a method for unifying the user-based and item-based collaborative filtering approaches within a memory-based context.

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 (Xu et al, 2006). 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 share 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 $\boldsymbol{\theta_r}$, i.e., we assume the existence of a function $g(\cdot)$ s.t. $\boldsymbol{r}(p,i) = g(\boldsymbol{\theta_r}, p, i)$ 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 $g(\boldsymbol{\theta_r}, p', i')$. This process is called *single-rating predictions*. Often, $g(\cdot)$ will be based on a statistical model of the conditional distribution of $\boldsymbol{R}(p,i)\,|\{\boldsymbol{r}, \boldsymbol{\theta_r}\}$, and the prediction is then either the expected value or the median value of that conditional distribution.[1] A more complicated problem is *multi-rating predictions*: One may, for instance, want to find items that a group of users (persons p_1 and p_2, say) will enjoy together. A naïve solution to the current example is to consider the multi-rating problem as a collection of single-rating problems, and then use $g(\boldsymbol{\theta_r}, p_1, i) + g(\boldsymbol{\theta_r}, p_2, i)$ 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. Only few CF systems give full support to multi-rating predictions.

4.1 A Data Compression Model

The main idea we will pursue when building our CF system is *data compression*, i.e., to find a representation $\boldsymbol{\theta_r}$ that is more compact than representing the original $\#U \times \#M$-matrix $\boldsymbol{R}$. The first approach we will describe is to assume 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. $\boldsymbol{\theta_r} = \{\boldsymbol{V}, \boldsymbol{W}\}$), and choose $\boldsymbol{\theta_r}$ s.t. $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{W}$ is the best rank-q approximation of $\boldsymbol{R}$. Here $q \le \min(\#U, \#M)$ 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 $g(\boldsymbol{\theta_r}, p, i) = \boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{w}_i$. 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 $\boldsymbol{V}$ and $\boldsymbol{W}$ that minimizes the observed error over the ratings. It is common to consider the squared error, i.e., the Frobenius norm denoted

[1]See Marlin (2004) for a discussion of the relative merits of these estimators.

by $\|\cdot\|_F$. Thus, the learning task can be stated as the following minimization problem:

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

We know how to solve Equation (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 (Salakhutdinov et al, 2007). This is an idea eagerly explored in the CF community (Truyen et al, 2009), where one of the leading approaches is to numerically minimize the objective function

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

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

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

where η is the learning rate.

One apparent problem with Equation (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

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

where λ is a parameter that balances parameter regularization and model fit (typically chosen as $\lambda \sim 5 \cdot 10^{-3}$).

4.2 A Simple Generative Model

Another 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,

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

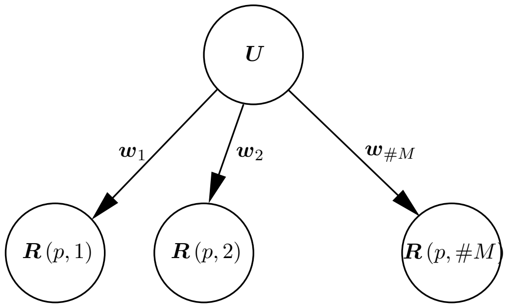

Figure 1: The user-based perspective on a collaborative filtering model.

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 Equation (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 Equation (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., Kendall (1980). The model is illustrated as a Bayesian network in Figure 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

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

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 easily be learned using the EM algorithm (Dempster et al, 1977; Kendall, 1980).

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 Equation (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 are usually poor. 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 Taking a dual perspective

As indicated above, most recommender systems are based on a clustering of either users or items. Unfortunately, by only considering one of these two perspectives one may potentially leave out important information, which could otherwise have improved the performance of the system, in particular when data is scarce. This observation is exploited in the following.

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 build 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 Equation (6) we have introduced the constants ϕ_p and ψ_i, which 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 $\epsilon \sim \mathcal{N}\,(0, \theta)$. Note 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., $|\boldsymbol{M}_r| = |\boldsymbol{M}_i|$). By examining the model more closely, the marginal distribution for $\boldsymbol{R}\,(p,i)$ can be written as

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

The main motivation for using the model is how correlations between *arbitrary* ratings are efficiently taken into account when making recommendations. Consider Figure 2, which shows a full BN 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).

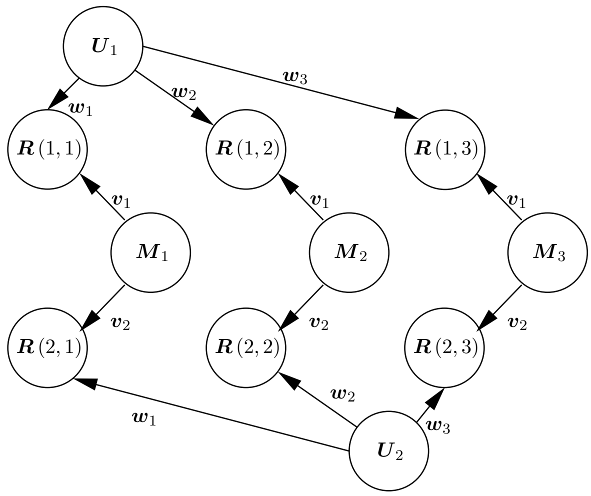

Figure 2: The full statistical model for collaborative filtering; this model has $\#M = 3$ and $\#U = 2$.

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, $\boldsymbol{R}(2,3) \not\perp\!\!\!\perp \boldsymbol{R}(1,2) \,|\{\boldsymbol{R}(1,1), \boldsymbol{R}(2,1)\}$, which clearly exemplifies the global view of the present model.

To summarize, contrary to standard (non-relational) 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).

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). 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:

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 the item that is enjoyed the best *on average*.

Maximin: Choose 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

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

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., Bishop (2006)), we obtain that the joint distribution over the latent variables conditioned on the observed ratings is given by

$$[\boldsymbol{M} \;\; \boldsymbol{U}]^{\mathrm{T}} \big| \, \boldsymbol{r} \sim \mathcal{N}(\boldsymbol{\nu}, \boldsymbol{\Sigma}).$$

Here, the covariance matrix is given by (see also Appendix A)

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

where $\boldsymbol{L}$ is the regression matrix 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

$$\left[\begin{array}{c} \boldsymbol{R}(p_1,i) \\ \boldsymbol{R}(p_2,i) \end{array}\right] \sim \mathcal{N}\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\nu} + \left[\begin{array}{c} \phi_{p_1} + \psi_i \\ \phi_{p_2} + \psi_i \end{array}\right], \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{A} + \theta\boldsymbol{I}\right). \tag{7}$$

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 (Herlocker et al, 1999) with three latent variables for each movie and one latent variable for each user, i.e., ($|\boldsymbol{M}| = 3$ and $|\boldsymbol{U}| = 1$).

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 $\hat{\boldsymbol{m}}_i = \mathbb{E}(\boldsymbol{M}_i|\boldsymbol{r})$ 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).[2] 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:

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

where $\hat{\boldsymbol{Q}}$ is the empirical precision 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.[3] Out of the 10 movies closest to *Star Wars*, 7 are movies that we (the authors) believe are well classified as "similar to Star Wars". *The Princess Bride* and *Raiders of the Lost Ark* are somewhat related in the sense that they are adventure movies, but *Private Parts* does not seem to fit in that well; we see a similar pattern for the movies closets to *Three Colors: Blue*. Considering that there are 1683 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* from mainly action and sci.-fi. movies.

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

[2] In the analyzes below, we only considered movies with at least 50 ratings.

[3] Although not part of the table, we would like to note that *Three Colors: Red* comes in at rank 11 relative to *Three Colors: Blue*.

1.	The Empire Strikes Back	1.	Welcome to the Dollhouse
2.	The Princess Bride	2.	Heavenly Creatures
3.	Star trek II	3.	Three Colors: White
4.	Return of the Jedi	4.	Wings of Desire
5.	Raiders of the Lost Ark	5.	Everyone Says I Love You
6.	Star Trek IV	6.	Muriel's Wedding
7.	Private Parts	7.	Dead Man Walking
8.	Star Trek VI	8.	The Nightmare Before Christmas
9.	Mystery Science Theater 3000	9.	Boogie Nights
10.	Men in Black	10.	To Die For

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

1.	Lost Highway	1.	Star Trek II
2.	Crash	2.	The Empire Strikes Back
3.	White Squall	3.	Return of the Jedi
4.	The First Wives Club	4.	Die Hard: With a Vengeance
5.	Four Rooms	5.	Raiders of the Lost Ark
6.	The Unbearable Lightness of Being	6.	Star Wars
7.	I Know What You Did Last Summer	7.	Independence Day
8.	Angels and Insects	8.	True Lies
9.	Breaking the Waves	9.	Star Trek IV
10.	Jane Eyre	10.	Twister

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

1.	Ace Ventura: Pet Detective	1.	Angels and Insects
2.	A Nightmare on Elm Street	2.	Big Night
3.	Die Hard: With a Vengeance	3.	Breaking the Waves
4.	True Lies	4.	Il Postino
5.	Twister	5.	Three Colors: Blue
6.	Independence Day	6.	The Crying Game
7.	Die Hard 2	7.	Breakfast at Tiffany's
8.	Top Gun	8.	Cold Comfort Farm
9.	Con Air	9.	Harold and Maude
10.	Happy Gilmore	10.	Muriel's Wedding

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 extend the movie appeals to a teenage audience.

1.	The Cook the Thief His Wife and Her Lover	1.	The First Wives Club
2.	Mystery Science Theater 3000: The Movie	2.	White Squall
3.	The City of Lost Children	3.	The Preacher's Wife
4.	Delicatessen	4.	Dirty Dancing
5.	Army of Darkness	5.	The Crucible
6.	Brazil	6.	Jane Eyre
7.	Star Wars	7.	Crash
8.	Star Trek II	8.	Pretty Woman
9.	The Empire Strikes Back	9.	The Mirror Has Two Faces
10.	This Is Spinal Tap	10.	Little Women

Table 4: The 10 movies with lowest and highest values in the second dimension in the latent space. A semantic interpretation might be that this dimension represent to what extend the movie appeals to a male/female audience.

of the three dimensions in the latent space. The results can be seen in Table 3–5. Based on the listed movies, one possible semantic interpretation might be that the first dimension encodes to what extend the movie would appeal to a teenage audience, the second dimension represent whether the movie appeals to a male/female audience, and the third dimension might represent whether the movie is a classic.

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 has 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 6, where each movie's position on the Internet Movie Database's (IMDB's) list of top 250 movies are given as reference.[4] Note that our model also picked out three "Wallace and Gromit" movies as contenders for the top-ten list. These movies are either short-movies or a compilation of such, and do therefore not qualify for the IMDB top 250-list. We have therefore removed them

[4] `http://www.imdb.com`

1.	It's a Wonderful Life	1.	Lost Highway
2.	Raiders of the Lost Ark	2.	Beavis and Butt-head Do America
3.	Sleepless in Seattle	3.	Event Horizon
4.	E.T. the Extra-Terrestrial	4.	Four Rooms
5.	The Empire Strikes Back	5.	Natural Born Killers
6.	Singin' in the Rain	6.	The Celluloid Closet
7.	Dave	7.	Boogie Nights
8.	The Firm	8.	Koyaanisqatsi
9.	Mary Poppins	9.	Crash
10.	Dirty Dancing	10.	The Ice Storm

Table 5: The 10 movies with lowest and highest values in the third dimension in the latent space. Semantically, this dimension may be interpreted as to what extend the movie is consider a classic.

from the results in Table 6 for ease of comparison. Note also that our dataset only contains movies released in 1998 or before, which explains why e.g. "The Dark Knight" (IMDB 8) and the "The Lord of the Rings" series (IMDB 13, 21, and 34) are not on our list.

1.	The Shawshank Redemption	IMDB: 1
2.	Schindler's List	IMDB: 6
3.	Star Wars	IMDB: 12
4.	Casablanca	IMDB: 11
5.	The Usual Suspects	IMDB: 22
6.	Rear Window	IMDB: 16
7.	Raiders of the Lost Ark	IMDB: 18
8.	The Silence of the Lambs	IMDB: 24
9.	One Flew Over the Cuckoo's Nest	IMDB: 8
10.	12 Angry Men	IMDB: 7

Table 6: The 10 "best" movies, i.e., the movies with the highest ψ_i value.

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 (Dempster et al, 1977). 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.

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

$$\begin{aligned}
\hat{\theta} &\leftarrow \frac{1}{d}\sum_{p=1}^{\#U}\sum_{i\in\mathcal{I}(p)} \mathbb{E}[(\boldsymbol{r}\,(p,i) - (\boldsymbol{v}_p^{\mathrm{T}}\boldsymbol{M}_i + \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{U}_p + \phi_p + \psi_i))^2];\\
\hat{\boldsymbol{v}}_p &\leftarrow \left[\sum_{i\in\mathcal{I}(p)} \mathbb{E}(\boldsymbol{M}_i\boldsymbol{M}_i^{\mathrm{T}})\right]^{-1}\left[\sum_{i\in\mathcal{I}(p)} (\mathbb{E}(\boldsymbol{M}_i)\boldsymbol{r}\,(p,i) - \mathbb{E}(\boldsymbol{M}_i\boldsymbol{U}_p^{\mathrm{T}})\boldsymbol{w}_i - \mathbb{E}(\boldsymbol{M}_i)(\phi_p+\psi_i))\right];\\
\hat{\phi}_p &\leftarrow \frac{1}{|\mathcal{I}(p)|}\sum_{i\in\mathcal{I}(p)} \big(\boldsymbol{r}\,(p,i) - (\boldsymbol{v}_p^{\mathrm{T}}\mathbb{E}(\boldsymbol{M}_i) + \boldsymbol{w}_i^{\mathrm{T}}\mathbb{E}(\boldsymbol{U}_p) + \psi_i)\big);\\
\hat{\boldsymbol{w}}_i &\leftarrow \left[\sum_{p\in\mathcal{P}(i)} \mathbb{E}(\boldsymbol{U}_p\boldsymbol{U}_p^{\mathrm{T}})\right]^{-1}\left[\sum_{p\in\mathcal{P}(i)} \mathbb{E}(\boldsymbol{U}_p)\boldsymbol{r}\,(p,i) - \mathbb{E}(\boldsymbol{U}_p\boldsymbol{M}_i^{\mathrm{T}})\boldsymbol{v}_p - \mathbb{E}(\boldsymbol{U}_p)(\phi_p+\psi_i)\right];\\
\hat{\psi}_i &\leftarrow \frac{1}{|\mathcal{P}(i)|}\sum_{p\in\mathcal{P}(i)} (\boldsymbol{r}\,(p,i) - (\boldsymbol{v}_p^{\mathrm{T}}\mathbb{E}(\boldsymbol{M}_i) + \boldsymbol{w}_i^{\mathrm{T}}\mathbb{E}(\boldsymbol{U}_p) + \phi_p)).
\end{aligned} \tag{8}$$

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).[5] 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 (Kohavi and John, 1997).

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 Equation 8) 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 (Marlin, 2004)), 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 numerical 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 Equation 8 is also the solution that minimizes the expected least squared error.[6] 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 (Hastie et al, 2001):

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

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,

[5] 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 unfortunately outside the scope of the present paper.

[6] 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 $\boldsymbol{X}_{i,:} = \boldsymbol{M}_i^{\mathrm{T}}$.

by letting $\boldsymbol{v}_p \sim \mathcal{N}(\mathbf{0}, \tau \boldsymbol{I})$, 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 (Bertie and Cran, 1985) we use the estimators for ϕ_p and ψ_i that were found *without* regularization.

6 Results

In this section we investigate the predictive performance of the proposed system. Specifically, we evaluate the system using the MovieLens dataset, which consists of 100.000 ratings between 1 and 5 from 943 users on 1682 movies (Herlocker et al, 1999). For the actual testing we performed five-fold cross validation using the folds supplied with the dataset.

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 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 Figure 3 illustrates the procedure; the figure shows the MAE as a function of the prior precision for the regression parameters. The plots are generated for different combinations of latent variables s.t. the plot at position $(|\boldsymbol{U}|, |\boldsymbol{M}|)$ correspond to a model with $|\boldsymbol{U}|$ latent user variables and $|\boldsymbol{M}|$ 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 $|\boldsymbol{U}| = 1$, $|\boldsymbol{M}| = 1$, 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 (Kohavi and John, 1997). Next, we considered non-visited neighboring candidate models that can be reached by either increasing $|\boldsymbol{U}|$ or $|\boldsymbol{M}|$. This gives the candidate structures $(|\boldsymbol{U}| = 1, |\boldsymbol{M}| = 2)$ and $(|\boldsymbol{U}| = 2, |\boldsymbol{M}| = 1)$; both evaluated as above. The best of these two candidate models is chosen (in this case, $(|\boldsymbol{U}| = 1, |\boldsymbol{M}| = 2)$ 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 $(|\boldsymbol{U}| = 1, |\boldsymbol{M}| = 2)$ 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, Figure 3 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 Figure 4. 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 $|\boldsymbol{U}| = 1$ and $|\boldsymbol{M}| = 2$ with an MAE of 0.685 (calculated using a prior precision of 25 for the regression parameters).

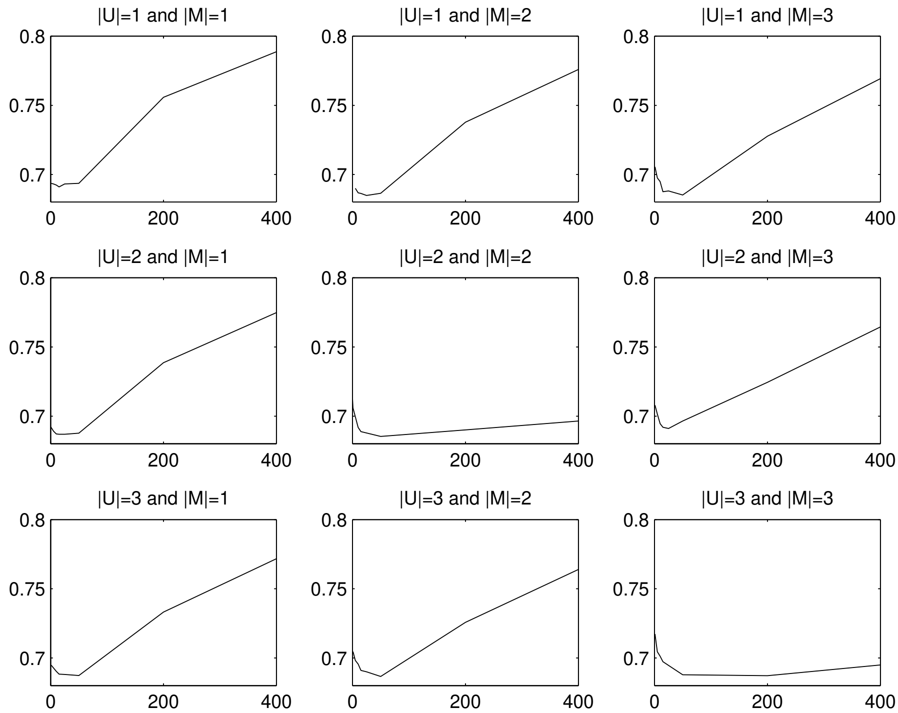

Figure 3: The figure shows the MAE as a function of the prior precision for the regression vectors. Each plot corresponds to a certain configuration of the number of latent variables.

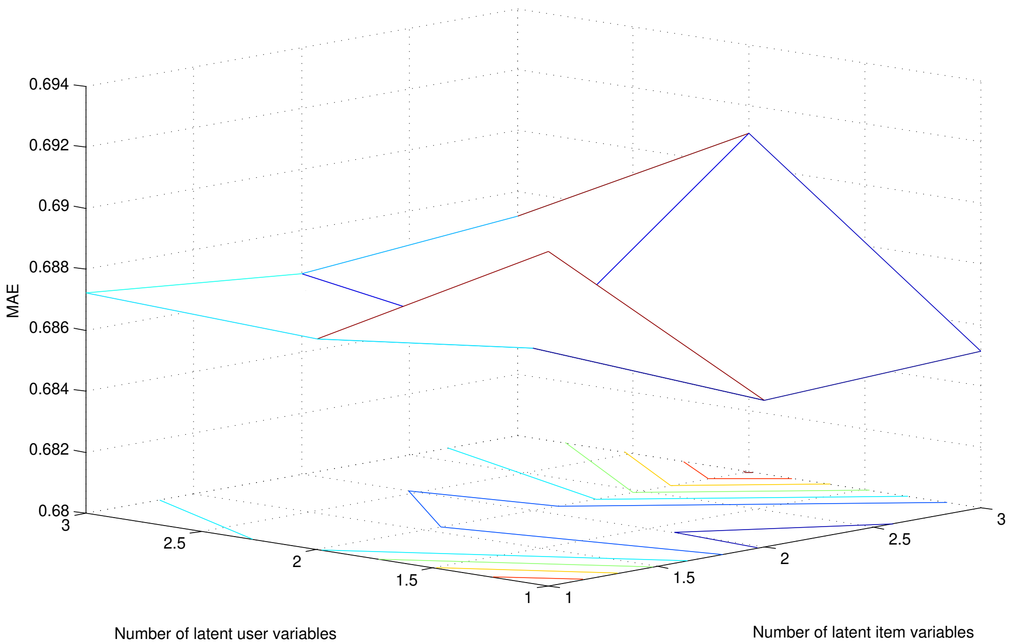

Figure 4: The figure shows the MAE as a function of the number of latent variables. A minimum (0.685) is found at $|\boldsymbol{U}| = 1$ and $|\boldsymbol{M}| = 2$.

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:

SVD(q, λ) which performs a singular value decomposition in q dimensions. λ is the regularization weight (see Equation (3)). For each setting of λ we ran experiments with values for q ranging from one to ten; only the best result is reported here. Two options were considered for λ: $\lambda = 0$ resulting in a non-regularized model, and $\lambda = 0.01$ (as done by Salakhutdinov et al (2007)).

FA-U(q) corresponds to the user-centered factor analysis model, where q denotes the number of latent variables (Kendall, 1980), see Equation (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 $[1, 10]$.

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

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 (Herlocker et al, 1999).

Euclidean(k) is the k-nearest neighbors algorithm, where the distance is calculated using Euclidean norm (Marlin, 2004).

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 (Jin et al, 2006).

The results are shown in Table 7, 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. 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 $|\boldsymbol{U}| = 1$ is 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 is related with the results of the proposed model, where $|\boldsymbol{M}| = 2$ is selected.

It is difficult to find results in the scientific literature that is 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 (Herlocker et al, 1999; Li and Kim, 2003; Mobasher et al, 2003; Kim and Yum, 2005; Chen and Yin, 2006).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean
Pearson(all)	0.72250	0.71330	0.70615	0.70630	0.71295	0.71224
Euclidean(all)	0.73055	0.71950	0.71805	0.72095	0.72105	0.72202
Pearson(10)	0.73665	0.72965	0.72300	0.72700	0.73105	0.72947
Euclid(10)	0.75315	0.73540	0.74095	0.74475	0.74875	0.74460
Pearson(25)	0.71850	0.70705	0.70645	0.69975	0.70815	0.70798
Euclidean(25)	0.73060	0.71915	0.72370	0.72125	0.72715	0.72437
Pearson(50)	0.71570	0.70489	0.71330	0.71070	0.71015	0.71095
Euclidean(50)	0.73730	0.73139	0.73150	0.73350	0.73050	0.73284
Pearson(75)	0.71395	0.70015	0.70265	0.69820	0.70430	0.70385
Euclidean(75)	0.72595	0.71465	0.71565	0.71600	0.72045	0.71854
DM	0.7583	0.7418	0.7284	0.7509	0.7497	0,74582
FA/U($q = 1$)	0.73235	0.727986	0.7257	0.72785	0.7208	0.7269
FA/I($q = 1$)	0.8048	0.80514	0.80385	0.8000	0.8067	0.804098
SVD($q = 5$, $\lambda = 0$)	0.70045	0.69085	0.6971	0.6918	0.6992	0.69588
SVD($q = 4$, $\lambda = 0.01$)	0.6987	0.68755	0.68985	0.68925	0.6926	0.69159
CF($\|\boldsymbol{U}\| = 1$, $\|\boldsymbol{M}\| = 2$, $\tau = 1/25$)	0.68365	0.686934	0.6846	0.68605	0.68275	0.68479

Table 7: 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.

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, we have shown through examples 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.

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. This also motivates the development of approximate inference algorithms that will allow the framework to be applied to even larger domains.

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

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

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:

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

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:

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

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

Bertie AJ, Cran GW (1985) Estimation of the constant term when using ridge regression. International Journal of Mathematical Education in Science and Technology 16:63 – 65

Bishop CM (2006) Pattern Recognition and Machine Learning, 1st edn. Information Science and Statistics, Springer-Verlag

Blei DM, Ng AY, Jordan MI, Lafferty J (2003) Latent Dirichlet allocation. Journal of Machine Learning Research 3:2003

Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, pp 43–52

Chen J, Yin J (2006) Recommendation based on influence sets. In: Proceedings of the Workshop on Web Mining and Web Usage Analysis

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B 39:1–38

Domingos P, Pazzani M (1997) On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29(2–3):103–130

Duda RO, Hart PE (1973) Pattern Classification and Scene Analysis. John Wiley & Sons, New York

Hastie T, Tibshirani R, Friedman J (2001) Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer-Verlag, New York

Heckerman D, Chickering D, Meek C, Rounthwaite R, Kadie C (2000) Dependency networks for inference, collaborative filtering and data visualization. Journal of Machine Learning Research 1:49–75

Herlocker J, Konstan J, Borchers A, Riedl J (1999) An algorithmic framework for performing collaborative filtering. In: Proceedings of the ACM 1999 Conference on Research and Development in Information Retrieval, pp 230–237

Hofmann T (2001) Learning what people (don't) want. In: Proceedings of the Twelfth European Conference on Machine Learning, Springer-Verlag, London, UK, pp 214–225

Hofmann T (2004) Latent semantic models for collaborative filtering. ACM Transactions on Information Systems 22(1):89–115, DOI http://doi.acm.org/10.1145/963770.963774

Jan TH, Puzicha (1999) Latent class models for collaborative filtering. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, CA, USA, pp 688–693

Jensen FV, Nielsen TD (2007) Bayesian Networks and Decision Graphs. Springer-Verlag, Berlin, Germany

Jin R, Si L, Zhai C (2006) A study of mixture models for collaborative filtering. Information Retrieval 9(3):357–382, DOI http://dx.doi.org/10.1007/s10791-006-4651-1

Kendall M (1980) Multivariate Analysis, 2nd edn. Charles Griffin & Co., London, UK

Kim D, Yum BJ (2005) Collaborative filtering based on iterative principal component analysis. Expert Systems with Applications 28(4):823 – 830, DOI DOI:10.1016/j.eswa.2004.12.037

Kohavi R, John GH (1997) Wrappers for feature subset selection. Artificial Intelligence 97(1–2):273–324

Langseth H, Nielsen TD (2005) Latent classification models. Machine Learning 59(3):237–265

Li Q, Kim BM (2003) Clustering approach for hybrid recommender system. In: WI '03: Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence, IEEE Computer Society, Washington, DC, USA, pp 33–38

Marlin B (2003) Modeling user rating profiles for collaborative filtering. In: Advances in Neural Information Processing Systems 15, The MIT Press, pp 627–634

Marlin B (2004) Collaborative filtering: A machine learning perspective. Master of Science Thesis, Graduate Department of Computer Science, University of Toronto

Mobasher B, Jin X, Zhou Y (2003) Semantically enhanced collaborative filtering on the web. In: Web Mining: From Web to Semantic Web, First European Web Mining Forum, EMWF 2003, no. 3209 in Lecture Notes in Computer Science, pp 57–76

Pearl J (1988) Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, San Mateo, CA.

Pennock DM, Horvitz E, Lawrence S, Giles CL (2000) Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, pp 473–480

Popescul A, Popescul R, Ungar LH, Pennock DM, Lawrence S (2001) Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, pp 437–444

Salakhutdinov R, Mnih A, Hinton G (2007) Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the Twenty-fourth International Conference on Machine Learning, vol 24, pp 791–798

Savia E, Puolamäki K, Sinkkonen J, Kaski S (2005) Two-way latent grouping model for user preference prediction. In: Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence, pp 518–525

Si L, Jin R (2003) Flexible mixture model for collaborative filtering. In: Proceedings of the Twentieth International Conference on Machine Learning, National Conference on Artificial Intelligence, pp 704–711

Truyen TT, Phung DQ, Venkatesh S (2009) Ordinal Boltzmann machines for collaborative filtering. In: Proceedings of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence

Wang J, de Vries AP, Reinders MJT (2006) Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, pp 501–508, DOI http://doi.acm.org/10.1145/1148170.1148257

Xu Z, Tresp V, Yu K, Kriegel HP (2006) Infinite hidden relational models. In: Proceedings of the Twenty-second Conference on Uncertainty in Artificial Intelligence