

Chapter 11

Network-Based Models for Social Recommender Systems



Antonia Godoy-Lorite, Roger Guimerà, and Marta Sales-Pardo

Abstract With the overwhelming online products available in recent years, there is an increasing need to filter and deliver relevant personalized advice for users. Recommender systems solve this problem by modelling and predicting individual preferences for a great variety of items such as movies, books or research articles. In this chapter, we explore rigorous network-based models that outperform leading approaches for recommendation. The network models we consider are based on the explicit assumption that there are groups of individuals and of items, and that the preferences of an individual for an item are determined only by their group memberships. The accurate prediction of individual user preferences over items can be accomplished by different methodologies, such as Monte Carlo sampling or Expectation-Maximization methods, the latter resulting in a scalable algorithm which is suitable for large datasets.

Keywords Human preferences · Prediction · Matrix factorization · Modelling · Social marketing

A. Godoy-Lorite (✉)
Department of Mathematics, Imperial College London, London, UK
e-mail: a.godoy-lorite@imperial.ac.uk

R. Guimerà
Institució Catalana de Recerca i Estudis Avançats (ICREA), Barcelona, Catalonia, Spain
Departament d'Enginyeria, Química, Universitat Rovira i Virgili, Tarragona, Catalonia, Spain
e-mail: roger.guimera@urv.cat

M. Sales-Pardo
Departament d'Enginyeria Química, Universitat Rovira i Virgili, Tarragona, Spain
e-mail: marta.sales@urv.cat

11.1 Introduction

The internet has changed the way business is done and how products are advertised, sold and distributed [1, 2]. Now we are a click away from an ever increasing array of products [3]. However, the availability of so many choices puts a stress on the customer who often has no time to browse over endless online product catalogues. As an illustration, Netflix has available around 5000 movies only in the United States, iTunes more than 37 million songs and Amazon up to 32 million books in different formats; if we were to spend 0.5 seconds per item, it would take us approximately 40 minutes to browse the whole catalogue in Netflix and over 200 full days to go through the whole catalogue of songs and books in iTunes and Amazon. The platforms that have best adapted to this situation are those that efficiently recommend items that fit personal preferences.

Recommender systems are algorithms precisely designed to predict user's preferences over a variable amount of items. A popular event that boosted research in recommender systems was the Netflix contest (2006–2009) [4–6]. Netflix sponsored a competition to improve the accuracy of their recommendation algorithm at the time, offering \$1,000,000 to the best performing team. This competition captured the attention of researchers on the topic and improved significantly the state-of-the-art algorithms and even resulted in the creation of companies (for instance, Gravity R&D or 4-Tell Inc. [7]) that played a major role in boosting e-commerce. The increase in the volume of online business coupled to the availability of data on online purchases of products by users has in recent years enhanced the interest in recommender systems, both in the private and academic sectors.

Currently, the main strategies for making social recommendations are content-based approaches, collaborative filtering and hybrid approaches [8]. Content-based approaches use available metadata on users or items such as demographics, overall top selling items, past buying habit of users or item reviews to guess user preferences. On the other hand, collaborative filtering (CF) methods are based on the plausible expectation that similar users relate to similar objects in a similar manner, i.e., they purchase similar items and rate the same item similarly. Hybrid methods aim at combining both approaches.

Importantly, CF approaches are in general more accurate at predicting user preferences than content-based approaches. Typically datasets available for recommendation are *sparse*—most users rate just a few items (<10 items) and most of the items have been rated only by a few users, which makes it hard for content-based methods to make good predictions. In contrast, CF algorithms have successfully addressed this problem by exploiting known preferences of like-minded users to provide item recommendations or predictions.

A major problem recommender systems face is the need to provide recommendations in a reasonably short amount of time. Taking into account that available datasets comprise millions of user-item ratings (and that is a small fraction of the data in a real industrial setting), the scalability of the algorithm with the number of observations is critical. Specifically, if the run-time of an algorithm scales linearly

in the number of observed ratings R , the time needed to obtain a recommendation if $R = 10,000$ is $\propto 10^5$, whereas if an algorithm scales quadratically with R , then the time needed to obtain a recommendation for $R = 10,000$ ratings will be $\propto 10^{10}$. As a good rule of thumb, linear (or sub-linear) CF algorithms will be good candidates to deal with the large and sparse datasets in an industrial set-up.

In this chapter, we focus on collaborative filtering models suitable for large sparse datasets. Specifically, we show how block model approaches to model a network of user-item ratings is superior to state-of-the-art approaches such as the Item-Item and Matrix Factorization approaches [9–12].

This chapter is organized as follows. In Sect. 11.2 we introduce the network framework for the recommendation problem, fundamental concepts of inference and the use of Stochastic Block Models to make inference on network data. In Sect. 11.3 we present the network-based approaches for recommender systems, the bipartite Stochastic Block Model recommender (SBM) and the Mixed-Membership Stochastic Block Model recommender (MMSBM). In Sect. 11.4 we give an overview of some of the most successful collaborative filtering approaches, one being the Item-Item model and two Matrix Factorization approaches: the “classical” Matrix Factorization (MF) and the Mixed-Membership Matrix Factorization (MMMMF), which we use as benchmark algorithms. In Sect. 11.5 we analyse and compare the predictive power of those algorithms and provide a practical guide to run the network-based algorithms with real datasets. To conclude, in Sect. 11.6 we provide overall discussion of the chapter.

11.2 Network Approach to Recommender Systems

Formally, the recommendation problem is the following. We have an observation R^\odot consisting of a collection of ratings R_{ij} of users (i) to items (j) (e.g., ratings of users to movies or books). Ratings are on a fixed discrete scale S , so that each observed rating r_{ij} takes a value within this scale (e.g., in a 5 point scale $S = \{1, 2, 3, 4, 5\}$). This observation is sparse so that out of a group of N users and M items we only observe a small fraction of the $N \times M$ possible ratings. The goal is then to predict the values of a set of query/unobserved ratings R^\top . In the recommender systems literature, the observation R^\odot is called the training set, while the query set R^\top is called the test set. This is because recommendation algorithms use the training dataset to train the algorithm and obtain the model parameters and the test set to assess the accuracy of the trained algorithm at making predictions for unobserved ratings. Note that being able to make accurate predictions on unobserved ratings is the first and necessary step towards being able to make suggestions of new items to users based on the rating predictions over those items.

Formally, this problem can be mapped into a problem of predicting unobserved edge values in a network. Specifically, since ratings occur between two different types of nodes (users and items), we can represent R^\odot as a bipartite network (see Fig. 11.2a). In this network we have an edge connecting each user with all the

items she has provided a rating for in the observation. Importantly, within this representation edges can take a value within the scale of ratings S . The problem of predicting ratings within the unobserved query set R^T then becomes that of predicting values of unobserved edges within this network.

Here, we focus on estimating the probability that a specific unobserved edge r_{ij} takes value r given the observed data R^\ominus , which formally is expressed as $p(r_{ij} = r | R^\ominus)$. To do that we use a Bayesian approach to perform inference on network data. In what follows, we introduce the basic concepts of Bayesian inference and the Stochastic Block Model, a general class of generative models suitable for network data.

11.2.1 Inference on Complex Networks Based on Stochastic Block Models

Let us assume our observed data is R^\ominus and we want to know the probability that a certain variable X (for instance, a rating) takes values x conditioned on the observed data, that is $p(X = x | R^\ominus)$. If we consider an ensemble of generative models \mathcal{M} for our observed data, we can express $p(X = x | R^\ominus)$ as

$$P(X = x | R^\ominus) = \int_{\mathcal{M}} p(X = x | M) p(\mathcal{M} | R^\ominus) dM, \quad (11.1)$$

where $p(X = x | M)$ is the probability of variable X being equal to x given model M (X is, for example, the rating that user u gives item i), and $p(M | R^\ominus)$ is the plausibility of model M given the observation R^\ominus . Using Bayes' theorem we can rewrite Eq. (11.1) as,

$$P(X | R^\ominus) = \frac{\int_{\mathcal{M}} p(X | M) p(R^\ominus | M) p(M) dM}{p(R^\ominus)}, \quad (11.2)$$

where $p(R^\ominus | M)$ is the probability that model M gives rise to the observed data R^\ominus , also called likelihood, and $p(M)$ is the prior probability that model M is the correct one, also called the prior. Importantly, the accuracy of the predictions depends strongly on the ability of some of the models in the family of models in \mathcal{M} to describe the observed data.

In our case we consider the family of Stochastic Block Models (SBM) [13–15] as generative models. SBMs are based on the simple assumption that there are groups of nodes and that nodes within a group have similar connectivity patterns. Within this class of models, the probability of two nodes being connected only depends

on the groups to which the nodes belong. Formally, a SBM $\mathcal{M} = (P, \mathbf{Q})$ is then completely determined by the partition P of nodes into groups and the matrix \mathbf{Q} of connection probabilities between pairs of nodes belonging to pairs of groups, so that Eq. (11.2) can be rewritten as

$$P_{\text{SBM}}(X = x | R^\odot) = \frac{\sum_{P \in \mathcal{P}} \int_{[0,1]^G} p(X = x | P, \mathbf{Q}) p_{\text{SBM}}(R^\odot | P, \mathbf{Q}) p(P, \mathbf{Q}) d\mathbf{Q}}{p(R^\odot)}, \tag{11.3}$$

where \mathcal{P} is the space of all possible partitions of nodes into groups and G is the total number of pairs of groups of nodes.

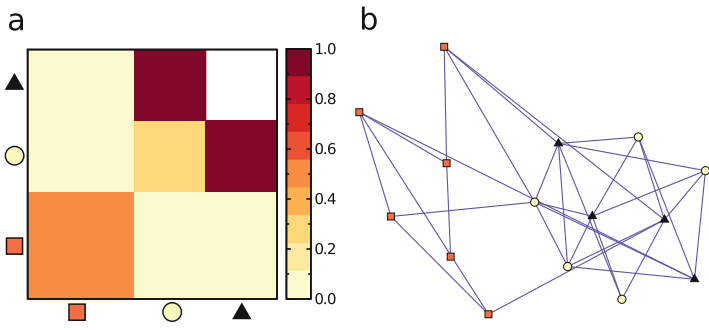


Fig. 11.1 Stochastic Block Models: A stochastic block model is fully specified by a partition of nodes into groups P and a connection probability matrix \mathbf{Q} . Each element $Q_{\alpha\beta}$ in the \mathbf{Q} matrix represents the probability that a node in group α connects to a node in group β . (a) An example of a \mathbf{Q} matrix of connection probabilities. We consider three groups of nodes comprising 4 (triangles), 5 (circles) and 6 (squares) nodes. We colour matrix elements according to their value following the colour bar on the right hand side. (b) A realization of the model in panel (a)

SBMs are suitable models to describe complex networks because they are versatile enough to capture the large variety of connectivity patterns observed in real networks (see Fig. 11.1 for an illustration). For instance, many real-world networks have been found to have a modular or assortative structure in which nodes within the same group (also called module or community) are more likely to be connected to nodes within the same group than to nodes in other groups [16–19]. A SBM with $Q_{\alpha\alpha} \gg Q_{\alpha\beta} \forall \alpha, \beta$ would generate networks with such structure. Interestingly, SBMs can also depict other connectivity patterns such as disassortative patterns in which nodes are more likely to connect to nodes in other groups or patterns that define distinct topological roles such as hubs and peripheral nodes in core-periphery structures [18, 20, 21]. Importantly, this family of models can be extended to directed [22], and weighted networks [23, 24].

11.3 Modelling Ratings Using Stochastic Block Models

In this section we will consider two network-based models for recommender systems: the simple bipartite Stochastic Block Model (SBM) [14] and the Mixed-Membership Bipartite Stochastic Block Model (MMSBM) [15]. In both models, there are different groups. The difference between these models is that while in the bipartite SBM each user and item belong solely to one group, in the MMSBM users (and items) have a certain probability of belonging to each group of users (items). Importantly, this fact allows us to describe the network of ratings using fewer groups of users and items and to implement more efficient inference algorithms.

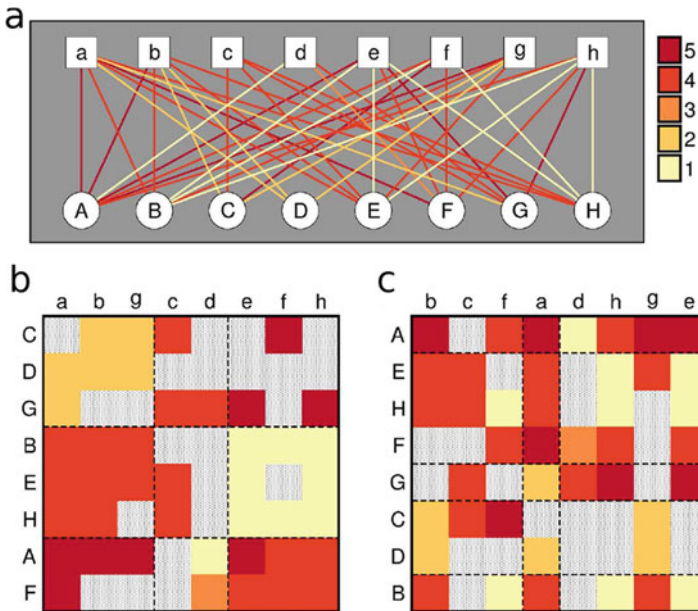


Fig. 11.2 Bipartite SBM for recommendation: (a) Eight users labelled A–H rate movies, labelled a–h, as indicated by the colours of the links. (b–c) Matrix representation of the ratings; grey elements represent unobserved ratings. Different partitions of the nodes into groups (indicated by the dashed lines) provide different explanations for the observed ratings. The partition in (b) has a high explanatory power because ratings in each pair of user-item groups are very homogeneous. For example, it seems plausible that user C would rate item a with a 2, given that all users in the same group as C rate 2 all items in the same group as a. Conversely, the partition in c has very little explanatory power. The predictions of partition (b) contribute much more than those of partition (c) to the inference of unobserved ratings. Reprinted from Guimerà R. et al. *Predicting Human Preferences Using the Block Structure of Complex Social Networks*. PLOS ONE 7(9):e44620, under the Creative Commons Attribution (CC BY) license at <https://creativecommons.org/licenses/by/4.0/>

11.3.1 Predictions Based on the Bipartite SBM Recommender System

Our inference problem is to estimate the probability $p(r_{ui} = r | R^\odot)$ that the unobserved rating of item i by user u is $r_{ui} = r$, given the observation R^\odot . Hence, by setting the observable x in Eq. (11.2) to $X = r_{ui}$ we obtain,

$$p(r_{ui} = r | R^\odot) = \frac{\int_M p(r_{ui} = r | M) P(R^\odot | M) p(M) dM}{\int_{\mathcal{M}'} p(R^\odot | M') p(M') dM'} . \quad (11.4)$$

where $p(r_{ui} = r | M)$ is the probability that $r_{ui} = r$ if the ratings were actually generated using model M , and $p(R^\odot | M)$ is the probability of model M generating the observed ratings R^\odot or likelihood.

As previously mentioned, we will use the family of Stochastic Block Models \mathcal{M}_{SBM} to describe the observed ratings [13, 21, 25]. In the bipartite SBM users and items are partitioned into two different and independent sets of groups. Therefore, \mathbf{Q} is a $g_u \times g_i$ rectangular matrix, with g_u and g_i being the number of user and item groups, respectively. Additionally, because in our network (ratings) edges can take $|S|$ different values, we have one such matrix for each value of r . Therefore, the probability that the rating of user u to item i is equal to $r_{ui} = r$ depends exclusively on the groups σ_u and σ_i to which user u and item i , respectively, belong, so that

$$p(r_{ui} = r) = Q_{\sigma_u \sigma_i}^r . \quad (11.5)$$

Because in the recommender system the possible rating values for a given edge are exclusive, we have the following constraint $\sum_r Q_{\sigma_u \sigma_i}^r = 1$ for each (user, item) pair.

Note that in the bipartite SBM, ratings are considered as independent categories without assuming that the distance between ratings is linear (that is, that $r = 3$ is as far from $r = 4$, as $r = 4$ is from $r = 5$). This poses an advantage over other approaches which assume linearity in the distances between ratings, since users have been found not to perceive equal differences between adjacent ratings (i.e., $r = 4$ and $r = 5$ might be perceived as closer in rating space than $r = 3$ and $r = 4$ [26]).

Assuming a flat prior over models $p(M) = \text{const.}$, the integral in Eq. (11.4) over all possible values of $Q_{\alpha\beta}$ can be carried out analytically, so that we obtain

$$p_{SBM}(r_{ui} = r | R^\odot) = \frac{1}{\tilde{Z}} \sum_{P_U \in \mathcal{P}_U, P_I \in \mathcal{P}_I} \left(\frac{n_{\sigma_u \sigma_i}^r + 1}{n_{\sigma_u \sigma_i} + |S|} \right) e^{-\mathcal{H}(P_U, P_I)} \quad (11.6)$$

where the sum is over all possible partitions of users and items into groups, $n_{\sigma_u \sigma_i}^r$ is the number of ratings with value r observed from users in group σ_u to items in group σ_i , and $n_{\sigma_u \sigma_i}$ is the total number of observed ratings from users in group σ_u to items in group σ_i . The $\mathcal{H}(P_U, P_I)$ is understood as an energy function or Hamiltonian which weighs the contribution of each partition of users and items (P_U, P_I) to the sum over all pairs of partitions,

$$\mathcal{H}(P_U, P_I) = \sum_{\alpha, \beta} \left[\ln((n_{\alpha\beta} + |S| - 1)!) - \sum_{s=1}^{|S|} \ln\left(\binom{n_{\alpha\beta}}{s}\right)! \right] \quad (11.7)$$

and $\mathcal{Z} = \sum_{(P_U, P_I)} e^{-\mathcal{H}(P_U, P_I)}$ is called the partition function. In order to estimate the sum over all partitions, [14] estimated $p_{SBM}(r_{ui} = r | R^{\circ})$ using Metropolis-Hastings sampling [13, 27]. Note that within this approach, no prior assumptions are made on the grouping of the users and items, or in the desired shape of the connection probability matrix, so that the algorithm itself samples/selects those SBMs which provide the best description of the data.

An advantage of this approach is that we obtain the whole distribution for each rating $P_{SBM}(r_{ui} = r | R^{\circ})$. Therefore, one can choose how to make predictions: using the most likely rating, the mean or the median, among others. In [14], the authors chose to select the most likely rating

$$r_{ui} = \arg \max_r \{p_{SBM}(r_{ui} = r | R^{\circ})\}. \quad (11.8)$$

This probabilistic prediction is in contrast to most recommender systems like matrix factorization and Item-Item algorithms, where the prediction is expressed as a single real number.

The bipartite SBM recommender we just described has two main advantages: (1) it is based on plausible hypotheses about how individuals' preferences arise, and (2) it is mathematically rigorous since it is the result of the full Bayesian probabilistic treatment of the model. However, the correct probabilistic treatment of the model comes at the cost of producing a slow algorithm. The approach above relies on Markov chain Monte Carlo sampling over partitions to make rating predictions, therefore, its computational time does not scale well with the size of the dataset (see Fig. 11.4b). This fact makes it impractical for datasets with millions of ratings [14].

11.3.2 Predictions Based on Mixed-Membership Stochastic Block Model

In this section, the Mixed-Membership Bipartite Stochastic Block Model (MMSBM) approach for recommendation is considered. As previously mentioned, mixed membership models allow nodes to belong to all possible (latent) groups with a finite probability [28, 29]. In our case, we consider a bipartite MMSBM in which we have latent groups for it assumes that each node in the bipartite graph of users and items belongs to a mixture of groups.

In the recommendation problem our goal is to estimate the probability $p(r_{ui} = r | R^{\circ})$. In order to do so, we need to compute the likelihood of the observed data given the model parameters. To that end, we define the model parameters as follows.

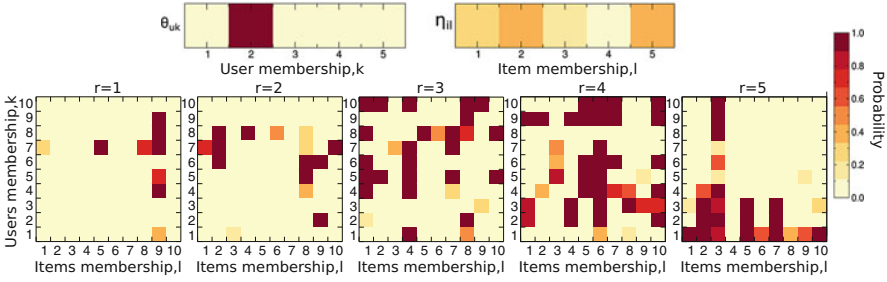


Fig. 11.3 Mixed-Membership Stochastic Block Model: We illustrate the parameters of a bipartite MMSBM for an equal number of latent groups of users and items $K = L = 5$ obtained for the MovieLens 100K dataset. In the top row, we show examples of mixed-membership vectors θ and η for user u and item i , respectively. Each vector component θ_{uk} (η_{il}) is the probability that user u (item i) belongs to group k (group l). Probabilities are shown in colours following the colour bar on the right hand side. In this example, user u has a higher probability to belong to group $k = 2$, while item i has similar probabilities to belong to any group. In the bottom row, we show the inferred values for the probability matrices \mathbf{Q} . From left to right, the five matrices correspond to the ratings $r = 1, 2, 3, 4, 5$. For each one of these matrices, the rows and columns correspond to user and item groups, respectively. Each matrix element is the probability $Q^r_{k\ell}$ that a user in group k gives a rating r to an item in group ℓ . Notice that there is no ordering of the probability matrices that would make them diagonal

In the bipartite MMSBM, we consider that there are K groups of users and L groups of items. For each pair of user-item groups k, ℓ , there is a probability $Q^r_{k\ell} \forall r \in S$ that users in group k give rating r to items in group ℓ . Note that because in R^0 each user-item edge has only one rating r the probability matrices $Q^r_{k\ell}$ are normalized

$$\forall k, \ell : \sum_{r \in S} Q^r_{k\ell} = 1. \tag{11.9}$$

To model mixed group memberships, each user u has a vector $\theta_u \in R^K$, where θ_{uk} denotes the extent to which user u belongs to group k . Similarly, each item i has a vector $\eta_i \in R^L$ (see Fig. 11.3). These vectors are normalized as,

$$\sum_k \theta_{uk} = 1, \tag{11.10}$$

$$\sum_\ell \eta_{i\ell} = 1. \tag{11.11}$$

Given the membership vectors θ_u and η_i , and the probability matrices $Q^r_{k\ell}$, the probability distribution of each rating r_{ui} is a convex combination,

$$p(r_{ui} = r) = \sum_{k, \ell} \theta_{uk} \eta_{i\ell} Q^r_{k\ell}. \tag{11.12}$$

Abbreviating all these parameters as θ, η, \mathbf{Q} , the likelihood of the observed ratings is thus

$$P(R^\circ | \theta, \eta, \mathbf{Q}) = \prod_{(u,i) \in R^\circ} \sum_{k,\ell} \theta_{uk} \eta_{i\ell} Q_{k\ell}^{r_{ui}}. \quad (11.13)$$

In order to perform a full Bayesian approach as for the simple bipartite SBM, we would have to compute the integral in Eq. (11.2) to obtain $p(r_{ui} = r | R^\circ)$. However, this is unfeasible for the current model. Therefore, in order to estimate $p(r_{ui} = r | R^\circ)$, we make a steepest descent approximation and evaluate the integral by considering the model parameters $\hat{\theta}, \hat{\eta}$ and $\hat{\mathbf{Q}}$ that maximize the likelihood in Eq. (11.13).

Note that while this approximation should in principle not perform as well as considering all possible model parameters, our results show that the maximum likelihood prediction for the bipartite MMSBM produces as accurate predictions as the full probabilistic treatment of the simple bipartite SBM (Fig. 11.5). Our results suggest that the introduction of mixed-membership vectors seems to already provide enough flexibility to the model to capture all the patterns covered by the model averaging in the simple bipartite SBM approach. The maximum likelihood parameters $\hat{\theta}, \hat{\eta}, \hat{\mathbf{Q}}$ are inferred using an efficient expectation-maximization algorithm (EM). We start with a standard variational trick that changes the log of a sum into a sum of logs, writing

$$\begin{aligned} \log P(R^\circ | \theta, \eta, \mathbf{Q}) &= \sum_{(u,i) \in R^\circ} \log \sum_{k\ell} \theta_{uk} \eta_{i\ell} Q_{k\ell}^{r_{ui}} \\ &= \sum_{(u,i) \in R^\circ} \log \sum_{k\ell} \omega_{ui}(k, \ell) \frac{\theta_{uk} \eta_{i\ell} Q_{k\ell}^{r_{ui}}}{\omega_{ui}(k, \ell)} \\ &\geq \sum_{(u,i) \in R^\circ} \sum_{k\ell} \omega_{ui}(k, \ell) \log \frac{\theta_{uk} \eta_{i\ell} Q_{k\ell}^{r_{ui}}}{\omega_{ui}(k, \ell)}. \end{aligned} \quad (11.14)$$

Here $\omega_{ui}(k, \ell)$ is the estimated probability that a given ranking r_{ui} is due to u and i belonging to groups k and ℓ , respectively, and the lower bound in the third line is Jensen's inequality $\log \bar{x} \geq \overline{\log x}$. The equality holds when

$$\omega_{ui}(k, \ell) = \frac{\theta_{uk} \eta_{i\ell} Q_{k\ell}^{r_{ui}}}{\sum_{k'\ell'} \theta_{uk'} \eta_{i\ell'} Q_{k'\ell'}^{r_{ui}}}, \quad (11.15)$$

giving us the update Eq. (11.15) for the expectation step. For the maximization step, we derive update equations for the parameters θ, η, \mathbf{Q} by taking derivatives of the log-likelihood (11.14). Including Lagrange multipliers for the normalization constraints (11.11), we obtain

$$\theta_{uk} = \frac{\sum_{i \in \partial u} \sum_l \omega_{ui}(k, \ell)}{\sum_{i \in \partial u} \sum_{k\ell} \omega_{ui}(k, \ell)} = \frac{\sum_{i \in \partial u} \sum_l \omega_{ui}(k, \ell)}{d_u}, \quad (11.16)$$

where d_u is the degree of the user u . Similarly,

$$\eta_{i\ell} = \frac{\sum_{u \in \partial i} \sum_k \omega_{ui}(k, \ell)}{\sum_{u \in \partial i} \sum_{k\ell} \omega_{ui}(k, \ell)} = \frac{\sum_{u \in \partial i} \sum_k \omega_{ui}(k, \ell)}{d_i}, \quad (11.17)$$

where d_i is the degree of item i . Finally, including a Lagrange multiplier for (11.9), we have

$$Q_{k\ell}^r = \frac{\sum_{(u,i) \in R^\circ | r_{ui}=r} \omega_{ui}(k, \ell)}{\sum_{(u,i) \in R^\circ} \omega_{ui}(k, \ell)}. \quad (11.18)$$

These equations can be solved iteratively with the following EM algorithm. Starting with an initial estimate of θ , η and \mathbf{Q} , we repeat the following steps until the parameters converge:

1. (Expectation step) use (11.15) to compute $\omega_{ui}(k, \ell)$ for $(u, i) \in R^\circ$,
2. (Maximization step) use (11.16)–(11.18) to compute θ , η and \mathbf{Q} .

The number of parameters and terms in the sums in Eqs. (11.15)–(11.18) is $NK + ML + |R^\circ|KL$. Assuming that K and L are constant, this is $O(N + M + |R^\circ|)$, and hence linear in the size of the observed ratings (see Fig. 11.4a). As the set of observed ratings R° is typically very sparse because only a small fraction of all possible user-item pairs have observed ratings, the expectation-maximization algorithm is feasible even for very large datasets.

In summary, the MMSBM approach has a double advantage: (i) it uses a model that is realistic and flexible, and (ii) the algorithm scales with the number of observed ratings, and is therefore suitable for very large datasets. In addition, it is consistent for sparse datasets, giving good results with few ratings per user (users in datasets in Sect. 11.5 rate typically less than 10 items, but they are enough to give good predictions).

11.4 State of the Art: Other Non-Network Based Collaborative Filtering Approaches

As already mentioned, collaborative filtering algorithms find similarities between users and items to make predictions, instead of focusing on the content or known external information regarding users or items other than user-item ratings. There are different strategies to identify these similarities or patterns in the recommender system. Two of the most representative approaches are neighbour-based models such as Item-Item or User-User approaches, and latent factor models such as Matrix

Factorization, commonly used also as benchmark algorithms to compare against novel recommendation models. While neighbour-based models are simple and intuitive, Matrix Factorization techniques are usually more effective because they allow us to discover the latent features underlying the interactions between users and items. Neighbour-based models are sometimes considered graph-based models, given that they use the structure of the bipartite network to compute similarities between users or between items, and they have also been called model-based algorithms [10]. In this chapter, we will consider as network-based models only those using network inference. Within this section, we explain the rationale for some of the most widely used CF algorithms and analyse some of the main theoretical differences between them and the network-based models SBM and MMSBM.

Item-Item

Neighbour-based CF models generate recommendations using only information about rating profiles for different users. There are two approaches, the User-User approach and the Item-Item approach. In the former, the algorithm finds users with a rating history similar to the query user (neighbours) and generates recommendations using this neighbourhood; for the Item-Item, the algorithm finds similar items to the query item based on their rating history, and generates recommendations using the query item's neighbourhood. For rating systems with much more users than items (as is the case in the datasets we analyse), the Item-Item approach gives better predictions than the User-User approach and is computationally more efficient, therefore from now on we will focus on the Item-Item algorithm, taking into account that the User-User model is computed analogously [10]. Let us assume that we have a list of users $U = \{u_1, \dots, u_N\}$ and items $I = \{i_1, \dots, i_M\}$, which the users have rated. The Item-Item approach assumes that the rating from user u to an item i should be similar to the rating she gives to similar items to i . Considering the vector $\vec{i} \in R^N$ of ones for users that have rated item i and zeros otherwise, we can obtain the similarity between item pairs (i, j) by computing the cosine similarity between \vec{i} and \vec{j} as,

$$\text{sim}(i, j) = \cos(i, j) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 \|\vec{j}\|_2}, \quad (11.19)$$

where $\|\cdot\|_2$ denotes the Euclidean norm of a vector. For other adjusted versions of the similarity see [10]. Note that we can only establish similarities between items that have been rated by the same users.¹ According to the similarity measure, we define the neighbourhood of an item i , ∂i as those k items with highest similarity i .

¹For the User-User model the reasoning is equivalent: each user is represented by a vector with all the items she has rated V_u . The similarity between users would be computed as in Eq. (11.19) as $\text{sim}(u, v)$.

Hence, the prediction of r_{ui} would be the average of the ratings that user u gave to the items in the neighbourhood of item i as,

$$r_{ui} = \frac{\sum_{j \in \partial i} (\text{sim}(i, j) \cdot r_{uj})}{\sum_{j \in \partial i} (|\text{sim}(i, j)|)}. \quad (11.20)$$

Note that if in the k -nearest neighbours there is no item rated by u , the algorithm cannot perform a prediction, which may happen for sparse datasets. Also, the algorithm assumes a linear psychological scale on the ratings, that is, a rating of 5 is seen as five times better than a rating of 1, but unfortunately this is not necessary in agreement with people's perception [26].

Matrix Factorization Approaches

The most widely used methods for recommendations are the Latent feature or Matrix Factorization methods [11, 30]. Latent feature models assume that there is a space of latent attributes of users and items that determine user-item ratings. Therefore, ratings are not independent from one another but set by the specific position in the latent feature space of users and items. Specifically, MF assumes that there exists a single latent feature space for both users and items, and that the rating of user u to item i is proportional to the closeness between the two in this space. The dimension of the latent space K is much smaller than the number of users and items, such that the problem is dimensionally reduced. Formally, this is equivalent to assuming that the matrix of observed ratings R^O (with a number of rows equal to the number of users N , and a number of columns equal to the number of items M) can be decomposed into

$$R^O = P Q, \quad (11.21)$$

where P is an $N \times K$ matrix associated with the users and Q is a $K \times M$ matrix associated with the items. Each row of the P matrix p_u could be seen as a K -dimensional vector with the feature values of user u that describe her, and each column of the Q matrix q_i is a K -dimensional vector with the values of the features that describe item i , with $K \ll N$ and $K \ll M$.

The most efficient method until now to factorize the rating matrix, although there are several methods, is the singular value decomposition (SVD) [30]. This method finds the two smaller matrices whose product minimizes the difference with the original ratings matrix (measured as a means squared error). In addition it uses gradient descent to learn a Matrix Factorization (by taking derivatives of the error function over the parameters of the model it is trying to infer). The predicted rating is then

$$r_{ui} = \sum_k p_{uk} q_{ik}. \quad (11.22)$$

SVD-MF algorithm is computationally very efficient and makes very good predictions. Also, it has the advantage that it results in intuitive meanings of the resultant matrices and that the resulting algorithm is scalable (see Fig. 11.4) so it can potentially handle very large datasets. The main problem with this approach is that features that describe the users and the items are the same, which imposes severe constraints on the expressiveness of the model (for instance, two users close in feature space must like the same type of items and there is little flexibility to account for the fact that some users might like some items but have different opinions about other items).

Moreover, as the prediction is (with some corrections) the scalar product of the users' and items' feature vectors, this is equivalent to assuming linearity between ratings instead of assuming that ratings are independent categories as was assumed for the bipartite SBM and MMSBM.

As an extension to the "classical" MF, we also consider a mixed-membership implementation of MF, the Mixed-Membership Matrix Factorization (MMMMF) [12]. The MMMF model combines Matrix Factorization with a mixed membership context bias. In MMMF, users and items are endowed with both latent factor vectors (p_u and q_i) and discrete topic distribution parameters ($\theta_{uk}^U \in K^U$ and $\theta_{ij}^M \in K^M$). Together with the user and item topics, MMMF models also introduce the affinity of user u to item topic k as c_u^k and the affinity of item i to user topic j as d_i^j . The topic distribution parameters and the affinity of users and items to the topics jointly specify a context bias β_{ui}^{jk} . Therefore, a user generates a rating for an item by adding the contextual bias to the MF inner product with some Gaussian noise,

$$r_{ui} \sim \mathcal{N}(p_u \cdot q_i + \beta_{ui}^{jk}, \sigma^2). \quad (11.23)$$

In [12] authors consider two different MMMF models that differ in how the contextual bias is built. The Topic-Indexed Bias Model (TIB) assumes that the contextual bias decomposes into a latent user bias and a latent item bias so that $\beta_{ui}^{jk} = \sum_{k=1}^{K^M} c_u^{t(t)} \theta_{ik}^{M(y)} + \sum_{j=1}^{K^U} d_i^{j(t)} \theta_{uj}^{U(t)}$. The Topic-Indexed factor Model (TIF) assumes that the joint contextual bias is an inner product of topic-indexed factor vectors, so that $\beta_{ui}^{jk} = \sum_{k=1}^{K^M} \sum_{j=1}^{K^U} \theta_{ik}^{M(y)} \theta_{uj}^{U(t)} c_u^{k(t)} \cdot d_i^{j(t)}$. They use a Gibbs sampling MCMC procedure to draw samples of topic and parameter variables. Then, the posterior mean prediction for each user-item pair under these MMMF models is,

$$\frac{1}{T} \sum_{t=1}^T \left(p_u^{(t)} \cdot q_i^{(t)} + \beta_{ui}^{jk} \right). \quad (11.24)$$

The results shown in Sect. 11.5 are for the MMMF-TIF model since it outperforms the MMMF-TIB in all the datasets. Note that analogously to the MF, MMMF also assumes linearity between ratings values.

11.4.1 Advantages of Network-Based Models

There are a number of advantages to using the network-based models we have presented (the bipartite SBM and the MMSBM) compared to previous work on collaborative filtering.

First, unlike Matrix Factorization approaches such as [11] or their probabilistic counterparts [31–33], the ratings $r_{ui} \in \{1, 2, 3, 4, 5\}$ are not treated as integers. As has been established in the literature, giving a movie a rating of 5 instead of 1 does not mean the user likes it five times more [26]. Indeed, the results in Sect. 11.5 suggest that it is better to think of different ratings simply as different labels on the links of the network.

Second, network-based methods yield a distribution over the possible ratings directly, rather than a distribution over integers or reals that must be somehow mapped to the space of possible ratings [31–33]. The network-based models we have presented considered the observed ratings as a bipartite network with metadata (or labels) on the links. An alternative approach would be to consider a multi-layer representation of the data as in [34].

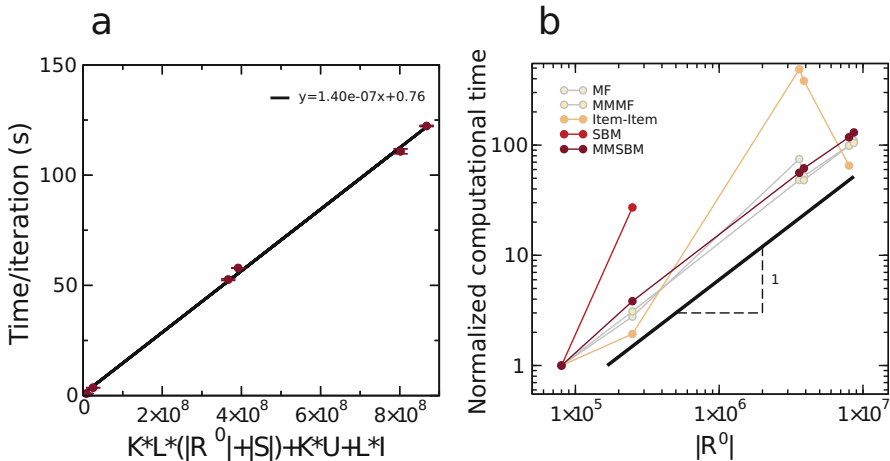


Fig. 11.4 Scalability: (a) Scalability of the MMSBM algorithm. Each point represents the average time per iteration in seconds for each of the datasets we use in the study (100K MovieLens, 10M MovieLens, Yahoo! Songs, W-M dating agency, M-W dating agency and Amazon books) versus the number of parameters computed at each iteration $K * L * (|R^O| + |S|) + K * N + L * M$ where N is the number of users, M is the number of items and $|R^O|$ is the number of observed ratings, $|S|$ is the number of different ratings values for each recommender systems and K and L are the number of groups for users and items, respectively ($K = L = 10$ for all the datasets; see Table 11.1 for remaining parameters for each dataset). The continuous line is the linear fit of the real data, which shows that the computational times per iteration scales linearly with the size of the corpus for the whole range. (b) Scaling of the different benchmark algorithms we consider in our analysis with the total number of observed ratings. The vertical axis is normalized by the computational time of the smallest dataset—100K MovieLens. MF, MMMF and MMSBM algorithms scale linearly with the total number of observed edges, while the Item-Item algorithm does not. Note that for the bipartite SBM we could only get results for the two smallest datasets, so we cannot establish a linear relationship in this case

Third, the bipartite SBM and the MMSBM do not assume that the matrices \mathbf{Q} have any particular structure. In particular, they do not assume either that groups of individuals correspond to groups of items or that individuals prefer items that belong to their own group (which mathematically would result in diagonal \mathbf{Q} matrices). Thus, the SBMs and the resulting algorithms can learn arbitrary couplings between groups of individuals and groups of items, and do so independently for each possible rating, thus overcoming the limitation of expressivity of MF factorization approaches that consider a diagonal \mathbf{Q} matrix. Importantly, the MMMF does not circumvent this issue despite considering arbitrary couplings between users/items and topics. In fact, MMMF rating predictions are the sum of a MF term and a correction that uses mixed group memberships that are unrelated to the feature vectors [12]. While this is an improvement over MF, it does not fundamentally remove the limiting assumption that each group of users has a corresponding group of items that they prefer. Indeed, our numerical results show that the performance of MMMF is fairly close to that of MF in the datasets we considered.

Finally, all the network-based models presented here do not assume that individuals only see movies (say) that they like, and they do not treat missing links as zeroes or low ratings as is typically done in MF algorithms that need a full matrix to decompose. There are other physics-inspired methods that exploit the structure of the bipartite user-item network and use classical physics processes to make recommendations such as random walks [35] or heat diffusion [36]. However, all these approaches are used for link prediction, that is, they only try to predict which item would be collected by a user [37].

11.5 Results

We show the performance of the network-based and the Item-Item and MF algorithms for six datasets: the MovieLens 100K and 10M datasets with 100,000 and 10,000,000 ratings, respectively (<https://movielens.org>), Yahoo! songs (R3—<https://webscope.sandbox.yahoo.com/catalog.php?datatype=r>), Amazon books [38, 39] (<http://jmcauley.ucsd.edu/data/amazon/>), and the dataset from LibimSeTi.cz dating agency [40] (<http://www.occamlab.com/petricek/data/>), which is split into two datasets, one consisting of males rating females and vice versa. These datasets are diverse in the types of items considered, the sizes $|S|$ of the sets of possible ratings and the density of observed ratings (see Table 11.1).

To check the predictive power of the different algorithms we show the results for a fivefold cross validation in each of the six datasets. That is, we divide each dataset into five equal parts and we make the five possible combinations of using one part as the test set and the other four as the training set. We measure the predictability in terms of accuracy, that is, the number of ratings predicted correctly, and the mean absolute error (MAE). Figure 11.5 shows the performance for the two network-based models, the simple bipartite SBM (using the approach in [14]) and the Mixed-Membership Stochastic Block Model (MMSBM) (using the approach

Table 11.1 Dataset characteristics

Dataset	Ratings scale S	#Users	#Items	#Ratings
MovieLens 100K	{1, 2, 3, 4, 5}	943	1682	100,000
MovieLens 10M	{0.5, 1, 1.5, ..., 5}	71,567	65,133	10,000,000
Yahoo! Songs	{1, 2, 3, 4, 5}	15,400	1000	311,700
M-W dating agency	{1, 2, ..., 10}	220,970	135,359	4,852,455
W-M dating agency	{1, 2, ..., 10}	135,359	220,970	10,804,040
Amazon book	{1, 2, 3, 4, 5}	73,091	539,145	4,505,893

The total number of possible ratings is different for each dataset; ratings are in a scale from 1 to 5 in all datasets for the two dating agency datasets, which have a rating scale from 1 to 10. Ratings are integers except for the MovieLens 10M dataset which allows half-integer values. Note that, in the latter case we expect a smaller MAE than if only integer values were allowed. All datasets have millions of ratings except for MovieLens 100K and Yahoo! Songs

in [15]). Moreover, we show the comparison of the network-based approaches with three benchmark algorithms (see Sect. 11.4): the Item-Item algorithm [10], which predicts r_{ui} based on the observed ratings of user u for items that are the most similar to i , a “classical” Matrix Factorization (MF) [11] and Mixed-Membership Matrix Factorization (MMMMF) [12]; as well as a baseline naive algorithm that assigns to each test rating r_{ui} the average of the observed ratings for item i , that is $r_{ui} = \frac{1}{d_i} \sum_{u' \in \partial_i} r_{u'i}$.

The results for the MMSBM are for $K = L = 10$, i.e., that is 10 groups of users and 10 groups of items (recall that there is any correspondence between these groups). The performance for larger choices of K and L does not improve significantly [15]. Since iterating the EM equation of Eqs. (11.15)–(11.18) can lead to different solutions depending on the initial conditions, the results correspond to an average of the predicted probability distribution of ratings over 500 independent runs. There is a freely available implementation of the MMSBM in gitHub by Bill Jeffries (<https://github.com/billjeffries/mixMemRec>). The code is written in Spark’s recommender library and can process large datasets. Notice however that the current implementation gives the results for a single run, therefore one should expect accuracies to be lower if a single run is considered.

The bipartite SBM does not require a pre-specification of model parameters since they are sampled by the algorithm. You can find a freely available implementation of the code (<http://seeslab.info/downloads/network-c-libraries-rgraph/inrgraph-2.2.1/recommender/>). For the Item-Item algorithm implemented in Lenskit, we set $k = 50$; for the Matrix Factorization we also used the Lenskit implementation with $k = 50$ features, a learning rate of 0.002 and an initialization of 0.1 for every user-feature and item-feature value as suggested in [26]. Finally, for MMMF we use the Matlab implementation provided by the authors (<https://code.google.com/archive/p/m3f/>).

Another thing to take into account is that the network-based models, both the bipartite SBM and the MMSBM, are probabilistic models. That means that for each rating a user gives an item we have a probability distribution of ratings that results from the average of the probabilities for all the sampling set. Therefore, we can choose how to make predictions from the probability distribution of ratings: the mode (that is the rating with the highest probability), the mean or the median. As stated earlier, we measure the performance in terms of accuracy and the mean absolute error (MAE), which gives us an idea how far predictions are from the real values. For the network-based model, the best estimator for the accuracy is the most likely rating from the probability distribution of ratings, while for the MAE the best estimator is the median. In contrast, the predictions of the MF, MMMF and Item-Item models are a single real per rating.

In Fig. 11.5 we find that in most cases the network-based approaches, the bipartite SBM and MMSBM, outperform the Item-Item algorithm, MF and MMMF. Indeed, when considering the accuracy the MMSBM is significantly better than MF and MMMF for all the datasets we tested, and better than the item-item algorithm in five out of six datasets, the only exception being the Amazon Books dataset. In terms of the mean absolute error (MAE), the MMSBM is the most accurate in four out of the six datasets (item-item, MF and MMMF produce smaller MAE in the Amazon Books and MovieLens 10M datasets). Note that the Amazon dataset is different from the others in that users only rate items after buying them, and knowing a priori the average rating of the item given by previous buyers, which might bias their choices.

Interestingly, the MMSBM approach produces results that are almost identical to those of the bipartite SBM [14] for the two examples for which inference with the bipartite SBM is feasible. In particular, the MMSBM achieves the same accuracy with $K = L = 10$ in the mixed-membership model as the bipartite SBM with sampling models with 50 groups on average. This suggests that many of the groups observed in [14] are in fact mixtures of larger groups, and that the additional expressiveness of the MMSBM allows us to succeed with a lower-dimensional model.

Given that in general Matrix Factorization approaches outperform the Item-Item model, and that the MMSBM and the bipartite SMB give similar results, we quantify the improvement of the MMSBM over the classical MF (with very similar results to the MMMF). To do so, we compute the relative improvements in the accuracy (%) as

$$\frac{(\text{acc}_{\text{MMSBM}}) - (\text{acc}_{\text{MF}})}{(\text{acc}_{\text{MF}})} * 100, \quad (11.25)$$

with improvements of 5% in the MovieLens 100K dataset, 45% in the MovieLens 10M, 41% in the Yahoo songs, 42% in the M-F LibimSeti agency, 27% in the F-M LibimSeti agency and finally a 3% improvement in the Amazon dataset. For the Amazon dataset, the relative improvement of the Item-Item over the MMSBM is of 13%.

11.6 Discussion

We have shown that network-based approaches, based on inference using the block structure of social networks, give predictions of human preferences that are in most of the cases significantly and considerably more accurate than leading collaborative filtering recommendation algorithms.

In the case of the simple bipartite SBM, it is worth noting that the gain in accuracy comes at the expense of computational cost as a result of the Monte Carlo sampling of the user and item partition space. Although the algorithm is able to give predictions on datasets in the order of ~ 1000 of users and items and $\sim 100,000$ of ratings, handling even one order of magnitude is challenging. Instead, the MMSBM inference using expectation-maximization method results in a scalable algorithm able to handle datasets with millions of ratings.

In any case, network-based recommender systems not only provide better predictions, but also have some desirable features: they are analytically tractable allowing for a mathematically rigorous approach, they are based on plausible social models, and they provide interpretable results.

With respect to mathematical rigour, the Bayesian approach used by the bipartite SBM [14] is the complete and correct probabilistic treatment of the observations. However, the results of the MMSBM suggest that introducing the mixed-membership of users and items is already equivalent to sampling over different sets of simple bipartite SBMs [15].

Importantly in both cases, we obtain an estimate of the whole probability distribution for each rating. From this, we can choose how to make predictions using the most likely rating, the mean or the median among others. In contrast, recommender systems like those based on Matrix Factorization or Item-Item give predictions that are a single number, the most likely rating (or a real number that should be rounded to the closer value in the ratings set), that may even be outside the rating range (for example, $r_{ui} = 1.1$ when $S \in \{0, 1\}$). Additionally, these algorithms assume that ratings are linearly spaced in the mind of users (that is, that the difference between $r = 1$ and $r = 2$ is the same as between $r = 4$ and $r = 5$), which does not seem to be in accordance with people's perception [26].

Finally, network-based approaches are based on models that were originally defined and are widely used to explain how social agents establish relationships, and are therefore in a better position to illuminate which social and psychological factors determine human preferences. As an interesting by-product of this, we note that it is possible to use them to infer demographic properties from ratings alone, a subject that is of much current interest for commercial purposes such as Social Marketing [41, 42].

The future of network-based recommender systems is likely to involve the introduction of contextual information about users and/or items into the inference process. The network-based models we have discussed in this chapter have the advantage that metadata can mathematically be introduced in the form of *a priori*

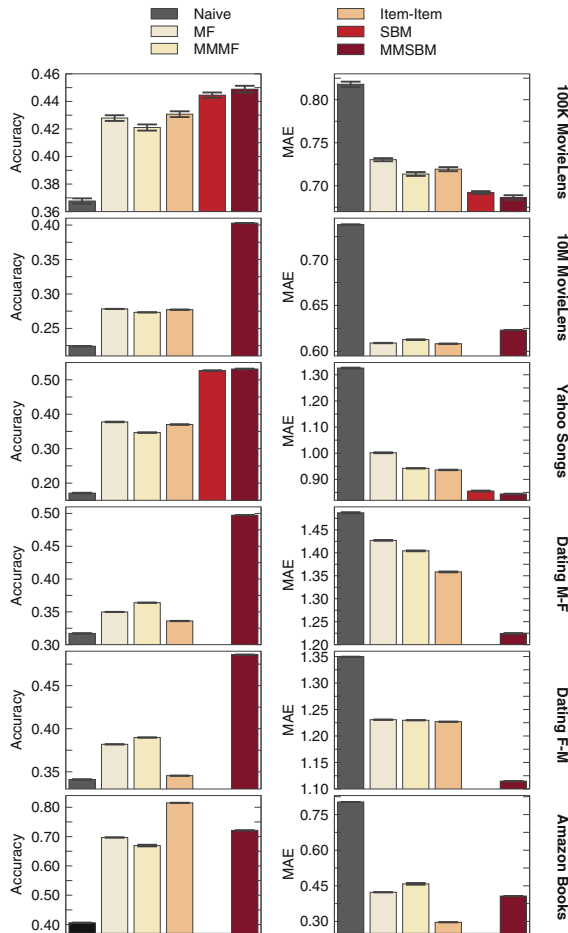


Fig. 11.5 Algorithm comparison: From top to bottom, the datasets are MovieLens 100K, MovieLens 10M, Yahoo Songs, men rating women (M-W) in the LibimSeTi dataset, women rating men (W-M) in the LibimSeTi dataset and Amazon books. The left column displays the accuracy of the algorithms in each dataset, i.e., the fraction of ratings that are exactly predicted by each algorithm. The right column displays the mean absolute error (MAE) in the predicted vs. actual rating, treated as an integer or half-integer. In all cases, the bars are the average of a fivefold cross-validation and the error bars correspond to the standard error of the mean. The bipartite SBM algorithm does not scale to the larger datasets, hence it was evaluated only on the MovieLens 100K and Yahoo Songs datasets. Importantly, bipartite SBM algorithm achieves similar accuracy to the MMSBM on the datasets it can handle. The MMSBM model and algorithm achieves the best (highest) accuracy in five out of six datasets, and the best (lowest) MAE in four out of six datasets

probabilities for model parameters and specifically for group membership vectors. The intuition behind this idea is simple: we expect users (items) with similar associated metadata to have similar membership vectors. With this type of approach, network-based models will be better suited to industrially relevant problems such as

the problem that arises when a new product is introduced into the market. The use of relevant metadata can be informative about the most plausible group membership vectors for each item and therefore help in identifying the range of users who could potentially be interested in that product.

References

1. Castells M (2001) *The Internet Galaxy: Reflections on the Internet, Business, and Society*. (Oxford University Press, Inc., New York, NY, USA).
2. Kalakota R, Robinson M (1999) *e-Business*. (Addison Wesley Roadmap for Success).
3. Brynjolfsson E, Hu Y, Smith MD (2003) Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers. *Management Science* 49(11):1580–1596.
4. Netflix Prize Rankings. Hacking NetFlix. <http://www.hackingnetflix.com/2006/10/>. Accessed 09-March-2017
5. The BellKor 2008 Solution to the Netflix Prize. http://www.research.att.com/people/Volinsky_Christopher_T/custom_index.html. Accessed 09-March-2017
6. Netflix Prize Leaderboard 2009. <http://www.netflixprize.com/leaderboard.html>. Accessed 09-March-2017
7. Lohr S (2009) A \$1 million research bargain for Netflix, and maybe a model for others. *The New York Times* 22.
8. Bobadilla J, and Ortega F, and Hernando A and Gutiérrez A (2013) A. Recommender systems survey *Knowledge-Based Systems* 46:109–132.
9. Deshpande M, Karypis G (2004) Item-based top-n recommendation algorithms. *Acm Transactions on Information Systems* 22:143–177.
10. Sarwar B, Karypis G, Konstan J, Riedl J (2001) *Item-based collaborative filtering recommendation algorithms*, WWW '01. (ACM, New York, NY, USA), pp. 285–295.
11. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37.
12. Mackey L, Weiss D, Jordan M I (2010) Mixed membership matrix factorization. *Proceedings of the 27th International Conference on Machine Learning*, pp 711–718.
13. Guimerà R, Sales-Pardo M (2009) Missing and spurious interactions and the reconstruction of complex networks. *Proc. Natl. Acad. Sci. U. S. A.* 106(52):22073–22078.
14. Guimerà R, Llorente A, Moro E, Sales-Pardo M (2012) Predicting human preferences using the block structure of complex social networks. *PLoS One* 7(9):e44620.
15. Godoy-Lorite A, Guimerà R, Moore C, Sales-Pardo M (2016) Accurate and scalable social recommendation using mixed-membership stochastic block models. *Proc. Natl. Acad. Sci. U. S. A* 113(50):14207–14212.
16. Girvan M, Newman ME (2002) Community structure in social and biological networks. *Proc. Natl. Acad. Sci. U.S.A.* 99(12):7821–7826.
17. Guimera R, Sales-Pardo M, Amaral LAN (2004) Modularity from fluctuations in random graphs and complex networks. *Physical Review E* 70(2):025101.
18. Guimera R, Sales-Pardo M, Amaral LA (2007) Classes of complex networks defined by role-to-role connectivity profiles. *Nature Physics* 3(1):63–69.
19. Fortunato S (2010) Community detection in graphs. *Physics Reports* 486(3):75–174.
20. Rombach MP, Porter MA, Fowler JH, Mucha PJ (2010) Core-Periphery Structure in Networks. *Physics Reports* 74(1):167–190.
21. White HC, Boorman SA, Breiger RL (1976) Social structure from multiple networks. I. Blockmodels of roles and positions. *American Journal of Sociology* pp. 730–780.

22. Rovira-Asenjo N, Gumí T, Sales-Pardo M, Guimerà R (2013) Predicting future conflict between team-members with parameter-free models of social networks. *Scientific Reports* 3:1999.
23. Aicher C, Jacobs AZ, Clauset A (2014) Learning latent block structure in weighted networks. *Journal of Complex Networks* 3(2):221–248.
24. Peixoto TP (2017) Nonparametric weighted stochastic block models. *arXiv preprint arXiv:1708.01432*.
25. Nowicki K, Snijders TAB (2001) Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association* 96(455):1077–1087.
26. Ekstrand MD, Ludwig M, Konstan JA, Riedl JT (2011) Rethinking the recommender research ecosystem: reproducibility, openness, and LensKit. *Proceedings of the Fifth ACM Conference on Recommender Systems* pp. 133–140.
27. Jaynes ET (2003) *Probability Theory: The Logic of Science*. (Cambridge University Press).
28. Airoldi EM, Blei DM, Fienberg SE, Xing EP (2008) Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.* 9(2008):1981–2014.
29. Ball B, Karrer B, Newman ME (2011) Efficient and principled method for detecting communities in networks. *Physical Review E* 84(3):036103.
30. Paterek A (2007) *Improving regularized singular value decomposition for collaborative filtering*. pp. 39–42.
31. Meeds E, Ghahramani Z, Neal RM, Roweis ST (2006) Modeling dyadic data with binary latent factors in *Advances in Neural Information Processing Systems 19*, eds. Schölkopf B, Platt J, Hoffman T. (MIT Press, Cambridge, MA), pp. 977–984.
32. Salakhutdinov R, Mnih A (2008) Probabilistic matrix factorization. *Advances in Neural Information Processing Systems (NIPS '08)* pp. 1257–1264.
33. Shan H, Banerjee A (2010) *Generalized Probabilistic Matrix Community for Collaborative Filtering*, Proceedings of the 2010 IEEE International Conference on Data Mining. (IEEE Computer Society, Washington, DC, USA), pp. 1025–1030.
34. Peixoto TP (2015) Model selection and hypothesis testing for large-scale network models with overlapping groups. *Phys. Rev. X* 5:011033.
35. Zhou T, Ren J, Medo M, Zhang Y-C (2007) Bipartite network projection and personal recommendation. *Physical Review E* 76(2007):046115
36. Zhou T, Kuscsik Z, Liu J-G, Medo M, Wakeling JR, Zhang Y-C (2010) Solving the apparent diversity-accuracy dilemma of recommender systems. *Proc. Natl. Acad. Sci. U. S. A* 107(2010):4511
37. Yu F, Zeng A, Gillard S, Medo M (2016) Network-based recommendation algorithms: A review. *Physica A: Statistical Mechanics and its Applications* 452(2016):192–208.
38. McAuley J, Targett C, Shi Q, van den Hengel A (2015) *Image-Based Recommendations on Styles and Substitutes*, SIGIR '15. (ACM, New York, NY, USA), pp. 43–52.
39. McAuley J, Pandey R, Leskovec J (2015) *Inferring Networks of Substitutable and Complementary Products*, KDD '15. (ACM, New York, NY, USA), pp. 785–794.
40. Brozovsky L, Petricek V (2007) *Recommender System for Online Dating Service*. (VSB, Ostrava).
41. Leo Y, Karsai M, Sarraute C, Fleury E (2016) Correlations of consumption patterns in social-economic networks. *arXiv preprint arXiv:1609.03756*.
42. Andreasen AR (2002) Marketing social marketing in the social change marketplace. *Journal of Public Policy & Marketing* 21(1):3–13.