Data Min Knowl Disc (2017) 31:1031–1059
DOI 10.1007/s10618-017-0504-3

Enhancing social collaborative filtering through the application of non-negative matrix factorization and exponential random graph models

Georgios Alexandridis[1] · **Georgios Siolas**[1] · **Andreas Stafylopatis**[1]

Received: 13 November 2015 / Accepted: 21 March 2017 / Published online: 31 March 2017

Abstract Social collaborative filtering recommender systems extend the traditional user-to-item interaction with explicit user-to-user relationships, thereby allowing for a wider exploration of correlations among users and items, that potentially lead to better recommendations. A number of methods have been proposed in the direction of exploring the social network, either locally (i.e. the vicinity of each user) or globally. In this paper, we propose a novel methodology for collaborative filtering social recommendation that tries to combine the merits of both the aforementioned approaches, based on the soft-clustering of the Friend-of-a-Friend (FoaF) network of each user. This task is accomplished by the non-negative factorization of the adjacency matrix of the FoaF graph, while the edge-centric logic of the factorization algorithm is ameliorated by incorporating more general structural properties of the graph, such as the number of edges and stars, through the introduction of the exponential random graph models. The preliminary results obtained reveal the potential of this idea.

Keywords Social collaborative filtering · Non-negative matrix factorization · Exponential random graph models · Recommender systems

Responsible editor: G. Karypis.

✉ Georgios Alexandridis
gealexandri@islab.ntua.gr

Georgios Siolas
gsiolas@islab.ntua.gr

Andreas Stafylopatis
andreas@cs.ntua.gr

[1] School of Electrical and Computer Engineering, National Technical University of Athens, 157 80 Zografou, Greece

1 Introduction

Collaborative-filtering (CF) is one of the most well-studied and widely-adopted approaches to recommendation (Adomavicius and Tuzhilin 2005). At the core of this methodology lies the utilization of the previously provided evaluations of items by some users as an indication of future preference for other users. The underlying logic of CF algorithms may be roughly put into three classes; *memory-based* (or *heuristic-based*), *model-based* and *hybrid*. The algorithms of the first class utilize all of the available evaluations of a specific user and/or item in order to produce a prediction, while the algorithms of the second class try to build models for the users out of the available data. Finally, the algorithms of the third class try to combine the advantages and diminish the disadvantages of the other two classes.

Each of the said approaches has its own individual strengths and weaknesses. Memory-based systems are able to produce recommendations fast and allow for a direct interpretation of the results (e.g. "item i is proposed because it is similar to other items you have consumed/other like-mined users have a preference for it"); however they might fail to produce recommendations for new items or users. Model-based systems, on the other hand, can handle sparse data more easily and when used as regression models they may optimize certain statistical accuracy metrics such as the root mean square error (i.e. the Netflix Prize contest; Bennett and Lanning 2007). Those merits, though, come at the expense imposed by the model building process and most often a trade-off between system performance and scalability is sought. Finally, hybrid approaches exhibit, in the general case, increased complexity and are more difficult to implement.

The work presented in this paper falls in the first class of the aforementioned categories. It is a memory-based recommendation algorithm that utilizes the available social information in the form of ties between users. It employs *non-negative matrix factorization*, a model-based technique, in a different context. While the traditional techniques, e.g., *ALSWR* (Zhou et al. 2008) and *RSVD++* (Koren 2008), factorize the matrix containing the evaluations of users on items, seeking to uncover latent user and item factors, our approach is to factorize the adjacency matrix of the induced subgraph of the FoaF network of each user. The purpose of this deviation is to position the user's neighbors into soft clusters. Additionally, the edge-centric logic of the antecedent process is alleviated through the introduction of the *exponential random graph models*, that are able to model more complex graph characteristics such as the number of edges and triangles.

2 Memory-based collaborative filtering recommendation

Figure 1 outlines the key components of a memory-based, classic as well as social, collaborative filtering recommender system. In the first step, *Evaluation Data* and the *Social Information* (if available) are fed into the *Similarity Estimation* module, whose role is to amass all of the available information for the purpose of assessing the degree to which users (or items) are alike.

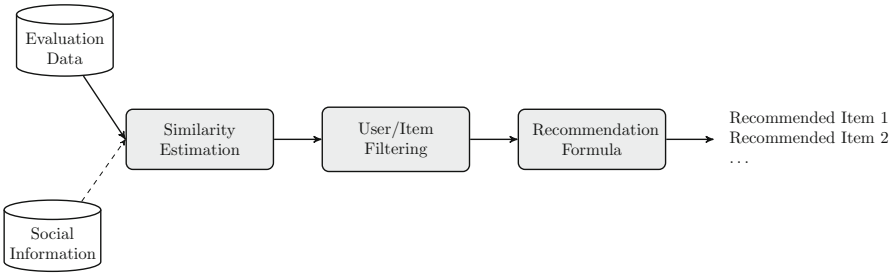

Fig. 1 Memory-based, classic and social collaborative-filtering recommender system

Classic collaborative filtering approaches rely solely on the evaluation data provided and employ standard *similarity indices* derived from the statistical literature, such as the *Pearson Correlation Coefficient*, the *Euclidean Distance*, the *Cosine Similarity* and others (Desrosiers and Karypis 2011), in order to estimate how similar or dissimilar users and items are. In general, no similarity index may be considered more descriptive than the other; their relative performance is heavily dependent on the structure and the characteristics of the underlying data.

Social collaborative filtering methodologies additionally resort to the *Social Information* and the explicit ties among users in order to estimate user similarity. There are a number of ways in which the social information is processed in order to derive user proximity that are going to be presented in more detail in the following section.

After user and item similarities are estimated, the role of the *User/Item Filtering* module is to specify how users or items group together, based on their similarity score. There are roughly two approaches to this task; either label as most similar all those users whose similarity index is above a predefined threshold or limit computation only to the N most similar users. Again, neither approach may be generally considered superior to the other; it is the application domain and the underlying data at hand that dictate which one is more fit for use.

The last step of the memory-based, collaborative filtering recommender system involves the actual *Recommendation Formula* that combines the similarity indices and the neighborhood for each target user in order to produce the recommendations. The most widely-adopted algorithm is the *Resnick formula* (Resnick et al. 1994) (Eq. 1)

$$\widehat{r_{u,i}} = \overline{r_u} + \frac{\sum_{n=1}^{N} (r_{n,i} - \overline{r_n}) \times \text{sim}(u, n)}{\sum_{n=1}^{N} \text{sim}(u, n)} \tag{1}$$

On the equation above, the term $\widehat{r_{u,i}}$ denotes the predicted utility value of item i to user u (who has not evaluated it yet), which is the sum of two terms; the said user's average evaluation value ($\overline{r_u}$) and the sum of the differences of his neighbors' scoring of item i ($r_{n,i}$) minus their mean evaluation value ($\overline{r_n}$), weighted by their similarity

to u ($\text{sim}(u, n)$) and normalized over the sum of the similarities of all neighbors (denominator of Eq. 1).

3 Social filtering algorithms

Social collaborative filtering algorithms, either memory-based or model based, have also been extensively studied in the literature (Yang et al. 2014; Bellogin et al. 2011). The intuition behind exploiting social information as a tool for user similarity estimation is rather straightforward when projected to the real life; people often resort to the opinion of their friends and acquaintances prior to purchasing an item or making a decision (Alexandridis et al. 2013).

The social information is most commonly incorporated into the recommendation algorithms in two different ways. The first is the local approach, in which only the social ties within a certain vicinity of each user are taken into account. The second is the global approach where all links in the social graph, or the connected component to which the examined user belongs, are processed in order to estimate user similarity (Alexandridis et al. 2015).

3.1 Local exploration

The social graph may be traversed on the local level in a breadth-first fashion. One of the earliest works in this field is the *MoleTrust* system (Massa and Avesani 2009) that was firstly applied in *Trust Networks*. Those networks are formed on the evaluations (usually private) that their members give on other members. The edges are directional and their weight is proportional to the degree of appreciation in-between their users.

For a given user, the MoleTrust algorithm reads a subgraph from the social network and transforms it into an acyclic form (a tree) by removing all loops in it. The target user is set to be the root node of the tree and the users in his social neighborhood the child nodes. Starting from the root node, the tree is traversed in a depth-first strategy. The *propagation horizon* determines the depth of the exploration; the most common configurations being MoleTrust-1 (only the immediate neighbors are considered) and MoleTrust-2 (both the immediate neighbors and their neighbors as well). When the propagation horizon is bigger than one, the weight values are accumulated in the path. For example, in the case of the MoleTrust-2 algorithm, if user u trusts a to a degree of $\text{trust}(u, a)$ and user a trusts b to a degree of $\text{trust}(a, b)$ in turn, then user u is considered to trust b to a degree of $\text{trust}(u, b) = \text{trust}(u, a) \times \text{trust}(a, b)$.

The social graph may also be traversed following a breadth-first strategy. This is the underlying principle of the *TidalTrust* algorithm (Golbeck 2005). Instead of setting a global value for the propagation horizon, the shortest path between the target user and each other user in his connected component of the social network is computed. However, only edges above a predefined threshold t are included in the computations and the users that those edges correspond to, form the *Web of Trust (WOT)* of each examined user. The TidalTrust formula is given in Eq. 2

$$\text{trust}(u, b) = \frac{\sum_{i \in WOT(t)} \text{trust}(u, i) \times \text{trust}(i, b)}{\sum_{i \in WOT(t)} \text{trust}(u, i)} \quad (2)$$

It should also be noted that TidalTrust is a recursive algorithm; if intermediate user i is not directly attached to b, then the weight trust(i, b) is computed using the same equation and the procedure is repeated until b's neighbors are reached. Finally, the computed similarity values (weights) of Eq. 2 are plugged into the formula of Eq. 1 in order to produce recommendations.

Local exploration approaches achieve good results in certain cases (i.e. when an active user in the social network has provided few evaluations) (Massa and Avesani 2009), in the general case, however, they fail to capture more advanced aspects of user-to-user interaction.

3.2 Global exploration

Apart from the local exploration, the social graph may be aggregated globally, in which case a *reputation* value is assigned to each user based on his connections and his specific position within the network (Nunez-Gonzalez et al. 2015). The most common choice is the family of *graph-based* collaborative filtering algorithms, which perform a *Random Walk* in the social graph and output as a reputation value the probabilities of the stationary distribution, like in the *TrustWalker* System (Jamali and Ester 2009). This methodology may be further extended by adding a restart probability to the random walk (Konstas et al. 2009), making its operation resemble the popular *PageRank* algorithm that crawls pages on the World Wide Web. Even though the global exploration of the social network provides in-depth insight of the relationships among its members, it is in many cases a computationally challenging task, especially in very large networks, that yields results the vast majority of which cannot be used in the recommendation process (i.e. the reputation of a user far away from the target user is very unlikely to be of any importance to him).

Other model-based social collaborative filtering approaches put an effort in incorporating the social information, as represented by the adjacency matrix of the whole social graph, into the matrix factorization process discussed in Sect. 1. Examples include, among others, the *SoRec* system and the *STE* model (Ma et al. 2009; Yang et al. 2012). Even though some of those systems achieve a certain optimization of specific statistical accuracy metrics (like the *SocialMF* system; Jamali and Ester 2010), the inclusion of the social information is not always a straightforward task. On top of that, the main problem of finding the optimum trade-off between system performance and scalability still remains. Our aim, in this work, is to exploit the benefits the factorization algorithms bring in, while, at the same time, to diminish the role of the negative side-effects. It is for this reason that we have placed our focus on the FoaF level instead of the whole graph. Contrary to the way the systems outlined in Sect. 3.1 function (directly estimating user similarities between the target user and her social "surroundings"), our approach uses the FoaF network of each user as the basis of clustering approach that places her peers in overlapping communities.

4 Non-negative matrix factorization

In the following sections, the basic components of the proposed methodology are going to be analyzed. The first is Non-negative Matrix Factorization, a member of the broader family of *dimensionality reduction* techniques (Alpaydin 2014), that try to obtain a parts-based representation of high dimensional data by projecting them to a lower dimensional space. Compared to other similar techniques (singular value decomposition, principal component analysis), NMF imposes the non-negativity constraint on the resulting matrices, allowing for a better interpretability of the results (Wang and Zhang 2013).

In our case, let $A \in \mathbb{R}_{+}^{n \times n}$ be the adjacency matrix of a social network of n users. It is decomposed to the non-negative *basis matrix* $W \in \mathbb{R}_{+}^{n \times r}$ and to the non-negative *coefficient matrix* $H \in \mathbb{R}_{+}^{r \times n}$, so that

$$A \approx \widetilde{A} \equiv WH \tag{3}$$

where r is the number of communities and $r << n$.

Every row of the $n \times r$ basis matrix W represents the participation of the respective user in each of the r communities (hence the soft-clustering approach). Every column of the $r \times n$ coefficient matrix H corresponds to the contribution of the respective user in the formation of each of the r communities. The similarity between users u_i and u_j stems from the inner product of the i-th row vector of W times the j-th column vector of H

$$\text{sim}(u_i, u_j) = \mathbf{w}_i^T \mathbf{h}_j \tag{4}$$

The NMF algorithm tries to compute the elements of the non-negative matrices W, H so that their product is as "close" to A as possible, measured by a *distance function* $\mathcal{D}$. More formally, NMF is the *convex optimization* problem of Eq. 5 (here described as a minimization problem)

$$\begin{aligned} &\min_{W,H} \mathcal{D}(A||WH) \\ &\text{subject to } W \geq 0, H \geq 0 \end{aligned} \tag{5}$$

In general, minimizing $\mathcal{D}$ is an NP-hard problem for which no known convex formulations in both W and H exist. However, $\mathcal{D}$ is individually convex in W and H and therefore most NMF algorithms iteratively and interchangeably adjust the aforementioned matrices (Lee and Seung 1999).

4.1 Bayesian NMF

Bayesian NMF (Psorakis et al. 2011) is a special instance of *probabilistic* NMF (Lee and Seung 2000) that approximates the parameters W, H by employing the classic formula of Bayesian Inference

$$P(W, H, \Theta|A) \propto P(A|W, H, \Theta)P(W, H|\Theta)P(\Theta) \tag{6}$$

where Θ is the hyper-parameter space. The left-hand side of Eq. 6 is the unknown *a-posteriori* distribution of the model parameters conditioned on the known data. The right-hand side of the same equation offers an approximation to the unknown distribution; that is the a-posteriori distribution is proportional to the product of the *likelihood* of the data times the *a-priori* distribution of the model parameters.

Careful choice of the likelihood function and the a-priori distributions can lead to an algorithm that exhibits faster and better convergence. The most common choice for the likelihood function is either the *Normal* or the *Poisson* distributions and a *conjugate prior* for those likelihood functions is selected for the a-priori distributions, such as the Normal, Gamma, Inverse Gamma and Wishart distributions.

4.2 The a-priori distribution

The a-priori distributions mentioned above qualify as a good choice, given all the other assumptions made so far. However, they do not capture the fact that in this specific case, the matrix to be factored represents a FoaF graph, whose edges obey specific structural properties. Instead, most NMF algorithms examine each element (edge) of the adjacency matrix independently of the others, as if there is no correlation in-between them, enforced by the network itself.

Since the role of the a-priori distribution in the Bayesian Inference is to capture this kind of information, it is important to vary its choice in a direction that would incorporate global structural properties of the graph. Ideally, this course of action is expected to balance the effects of the local structural properties of the graph imposed by the likelihood function. Therefore, in addition to the application of the Bayesian NMF to the adjacency matrix of the FoaF network of each user, we propose to statistically model the social graph via the use of the *exponential random graph models* (ERGM).

5 Exponential random graph models

Exponential random graphs are a category of *ensemble models* that consist of all the possible *simple graphs* without self-loops on n vertices (Robins et al. 2007). For every graph G in the model $\mathcal{G}$ a probability (or a *configuration*) $P(G)$ is defined as

$$P(G) = \frac{1}{Z}\mathrm{e}^{H(G)}, \quad Z = \sum_{G \in \mathcal{G}} \mathrm{e}^{H(G)} \tag{7}$$

where Z denotes the *partition function*; it is equal to the sum of all possible configurations, ensuring in this way that $P(G)$ takes the form of a distribution. The properties of the model are encoded in the *Graph Hamiltonian* $H(G)$

$$H(G) = \sum_{i=1}^{r} \theta_i x_i(G) \tag{8}$$

where the parameters θ_i specify the influence of each *network observable* $x_i(G)$ in the overall model. Network observables are properties of the graph, such as the number of simple edges, reciprocal edges, stars, triangles etc. A network observable can be turned into a *network statistic* by computing its expected value as follows

$$\langle x_i \rangle = \sum_{G \in \mathcal{G}} x_i(G) P(G) = \frac{1}{Z} \left(\sum_{G \in \mathcal{G}} x_i(G) \exp \left\{ \sum_{i=1}^{r} \theta_i x_i(G) \right\} \right) \Rightarrow$$
$$\langle x_i \rangle = \frac{1}{Z} \left(\frac{\partial}{\partial \theta_i} \sum_{G \in \mathcal{G}} \exp \left\{ \sum_{i=1}^{r} \theta_i x_i(G) \right\} \right) = \frac{1}{Z} \frac{\partial Z}{\partial \theta_i} = \frac{\partial F}{\partial \theta_i} \tag{9}$$

where $F \equiv \ln Z$ is the *free energy* of the model (Newman 2010, pp. 569–570). In the general case, an analytic computation of Z is impossible; in simple models, however, such a computation is indeed possible (Park and Newman 2004b).

5.1 Bernoulli graphs

The simplest ERGM is the *Bernoulli* graph or the *Erdős–Rèyni* model. It only takes into account the expected number of edges $\langle m \rangle$ in the network. In this case, the graph Hamiltonian is expressed by the following analytical form (for undirected graphs)

$$H(G) = \theta m(G), \quad m(G) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \tag{10}$$

where $m(G)$ is the network observable for the total number of the edges of the graph. The partition function is equal to

$$Z = \left[1 + \mathrm{e}^{\theta} \right]^{\binom{n}{2}} \tag{11}$$

If we know $\langle m \rangle$, we may as well compute the value of the parameter θ

$$\langle m \rangle = \frac{1}{Z} \frac{\partial Z}{\partial \theta} = \binom{n}{2} \frac{1}{1 + \mathrm{e}^{\theta}} \Rightarrow \theta = \ln \frac{\langle m \rangle}{\binom{n}{2} - \langle m \rangle} \tag{12}$$

and therefore fully define the model.

Other ERGMs that may be analytically computed include the *generalized random graphs*, where the network observable being modeled is the degree distribution of the nodes, and the *reciprocity model*, where the network observables are both the number of edges and the number of the reciprocal edges. More complex models cannot be analytically solved and therefore one has to resort to approximation techniques in order to find a solution (Sect. 6.1; Appendix 1; Park and Newman 2004b)

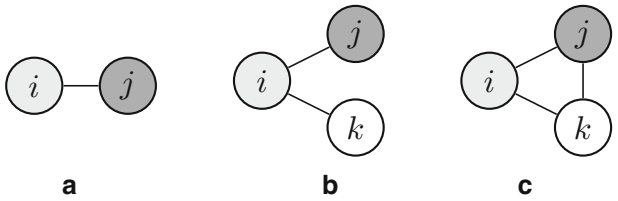

Fig. 2 Basic network observables of undirected graphs: **a** Edge, **b** 2-star and **c** triangle

6 Bayesian NMF and ERGM

In the previous sections, the methodological novelty of introducing ERGMs as the a-priori distribution for the Bayesian NMF has been outlined. Starting from Eq. 6, we are going to further elaborate on the proper choice of the a-priori distribution (Sect. 6.1) and the likelihood function (Sect. 6.2).

6.1 Fixing the a-priori distribution

We begin our reasoning by observing that, since we are going to apply NMF to a FoaF network, the induced subgraph is expected to have a few triangles and cliques but a large number of *stars* ("open" triangles). Those stars are going to be modeled in their simplest form, the *2-star* model (Fig. 2b).

The Hamiltonian of this model is defined as follows

$$\begin{aligned} H &= \theta m(G) + \tau s(G) \\ m(G) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}, &\quad s(G) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} \sum_{k=1(\neq j)}^{n-1} a_{ik} \end{aligned} \tag{13}$$

where the term $m(G)$ is equal to number of the edges in the graph (whose influence is regularized by the hyper-parameter θ) and the term $s(G)$ is equal to the number of the open triangles (whose influence is regularized by the hyper-parameter τ respectively).

Unfortunately, no analytical solution exists for the ERGM described by the Hamiltonian of Eq. 13 and therefore we have to resort to approximation techniques in order to estimate the hyper-parameters θ and τ. Ideally, we would like to "merge" the aforementioned (lowercase) hyper-parameters into a single hyper-parameter bound on the number of edges in the graph ($m(G)$). In other words, it would be desirable to express the Hamiltonian of the 2-star model (Eq. 13) in a form that resembles the Bernoulli Graphs (Eq. 11) for which an analytic solution exists (Sect. 5.1).

As a first step, the network statistics of the Hamiltonian of Eq. 13 are re-written into a form that makes them dependent on the degree k_i of each node (instead of the edges between nodes)

$$m(G) = \frac{1}{2}\sum_{i=1}^{n} k_i, \quad s(G) = \frac{1}{2}\sum_{i=1}^{n} k_i(k_i - 1) = \frac{1}{2}\sum_{i=1}^{n} k_i^2 - m(G)$$

Then, the Hamiltonian of Eq. 13 becomes

$$H = \theta m(G) + \tau s(G) = \frac{\tau}{2} \sum_{i=1}^{n} k_i^2 + \frac{\theta - \tau}{2} \sum_{i=1}^{n} k_i \tag{14}$$

In order to ease the calculations that follow, hyper-parameters θ and τ are substituted by the complementary hyper-parameters J and B, which are defined as

$$J = \frac{(n-1)\tau}{2}, \quad B = \frac{\theta - \tau}{2} \tag{15}$$

and the Hamiltonian of Eq. 14 takes its final form

$$H = \frac{J}{n-1} \sum_{i=1}^{n} (k_i)^2 + B \sum_{i=1}^{n} k_i, \tag{16}$$

Comparing Eqs. 13–16, it may be argued that both Hamiltonians describe the same ERGM, the 2-star model, using different network statistics; Eq. 13 uses the number of edges (regularized by hyper parameter θ) and number of open triangles (hyper parameter τ) while Eq. 16 uses the sum of the squares of node degrees (regularized by hyper parameter $\frac{J}{n-1}$) and the sum of node degrees (hyper parameter B). Both representations are totally equivalent except for one difference; the latter (Eq. 16) may be approximated by the application of non-perturbative techniques from the statistical mechanics of networks, using the *mean-field theory* (Park and Newman 2004b).

Mean-Field Theory Inspired by an analogous technique of Park and Newman (2004a), we derive a solution for the ERGM described by the Hamiltonian of Eq. 16 that is based on the Hubbard-Stratonovich transformation and a saddle-point expansion around the most probable solution. The free energy of the 2-star model (Sect. 5) is approximated by the following equation (Appendix 1)

$$\begin{aligned} F = &-n(n-1)J(\phi_0)^2 + \frac{1}{2}n(n-1)\ln\left(1 + \mathrm{e}^{4J\phi_0 + 2B}\right) \\ &+ \frac{n}{2}\ln[(n-1)J] - \frac{n}{2}\ln 4\pi \end{aligned} \tag{17}$$

where ϕ_0 is the mean-field solution defined by the equation

$$\phi_0 = \frac{1}{2}\left[\tanh(2J\phi_0 + B) + 1\right] \tag{18}$$

Eq. 18 above demonstrates a direct relationship between the two hyper parameters of the Hamiltonian of model, an observation that will be exploited in the forthcoming computations.

Following the reasoning of Sect. 5, the partial derivative of the free energy with respect to the hyper parameter B is equal to the expected value of the sum of the degrees of the nodes

$$\left\langle \sum_{i=1}^{n} k_i \right\rangle = \frac{\partial F}{\partial B} \Rightarrow \sum_{i=1}^{n} \langle k_i \rangle = \frac{\partial F}{\partial B} \tag{19}$$

Approximating $\langle k_i \rangle$ by $\langle k \rangle$ (its most probable/expected value) and plugging it in Eq. 19, yields

$$\sum_{i=1}^{n} \langle k_i \rangle = n\langle k \rangle = \frac{\partial F}{\partial B} \Rightarrow \langle k \rangle = \frac{1}{n} \frac{\partial F}{\partial B} = (n-1)\phi_0 \tag{20}$$

making ϕ_0 equal to

$$\phi_0 = \frac{\langle k \rangle}{n-1}, \quad \phi_0 \in (0, 1) \tag{21}$$

From Eq. 21, it is deduced that ϕ_0, the mean-field solution of Eq. 17 is directly related to the expected node degree of the network. Park and Newman (2004a) term this quantity as the *connectance* of the network and further observe that it expresses the degree to which the underlying graph is a *clique* (every node of a n-clique has a degree of $n-1$).

In a similar manner, the partial derivative of the free energy with respect to the hyper parameter $\frac{J}{n-1}$ is equal to the expected value of the sum of the squares of node degrees

$$\left\langle \sum_{i=1}^{n} k_i^2 \right\rangle = \sum_{i=1}^{n} \langle k_i^2 \rangle = \frac{\partial F}{\partial (\frac{J}{n-1})}$$

Once again, approximating $\langle k_i^2 \rangle$ by $\langle k^2 \rangle$ (its most probable/expected value), yields

$$n\langle k^2 \rangle = (n-1)\frac{\partial F}{\partial J} \Rightarrow \langle k^2 \rangle = \frac{n-1}{n}\frac{\partial F}{\partial J}$$

and after a series of computations, in conjunction with Eq. 21, the hyper-parameter J is found to be equal to

$$J = \frac{(n-1)}{2(\langle k^2 \rangle + \langle k \rangle^2 - 2(n-1)\langle k \rangle)} \tag{22}$$

Combining Eqs. 18, 22, a similar relationship for hyper-parameter B arises

$$B = \frac{1}{2} \ln \phi_0 - \frac{1}{\langle k^2 \rangle + \langle k \rangle^2 - 2(n-1)\langle k \rangle} \tag{23}$$

and the 2-star model is fully approximated under the mean-field assumption.

Taking the initial Hamiltonian of the solvable 2-star model (Eq. 16), it may be observed that the following relationship holds true

$$H = \frac{J}{n-1}\sum_{i=1}^{n} k_i^2 + B\sum_{i=1}^{n} k_i = \frac{\partial F}{\partial J}J + \frac{\partial F}{\partial B}B = \nabla F$$

In other words, the Hamiltonian of the model is equated to the gradient of its free energy. Replacing the hyper parameters B, J in the above equation with their counterparts of Eqs. 22, 23 we arrive at the following equation

$$H = -n(n-1)J(\phi_0)^2 + 2n(n-1)J\phi_0 + \frac{n}{2} + n(n-1)B\phi_0$$

which in turn, after a series of computations and simplifications, yields to

$$H = \Theta m(G), \qquad \Theta = \ln\phi_0 + 2(2-3\phi_0)J \tag{24}$$

This is a pivotal point since, thus far, two goals have been achieved; an approximate solution to the 2-star model has been found, as well as an expression (Eq. 24) that allows its easy integration into the factorization algorithm, as it shall be demonstrated next.

6.2 The likelihood function

So far, the global graph properties have been examined and they have been encoded into the a-priori distribution, by the application of the exponential random graph models. The likelihood function, on the other hand, models local edge properties. As it has already been mentioned in Sect. 4.1, the two most-common probability distributions used as likelihood functions are the *Normal* and *Poisson* distributions. Both make the assumption that each edge in the network $(i, j) \in E$ is created independently of all the others, or equivalently, that each element a_{ij} of the adjacency matrix A assumes a value independently of all the other elements.

More specifically, the Normal Distribution considers the probability of edge creation between nodes i and j to originate from a Gaussian source with mean value $\mu = a_{ij}$ and variance σ, while the Poisson distribution considers the probability of edge creation between nodes i and j to originate from a Poisson source with rate $\lambda = a_{ij}$. When both distributions are used for probabilistic NMF in the Bayesian context, then it can be proven that the normal distribution optimizes the Euclidean distance between A and WH while the Poisson distribution optimizes the generalized Kullback–Leibler divergence (Lee and Seung 2000):

$$\mathcal{D}(a_{ij}||\widetilde{a_{ij}}) = a_{ij}\ln\frac{a_{ij}}{\widetilde{a_{ij}}} - a_{ij} + \widetilde{a_{ij}} \tag{25}$$

In our case, the form of the a-priori distribution (Eq. 24) dictates the use of the Poisson distribution as the likelihood function because the necessary computations are simplified a lot, as it will be presented in the following section.

7 Algorithm

Having reasoned on the choice of the likelihood and the a-priori functions in the previous sections, we now proceed to develop the algorithm for the incorporation of the ERGM into the probabilistic NMF. We will start with the standard multiplicative update rules of Lee and Seung (2000) for the Poisson distribution as the likelihood function

$$W \leftarrow W \circ \frac{\frac{A}{WH} H^\top}{E H^\top} \tag{26}$$

$$H \leftarrow H \circ \frac{W^\top \frac{A}{WH}}{W^\top E} \tag{27}$$

where $\circ$ symbolizes the Hadamard (element-by-element) matrix multiplication and E the $n \times n$ matrix whose elements are all equal to one.

Plugging in the a-priori distribution in the probabilistic NMF alters the multiplicative update rules of Eqs. 26, 27 as follows (Appendix 2)

$$W \leftarrow W \circ \frac{\frac{A}{WH} H^\top}{(1-\Theta) E H^\top} \tag{28}$$

$$H \leftarrow H \circ \frac{W^\top \frac{A}{WH}}{(1-\Theta) W^\top E} \tag{29}$$

Comparing the set of Eqs. 26, 27 with Eqs. 28, 29 it is easily observable that the factorization process for both matrices is adjusted by a factor of

$$\frac{1}{1-\Theta} \tag{30}$$

which encodes the contribution of the a-priori distribution (that models the global graph properties) in the factorization process.

In order for the NMF to be valid, the factor of Eq. 30 has to be non-negative as well, which yields the following constraint on the parameter Θ (combined with Eq. 24)

$$\frac{1}{1-\Theta} > 0 \Rightarrow \Theta < 1 \Rightarrow \ln \phi_0 + 2(2 - 3\phi_0) J < 1 \tag{31}$$

The above inequality holds true for social network graphs because they are far from being cliques and therefore their connectance (Eq. 21) is close to zero ($\phi_0 << 1$). As a result, the first term of the sum of Eq. 31 assumes very low negative values that counter the effect of the small positive second term (hyper-parameter J is positive due to the substitution of Eq. 36).

Table 1 Datasets used in experiments

	`lastfm-2k` (Cantador et al. 2011)	`flixster` (Jamali 2010)
Users	1.8k	147k
Items	17.6k	48.7k
Evaluations	92.8k	8.2M
Evaluation density	0.278%	0.114%
Edges	25.4k	7M
Edge density	1.42%	0.06%
Maximum degree	119	1045
Average degree	13	15

8 Experiments and evaluation

8.1 Datasets

Two social recommender datasets (consisting of evaluation data and an undirected social network of users) have been selected in order to conduct the experiments; `lastfm-2k` (Cantador et al. 2011) and `flixster` (Jamali 2010). Both datasets were collected out of the respective services and their properties are summarized in Table 1. The first is a small dataset containing usage data (the number of times each user played a specific music track) while the second one is a medium-sized dataset that consists of movie ratings on the 5-star scale. Despite their different nature, they are very sparse and they also exhibit the characteristics of scale-free networks both in terms of the evaluations they contain and their node-degree distributions (a few nodes are attached to many edges while the overwhelming majority of them are incident to very few nodes). The latter could also be deduced from the difference between the maximum and average node degree on the last two rows of the table.

8.2 Experimental protocol

The applied experimentation protocol has been the leave-one-out cross validation on the available evaluation data (Table 1). At each iteration of the protocol, a user is extracted from the dataset and his evaluations are split in two distinct sets, training and test. Then, the training set is put back into the dataset, which in turn is provided to the recommender system. In the following step, predictions are performed by the algorithm and a list of items is returned as output, which is compared to the list in the test set, on the evaluation metrics described in the following subsection. This procedure is repeated for 5 times, for recommendation lists ranging from 5 to 25 items. In order to ensure that the recommendations are meaningful, at each repetition of the experimentation protocol, only the users who had at least twice as many ratings as the list size are selected. The results appearing on Figs. 3 and 4 are averaged.

A number of memory-based collaborative filtering algorithms have been implemented in order to evaluate the quality of the produced recommendations in different

settings. The baseline system is the pure collaborative filtering approach, where user similarity is estimated solely on their evaluations, without examining their social ties. The similarity function used in this case is the *log likelihood* of the provided data, because it exhibited the best performance on the evaluation metrics (Sect. 8.3) when compared to other similarity indices (i.e. cosine or Pearson correlation).

Following, a number of social similarity indices that extend to the local vicinity of each user were examined, based on the systems outlined in Sect. 3.1. *MoleTrust-1* considers as similar those users who share a tie with the target user in the social network (rows/columns of the adjacency matrix A), while *MoleTrust-2* extends this notion of similarity to those other users that are incident to the neighbors of the target user (FoaF network) as well (rows/columns of A^2), removing any cycles if present. Finally, *TidalTrust* takes into account the shortest path between the target user and each other user in the same connected component of the social network (again, by directly manipulating the rows and columns of the adjacency matrix A).

The implementation of a global exploration approach that computes the reputation of each user has also been included in the experiments. The algorithm of choice has been the *TrustWalker* system (Sect. 3.2) that performs a random walk on the social network, hopping to the next neighbor uniformly at random. The random walk originates from the target user and when it reaches its stationary distribution those nodes that are most probable to be visited are returned as the most similar users. In addition to the aforementioned algorithms, two model-based social collaborative filtering systems have been included in the experimental procedure; the *SoRec* system and the *STE* model (Sect. 3.2).

Finally, the proposed methodology has also been tested in two different configurations in order to examine the relative performance of the ERGMs as the a-priori distribution. The first configuration included the *probabilistic NMF* approach based on the Poisson distribution as the likelihood function (Eqs. 26, 27) while the second configuration has been extended to include the *Exponential Random Graph Models* as the a-priori distribution (28, 29), using *Alternating Least Squares* (ALS) as the training method (Hu et al. 2008). In order to be fair in our comparisons, all four model-based methods (*SoRec*, *STE*, *NMF* and *NMF–ERGM*) were evaluated on the adjacency matrices of the subgraphs that correspond to the FoaF networks of each target user.

At this point it should also be noted that even though the leave-one-out cross validation is not chosen very often for evaluating model-based systems, it is not unusual (Ma et al. 2009; Weimer et al. 2007; Shi et al. 2010). Since, in general, recommender system and social network datasets are very sparse, it is desirable to be able to use as much data as possible in the training phase. More specifically, in our approach, we do not factor the whole social network at each iteration, but rather the induced adjacency matrix of the FoaF network of each target user, which is of considerable smaller size, thus allowing the implementation of the leave-one-out cross-validation methodology.

The *User Filtering* method used in all systems outlined above is the `NearestN` formula, which means that only the N "closest" users to the target user are selected. After validation, the value of hyper-parameter N was set to 5. For lower values, system performance became unstable while for larger values it clearly deteriorated. However, the relative performance and the ordering of all systems (as depicted in Figs. 3, 4) was not influenced by the value of N.

8.3 Evaluation metrics

We have evaluated the performance of the algorithms into a set of three metrics that fall in two groups; the first group (*Precision* and *Recall*) measures the accuracy of the recommendations (de Wit 2008) while the second (*nDCG*) examines the order of the recommended items in the list.

Given that a recommender system returns a list of N items for a specific target user, Precision measures the proportion of the relevant items selected in the list (N_{rs}) versus the total number of the selected items (N_s) on the percentage scale

$$\text{Precision} = \frac{N_{rs}}{N_s}(\%)$$

Recall, on the other hand, expresses the proportion of the relevant items selected in the list over the total number of relevant items (N_r) that exist in the test set for the said user (again, on the percentage scale)

$$\text{Recall} = \frac{N_{rs}}{N_r}(\%)$$

The last metric is the *Normalized Discounted Cumulative Gain* (nDCG), which measures the rank accuracy of the recommendation list. It is defined as the ratio of the Discounted Cumulative Gain over the Idealized Discounted Cumulative Gain (IDCG)

$$\text{nDCG}_N = \frac{\text{DCG}_N}{\text{IDCG}_N}$$

where DCG is set to be equal to

$$\text{DCG}_N = \sum_{i=1}^{N} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

and rel_i is a binary indicator which is equal to one if the i-th element of the list is relevant to the user (zero, otherwise). The idealized DCG (IDCG) at position i designates the magnitude of DCG, had all items in the recommendation list been relevant to the target user. The ratio of DCG over IDCG (and thus, the metric nDCG) is estimated in order to be able to directly compare lists of different lengths, as in our case.

Given the above definitions, a recommender system that exhibits higher Precision, Recall and nDCG is considered to be "better", i.e. more more accurate, when compared to a second one that outputs lower values in the aforementioned metrics.

9 Results

Figures 3 and 4 summarize the results on the evaluation metrics, on both datasets. A first observation is that all algorithms achieve lower results on the `flixster` dataset.

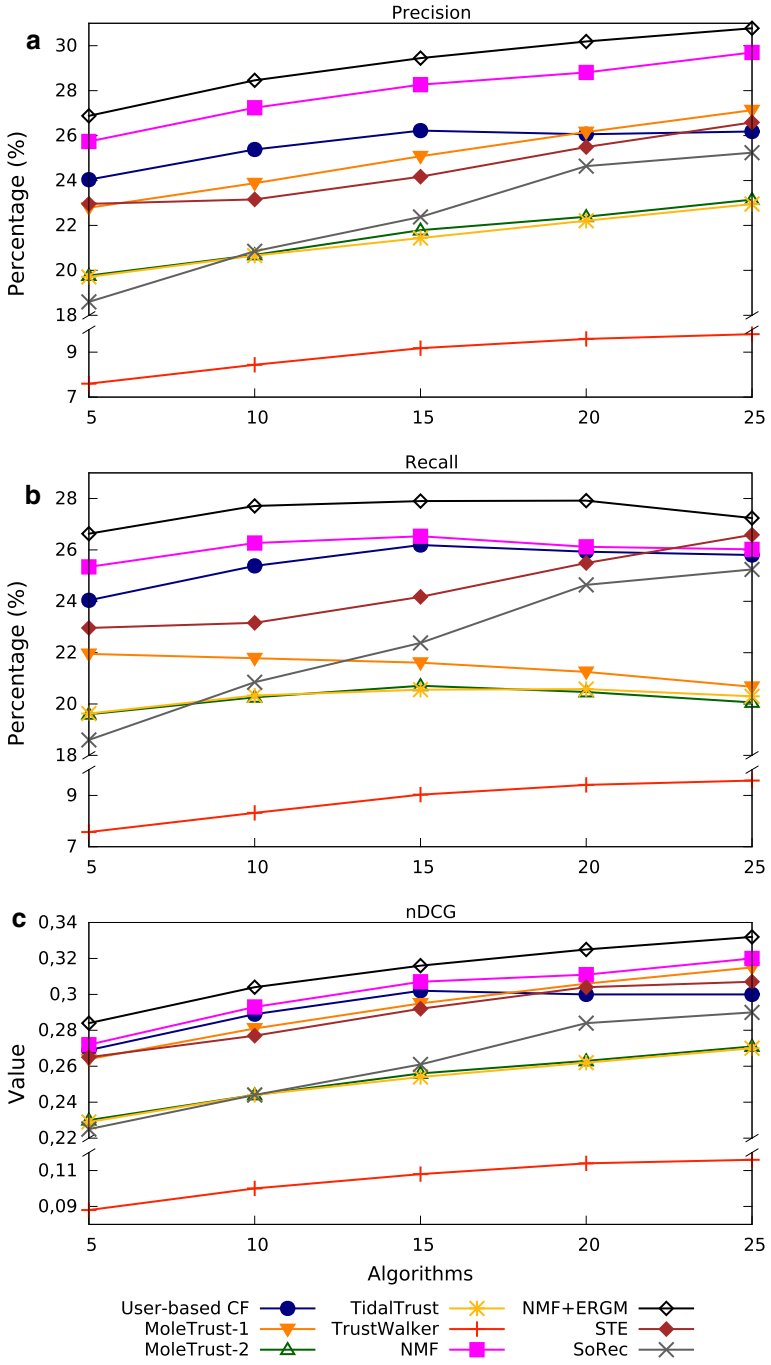

Fig. 3 Evaluation results on the lastfm-2k dataset

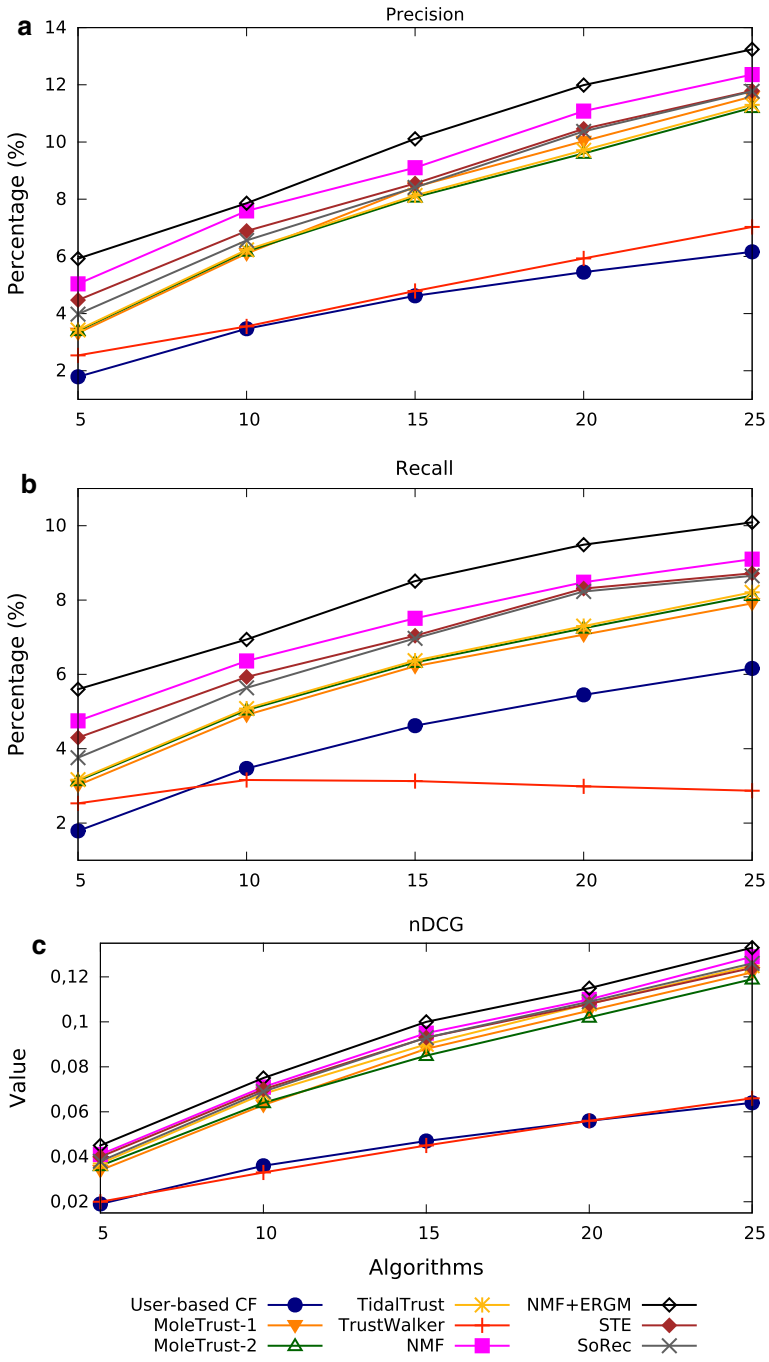

Fig. 4 Evaluation results on the flixster dataset

This behavior is attributed to two factors; the information *sparsity* and the *nature* of the datasets. The `lastfm-2k` dataset is twice as dense than the `flixster` dataset and apart from that it consists of usage data. Therefore, any user-to-item evaluation is considered relevant in the experimentation protocol, regardless of the number of the times each track had been accessed (Sect. 8.1). On the other hand, the `flixster` dataset contains movie ratings on the 5-star scale and only movies with 3 stars and above are considered to be relevant for recommendation (around 63% of the dataset). As a result, the `flixster` dataset appears to be about one fourth as dense as the `lastfm-2k`. An interesting observation is that this ratio on density is reflected almost linearly on the metrics of all algorithms except for the *TrustWalker* system.

Another important remark is that in sparse datasets, the ego-network of a specific user is almost as good as an information source for recommendation as the pure collaborative-filtering approach. This is evident when comparing the performance of the *User-based CF* and the *MoleTrust-1* algorithms in the `lastfm-2k` dataset; even though the former outperforms the latter, the lead is slight and it cannot be considered to be a clear indication of the superiority of this method over the other. This is not the case, though, for the algorithms that aggregate all ties in the FoaF network (and beyond) of each user. *MoleTrust-2* and *TidalTrust* clearly fall behind in all metrics in the same dataset, an indication that in most cases the FoaF network contains "noise" that needs to be filtered out in order to make meaningful recommendations.

When data sparsity arises, however, the efficacy of the *User-based CF* approach naturally deteriorates, as there are even fewer users whose evaluations coincide. It is in this scenario that the social filtering approaches reveal their full potential. Results on the `flixster` dataset (Fig. 4) clearly demonstrate the superiority of the local social filtering methods on all recommendation metrics; it is also noteworthy that in this case all three local approaches display similar performance results on the recommendation metrics, even though they explore the vicinity of each user in the social network to different extends.

On the other hand, the *TrustWalker* algorithm performs poorly on both datasets; it clearly falls behind in the `lastfm-2k` dataset and it is on the same level as the least effective algorithm in the `flixster` dataset. Another interesting observation is that the *TrustWalker* algorithm is the only of the systems implemented to display the same levels of efficiency in both datasets, irrespective of their individual density. The aforementioned reasoning leads to the conclusion that basing predictions solely on the most popular (or most visited in the random walk) nodes of the social network does not always guarantee a similarity in taste. However, this is not the case for the model-based social collaborative filtering systems (*SoRec* and *STE*) which exhibit a satisfactory performance level on both datasets.

Finally, the methodologies proposed in this paper (*NMF* and *NMF + ERGM*) seem to achieve better results than both the local and the global approaches. Indeed, the soft clustering of the nodes in a user's FoaF network allows for a more thorough analysis of user proximity in the network, especially when compared to the basic assumptions *MoleTrust-2* and *TidalTrust* make. This analysis is further enhanced with the addition of the exponential random graph models as the a-priori distribution; extending the edge centric logic of the probabilistic NMF approach through the inclusion of structural graph information leads to a visible performance boost.

10 Conclusion and future work

In this paper, we have proposed a novel memory-based, social collaborative filtering recommender system. We have applied Bayesian NMF in a new context; as a tool for estimating user similarity in FoaF networks instead of using it to uncover latent user and item factors (like certain model-based approaches do). On top of that, the algorithm has been further fine-tuned to deal with the fact that the matrix being factorized contains edges that pertain to a graph, an object that possesses specific structural properties. The introduction of the ERGMs as the a-priori distribution alleviated the edge-centric logic of the aforementioned procedure and allowed for the modeling of more general network observables, such as the number of edges and 2-stars.

The obtained results show a steady improvement over all evaluation metrics when compared to the other approaches; the *User-based CF* algorithm and the social filtering algorithms that either examine a vicinity around each user or process the whole social graph. This behavior is attributed to the filtering capabilities of the proposed methodology on the FoaF levels; instead of just accumulating either too many or too few users, it tries to uncover patterns in their social ties by soft-clustering them into areas that resemble the way overlapping communities are discovered.

Nevertheless, there is still room for improvement. An obvious course of action would be to extend the proposed methodology to Social Recommender Systems that contain directed relationships between their members. Another possible research direction would be to incorporate more complex network observables in the factorization process and, more specifically, triangles. In this case, however, the computations of the model become very complicated and it is not easy to derive an approximate solution like that of Sect. 6.1. Park and Newman (2005) provide insight in this field, however a more generalized approach to parameter estimation might yield better results.

Appendix 1: Approximating the free energy of the 2-star model

The analysis that follows aims to show how the exact form of the free energy of the 2-star model is derived. Starting with the Hamiltonian of the aforementioned model (Eqs. 16 and 32 below)

$$H = \frac{J}{n-1}\sum_{i=1}^{n}(k_i)^2 + B\sum_{i=1}^{n}k_i \tag{32}$$

the partition function Z becomes

$$\begin{aligned} Z &= \sum_{G\in\mathcal{G}} \mathrm{e}^{H(G)} = \sum_{G\in\mathcal{G}} \exp\left\{\frac{J}{n-1}\sum_{i=1}^{n}(k_i)^2 + B\sum_{i=1}^{n}k_i\right\} \\ &= \sum_{G\in\mathcal{G}} \exp\left\{\frac{J}{n-1}\sum_{i=1}^{n}(k_i)^2\right\} \times \exp\left\{B\sum_{i=1}^{n}k_i\right\} \end{aligned} \tag{33}$$

According to Park and Newman (2004b), sums like that of Eq. 33 (containing terms of the form of e^{k^2}) are encountered in the study of interacting quantum systems and may be calculated through the application of the Hubbard–Stratonovich transformation (Hubbard 1959). The said transformation is used for the conversion of a particle theory (in this case, the node degree k) to the corresponding field theory, through the introduction of an auxiliary scalar field (in this case, ϕ_i, as it will be demonstrated next).

The Hubbard-Stratonovich transformation is based upon the Gaussian Integrals, that may be written in any of the following two forms

$$\int_{-\infty}^{\infty} e^{-\alpha x^2} dx = \sqrt{\frac{\pi}{\alpha}} \tag{34}$$

$$\int_{-\infty}^{\infty} e^{-ax^2+bx+c}\, dx = \sqrt{\frac{\pi}{a}}\, e^{\frac{b^2}{4a}+c} \tag{35}$$

with the real constant α taking non-negative values ($\alpha > 0$). Substituting α and x according to Eq. 36 below

$$\alpha \leftarrow (n-1)J, \quad x \leftarrow \phi_i - \frac{k_i}{n-1} \tag{36}$$

and subsequently plugging them into the Gaussian Integral of Eq. 34, yields

$$\begin{aligned}
&\int_{-\infty}^{\infty} \exp\left\{-(n-1)J\left(\phi_i - \frac{k_i}{n-1}\right)^2\right\} d\left(\phi_i - \frac{k_i}{n-1}\right) = \sqrt{\frac{\pi}{(n-1)J}} \Rightarrow \\
&\int_{-\infty}^{\infty} \exp\left\{-(n-1)J\phi_i^2 + 2Jk_i\phi_i - \frac{Jk_i^2}{n-1}\right\} d\left(\phi_i - \frac{k_i}{n-1}\right) = \sqrt{\frac{\pi}{(n-1)J}} \Rightarrow \\
&e^{-\frac{Jk_i^2}{n-1}} \times \int_{-\infty}^{\infty} \exp\left\{-(n-1)J\phi_i^2 + 2Jk_i\phi_i\right\} d\phi_i - \\
&\quad -\frac{1}{n-1} e^{-(n-1)J\phi_i^2} \times \int_{-\infty}^{\infty} \exp\left\{2Jk_i\phi_i - \frac{Jk_i^2}{n-1}\right\} dk_i = \sqrt{\frac{\pi}{(n-1)J}}
\end{aligned} \tag{37}$$

In order to eliminate the integral in Eq. 37, the second form of the Gaussian Integral is being used (Eq. 35). Substituting the constants α, b, c and the unknown variable x according to Eq. 38 below

$$a \leftarrow \frac{J}{n-1}, \quad b \leftarrow 2J\phi_i, \quad c \leftarrow 0, \quad x \leftarrow k_i \tag{38}$$

yields

$$\int_{-\infty}^{\infty} \exp\left\{2Jk_i\phi_i - \frac{Jk_i^2}{n-1}\right\} dk_i = \sqrt{\frac{(n-1)\pi}{J}} e^{(n-1)J\phi_i^2} \tag{39}$$

The second term of the left-hand side of Eq. 37 in conjunction with Eq. 39, becomes

$$-\frac{1}{n-1}\mathrm{e}^{-(n-1)J\phi_i^2}\sqrt{\frac{(n-1)\pi}{J}}\mathrm{e}^{(n-1)J\phi_i^2} = -\sqrt{\frac{\pi}{(n-1)J}} \tag{40}$$

and Eq. 37 in conjunction with Eq. 40 becomes

$$\begin{aligned}
&\mathrm{e}^{-\frac{Jk_i^2}{n-1}} \times \int_{-\infty}^{\infty} \exp\left\{-(n-1)J\phi_i^2 + 2Jk_i\phi_i\right\} \mathrm{d}\phi_i - \sqrt{\frac{\pi}{(n-1)J}} = \sqrt{\frac{\pi}{(n-1)J}} \Rightarrow \\
&\mathrm{e}^{-\frac{Jk_i^2}{n-1}} \times \int_{-\infty}^{\infty} \exp\left\{-(n-1)J\phi_i^2 + 2Jk_i\phi_i\right\} \mathrm{d}\phi_i = 2\sqrt{\frac{\pi}{(n-1)J}} \Rightarrow \\
&\mathrm{e}^{\frac{J}{n-1}k_i^2} = \frac{1}{2}\sqrt{\frac{(n-1)J}{\pi}} \times \int_{-\infty}^{\infty} \exp\left\{-(n-1)J\phi_i^2 + 2Jk_i\phi_i\right\} \mathrm{d}\phi_i
\end{aligned} \tag{41}$$

Taking the product of the left-hand side of Eq. 41 for all n nodes of the graph and making use of the property of iterated integrals for multiple-variable functions, yields

$$\begin{aligned}
&\exp\left\{\frac{J}{n-1}\sum_{i=1}^{n}(k_i)^2\right\} = \left[\frac{(n-1)J}{4\pi}\right]^{\frac{n}{2}} \times \prod_{i=1}^{n}\int_{-\infty}^{\infty} \exp\left\{-(n-1)J\phi_i^2 + 2Jk_i\phi_i\right\} \\
&\quad \mathrm{d}\phi_i \Rightarrow \\
&\exp\left\{\frac{J}{n-1}\sum_{i=1}^{n}(k_i)^2\right\} = \left[\frac{(n-1)J}{4\pi}\right]^{\frac{n}{2}} \times \\
&\quad \times \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \exp\left\{-(n-1)J\sum_{i=1}^{n}(\phi_i)^2 + 2J\sum_{i=1}^{n}k_i\phi_i\right\} \mathrm{d}\phi_1 \ldots \mathrm{d}\phi_n
\end{aligned} \tag{42}$$

The integral of the right-hand side of Eq. 42 is once again calculated based on quantum mechanics (Park and Newman 2004a). More specifically, every ϕ_i is thought to symbolize the contribution of a respective field during the movement of a particle in an one-dimensional system. Consequently, the effect of the overall field in the particle movement is approximated by the superposition of each individual field ϕ_i, which is mathematically formulated as a path integral (Eq. 43)

$$\begin{aligned}
&\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \exp\left\{-(n-1)J\sum_{i=1}^{n}(\phi_i)^2 + 2J\sum_{i=1}^{n}k_i\phi_i\right\} \mathrm{d}\phi_1 \ldots \mathrm{d}\phi_n \\
&\quad = \int_{-\infty}^{\infty} \exp\left\{-(n-1)J\sum_{i=1}^{n}(\phi_i)^2 + 2J\sum_{i=1}^{n}k_i\phi_i\right\} \mathscr{D}\phi
\end{aligned} \tag{43}$$

As a consequence, Eq. 42 is rewritten in the form

$$\exp\left\{\frac{J}{n-1}\sum_{i=1}^{n}(k_i)^2\right\} = \left[\frac{(n-1)J}{4\pi}\right]^{\frac{n}{2}}\int_{-\infty}^{\infty} \exp\left\{-(n-1)J\sum_{i=1}^{n}(\phi_i)^2 + 2J\sum_{i=1}^{n}k_i\phi_i\right\}\mathscr{D}\phi \tag{44}$$

Substituting Eq. 44 into Eq. 33, the partition function Z becomes

$$\begin{aligned} Z &= \left[\frac{(n-1)J}{4\pi}\right]^{\frac{n}{2}}\sum_{G\in\mathcal{G}}\int_{-\infty}^{\infty}\exp\left\{-(n-1)J\sum_{i=1}^{n}(\phi_i)^2 + 2J\sum_{i=1}^{n}k_i\phi_i\right\}\mathscr{D}\phi \\ &\quad\times\exp\left\{B\sum_{i=1}^{n}k_i\right\} \Rightarrow \\ Z &= \left[\frac{(n-1)J}{4\pi}\right]^{\frac{n}{2}}\sum_{G\in\mathcal{G}}\int_{-\infty}^{\infty}\exp\left\{-(n-1)J\sum_{i=1}^{n}(\phi_i)^2 + \sum_{i=1}^{n}(2J\phi_i+B)k_i\right\}\mathscr{D}\phi \Rightarrow \\ Z &= \left[\frac{(n-1)J}{4\pi}\right]^{\frac{n}{2}}\int_{-\infty}^{\infty}\left[\sum_{G\in\mathcal{G}}\exp\left\{-(n-1)J\sum_{i=1}^{n}(\phi_i)^2+\sum_{i=1}^{n}(2J\phi_i+B)k_i\right\}\right]\mathscr{D}\phi \end{aligned} \tag{45}$$

In the above equation, the order of the integration and the sum of all the possible graphs in the model has been interchanged. It should also be noted that the term containing the sum of the squares of the fields ϕ_i is independent of the possible configurations of the graphs in the model and therefore Eq. 45 may take the following form

$$Z=\left[\frac{(n-1)J}{4\pi}\right]^{\frac{n}{2}}\int_{-\infty}^{\infty}\exp\left\{-(n-1)J\sum_{i=1}^{n}(\phi_i)^2\right\}\sum_{G\in\mathcal{G}}\exp\left\{\sum_{i=1}^{n}(2J\phi_i+B)k_i\right\}\mathscr{D}\phi \tag{46}$$

At this point, the sum of all possible model configurations has to be evaluated and the second term of the product within the integral of Eq. 46 is further analyzed as follows

$$\begin{aligned} \sum_{i=1}^{n}(2J\phi_i+B)k_i &= \sum_{i=1}^{n}(2J\phi_i+B)\sum_{j=1}^{n}a_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n}(2J\phi_i+B)a_{ij} \\ &= \sum_{i=1}^{n}\sum_{j=i+1}^{n}\left[(2J\phi_i+B)a_{ij}+(2J\phi_j+B)a_{ji}\right] \\ &= \sum_{i=1}^{n}\sum_{j=i+1}^{n}\left(2J(\phi_i+\phi_j)+2B\right)a_{ij} \end{aligned}$$

and

$$\sum_{G\in\mathcal{G}} \exp\left\{\sum_{i=1}^{n}(2J\phi_i + B)k_i\right\} = \prod_{i=1}^{n}\prod_{j=i+1}^{n}\sum_{a_{ij}=0}^{1} \exp\left\{(2J(\phi_i+\phi_j)+2B)a_{ij}\right\}$$
$$= \prod_{i=1}^{n}\prod_{j=i+1}^{n}\left(1+\mathrm{e}^{2J(\phi_i+\phi_j)+2B}\right) = \exp\left\{\sum_{i=1}^{n}\sum_{j=i+1}^{n}\ln\left(1+\mathrm{e}^{2J(\phi_i+\phi_j)+2B}\right)\right\} \tag{47}$$

Substituting Eq. 47 into Eq. 46, yields

$$Z = \left[\frac{(n-1)J}{4\pi}\right]^{\frac{n}{2}}\int_{-\infty}^{\infty}\exp$$
$$\times\left\{-(n-1)J\sum_{i=1}^{n}(\phi_i)^2 + \sum_{i=1}^{n}\sum_{j=i+1}^{n}\ln\left(1+\mathrm{e}^{2J(\phi_i+\phi_j)+2B}\right)\right\}\mathscr{D}\phi \Rightarrow$$
$$Z = \int_{-\infty}^{\infty}\exp\left\{-(n-1)J\sum_{i=1}^{n}(\phi_i)^2 + \sum_{i=1}^{n}\sum_{j=i+1}^{n}\ln\left(1+\mathrm{e}^{2J(\phi_i+\phi_j)+2B}\right)\right.$$
$$\left.+\frac{n}{2}\ln[(n-1)J] - \frac{n}{2}\ln 4\pi\right\}\mathscr{D}\phi$$
$$Z = \int_{-\infty}^{\infty}\mathrm{e}^{\mathscr{H}(\phi)}\mathscr{D}\phi \tag{48}$$

where $\mathscr{H}(\phi)$ is the *effective hamiltonian*

$$\mathscr{H}(\phi) = -(n-1)J\sum_{i=1}^{n}(\phi_i)^2 + \sum_{i=1}^{n}\sum_{j=i+1}^{n}\ln\left(1+\mathrm{e}^{2J(\phi_i+\phi_j)+2B}\right)$$
$$+\frac{n}{2}\ln(n-1)J - \frac{n}{2}\ln 4\pi \Rightarrow$$
$$\mathscr{H}(\phi) = -(n-1)J\sum_{i=1}^{n}(\phi_i)^2 + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1(\neq i)}^{n}\ln\left(1+\mathrm{e}^{2J(\phi_i+\phi_j)+2B}\right)$$
$$+\frac{n}{2}\ln[(n-1)J] - \frac{n}{2}\ln 4\pi \tag{49}$$

The importance of Eq. 48 above lies within the fact that it has been made possible to transform the initial model (Eq. 33) to a field theory of a continuous variable (evaluated at n points). Unfortunately, the integral of the equation above cannot be calculated in closed form (Park and Newman 2004b), but only through approximation techniques, like the mean-field theory.

Appendix 1.1: Mean-field theory

The simplest possible approximation is that of the mean-field, under which fluctuations in the field are ignored and ϕ_i is always set to its most probable value, located at the saddle point where the first derivative is equal to zero (Park and Newman 2004a).

$$\frac{\partial \mathscr{H}(\phi)}{\partial \phi_i} = 0 \Rightarrow -2(n-1)J\phi_i + J \sum_{j=1(\neq j)}^{n} \frac{e^{2J(\phi_i+\phi_j)+2B}}{1+e^{2J(\phi_i+\phi_j)+2B}} = 0$$

Using the identity

$$\frac{e^x}{1+e^x} = \frac{1}{2}\left[\tanh\frac{x}{2}+1\right] \tag{50}$$

the following equation emerges

$$-2(n-1)\phi_i + \sum_{j=1(\neq i)}^{n} \left[\tanh(J(\phi_i+\phi_j)+B)+1\right] = 0$$

which has the symmetric solution $\phi_0 = \phi_i$ for every i

$$\begin{aligned} &-2(n-1)\phi_0 + \sum_{j=1(\neq i)}^{n} [\tanh(2J\phi_0+B)+1] = 0 \Rightarrow \\ &-2(n-1)\phi_0 + (n-1)\,[\tanh(2J\phi_0+B)+1] = 0 \Rightarrow \phi_0 \\ &= \frac{1}{2}\,[\tanh(2J\phi_0+B)+1] \end{aligned} \tag{51}$$

In this case, the path integral of Eq. 48 is simplified to n independent Gaussian Integrals, in which case the partition function becomes

$$Z = \int_{-\infty}^{\infty} e^{\mathscr{H}(\phi)}\mathscr{D}\phi = \int_{-\infty}^{\infty} e^{n\mathscr{H}(\phi_0)}\mathscr{D}\phi = e^{n\mathscr{H}(\phi_0)}\int_{-\infty}^{\infty}\mathscr{D}\phi \Rightarrow Z = e^{n\mathscr{H}(\phi_0)} \tag{52}$$

and finally, the free energy, becomes

$$\begin{aligned} F \equiv \ln Z = n\mathscr{H}(\phi_0) \Rightarrow \\ F = -n(n-1)J(\phi_0)^2 + \frac{1}{2}n(n-1)\ln\left(1+e^{4J\phi_0+2B}\right) \\ +\frac{n}{2}\ln[(n-1)J] - \frac{n}{2}\ln 4\pi \end{aligned} \tag{53}$$

Appendix 2: Introducing ERGM into Bayesian NMF

Having chosen the likelihood function (Eq. 25) and the a-priori distribution (Eq. 24), we may employ the classic formula of Bayesian Inference (Eq. 6) to approximate the

a-posteriori probability of the model parameters (elements of matrix $\widetilde{A} = WH$), given the data (elements of matrix A) and the hyper-parameters (Θ)

$$P(\widetilde{a_{ij}}|a_{ij},\Theta) \propto \mathcal{L}(a_{ij}|\widetilde{a_{ij}}) \times P(\widetilde{a_{ij}}|\Theta) \times P(\Theta) \Rightarrow P(\widetilde{a_{ij}}|a_{ij},\Theta) \propto \frac{\widetilde{a_{ij}}^{a_{ij}}}{a_{ij}!} \mathrm{e}^{-\widetilde{a_{ij}}} \times \frac{1}{Z} \times \mathrm{e}^{\Theta \widetilde{a_{i,j}}} \times P(\Theta) \tag{54}$$

The partition function Z is independent of the parameters $\widetilde{a_{ij}}$ and the data a_{ij} of the model. Hyper-parameter Θ is also deterministically defined (Eq. 24). Therefore, Eq. 54 is simplified to

$$P(\widetilde{a_{ij}}|a_{ij},\Theta) \propto \frac{\widetilde{a_{ij}}^{a_{ij}}}{a_{ij}!} \mathrm{e}^{-\widetilde{a_{ij}}} \times \mathrm{e}^{\Theta \widetilde{a_{i,j}}} \Rightarrow P(\widetilde{a_{ij}}|a_{ij},\Theta) \propto \frac{\widetilde{a_{ij}}^{a_{ij}}}{a_{ij}!} \mathrm{e}^{(\Theta-1)\widetilde{a_{i,j}}} \tag{55}$$

The element $\widetilde{a_{ij}}$ of matrix $\widetilde{A}$ is computed from the inner product of i^{th} row vector of matrix W times the j^{th} column vector of matrix H

$$P(\mathbf{w_i}^\top \mathbf{h_j}|a_{ij},\Theta) \propto \frac{\mathbf{w_i}^\top \mathbf{h_j}^{a_{ij}}}{a_{ij}!} \mathrm{e}^{(\Theta-1)\mathbf{w_i}^\top \mathbf{h_j}} \tag{56}$$

The objective is to find those values for $\mathbf{w_i}^\top$, $\mathbf{h_j}$ that maximize the a-posteriori probability of the parameters of the model (right-hand side of Eq. 56). As NMF has been defined as a minimization problem (Eq. 5), Eq. 56 above must be converted to an equivalent minimization problem. This conversion is achieved by taking the negative natural logarithm of the aforementioned equation (Wang and Zhang 2013)

$$\mathcal{D}(a_{ij}, \mathbf{w_i}^\top \mathbf{h_j}) \equiv -\ln P(\mathbf{w_i}^\top \mathbf{h_j}|a_{ij},\Theta) = \ln(a_{ij}!) - a_{ij} \ln \mathbf{w_i}^\top \mathbf{h_j} + (1-\Theta)\mathbf{w_i}^\top \mathbf{h_j} \tag{57}$$

Using the Stirling Formula ($\ln(x!) = x \ln x - x$) for the term $\ln(a_{ij}!)$, yields

$$\begin{aligned} \mathcal{D}(a_{ij}, \mathbf{w_i}^\top \mathbf{h_j}) &= a_{ij} \ln a_{ij} - a_{ij} - a_{ij} \ln \mathbf{w_i}^\top \mathbf{h_j} + (1-\Theta)\mathbf{w_i}^\top \mathbf{h_j} \\ &= a_{ij} \ln \frac{a_{ij}}{\mathbf{w_i}^\top \mathbf{h_j}} - a_{ij} + (1-\Theta)\mathbf{w_i}^\top \mathbf{h_j} \end{aligned} \tag{58}$$

The gradient of Eq. 58 with respect to vectors $\mathbf{w_i}^\top$, $\mathbf{h_j}$ is computed as follows

$$\begin{aligned} \nabla \mathcal{D}_{\mathbf{w_i}^\top}(a_{ij}, \mathbf{w_i}^\top \mathbf{h_j}) &= \sum_{j=1}^{k} \left[-\frac{a_{ij}}{\mathbf{w_i}^\top \mathbf{h_j}} \mathbf{h_j}^\top + (1-\Theta)\mathbf{h_j}^\top \right] \\ &= -\frac{\mathbf{a_i}^\top}{\widetilde{\mathbf{a}}_\mathbf{i}^\top} H^\top + (1-\Theta)\mathbf{e}^\top H^\top \end{aligned} \tag{59}$$

$$\nabla \mathcal{D}_{\mathbf{h_j}}(a_{ij}, \mathbf{w_i}^\top \mathbf{h_j}) = \sum_{i=1}^{k} \left[-\mathbf{w_i}^\top \frac{a_{ij}}{\mathbf{w_i}^\top \mathbf{h_j}} + (1-\Theta)\mathbf{w_i}^\top \right]$$
$$= -W^\top \frac{\mathbf{a_i}^\top}{\widetilde{\mathbf{a}}_\mathbf{i}^\top} + (1-\Theta) W^\top \mathbf{e}^\top \tag{60}$$

and vectors $\mathbf{w_i}^\top, \mathbf{h_j}$ are updated according to the multiplicative update rules below, where the update factor is the ratio of the negative component of the gradient versus the positive component (Lee and Seung 2000)

$$(\mathbf{w_i}^\top)^{(t+1)} \leftarrow (\mathbf{w_i}^\top)^{(t)} \circ \frac{\nabla \mathcal{D}^-_{\mathbf{w_i}^\top}}{\nabla \mathcal{D}^+_{\mathbf{w_i}^\top}} \Rightarrow (\mathbf{w_i}^\top)^{(t+1)} \leftarrow (\mathbf{w_i}^\top)^{(t)} \circ \frac{\frac{\mathbf{a_i}^\top}{\widetilde{\mathbf{a}}_\mathbf{i}^\top} H^\top}{(1-\Theta)\mathbf{e}^\top H^\top}$$
$$\mathbf{h_j}^{(t+1)} \leftarrow \mathbf{h_j}^{(t)} \circ \frac{\nabla \mathcal{D}^-_{\mathbf{h_j}}}{\nabla \mathcal{D}^+_{\mathbf{h_j}}} \Rightarrow \mathbf{h_j}^{(t+1)} \leftarrow \mathbf{h_j}^{(t)} \circ \frac{W^\top \frac{\mathbf{a_i}^\top}{\widetilde{\mathbf{a}}_\mathbf{i}^\top}}{(1-\Theta) W^\top \mathbf{e}^\top} \tag{61}$$

yielding to the following update rules for the basis and coefficient matrices W, H

$$W^{(t+1)} \leftarrow W^{(t)} \circ \frac{\frac{A}{WH} H^\top}{(1-\Theta) E H^\top} \tag{62}$$

$$H^{(t+1)} \leftarrow H^{(t)} \circ \frac{W^\top \frac{A}{WH}}{(1-\Theta) W^\top E} \tag{63}$$

References

Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749. doi:10.1109/TKDE.2005.99

Alexandridis G, Siolas G, Stafylopatis A (2013) Improving social recommendations by applying a personalized item clustering policy. In: Proceedings of the fifth ACM RecSys workshop on recommender systems and the social web co-located with the 7th ACM conference on recommender systems (RecSys 2013), Hong Kong, China, 13 Oct 2013. http://ceur-ws.org/Vol-1066/Paper1

Alexandridis G, Siolas G, Stafylopatis A (2015) Accuracy versus novelty and diversity in recommender systems: a nonuniform random walk approach. In: Ulusoy O, Tansel AU, Arkun E (eds) Recommendation and search in social networks, lecture notes in social networks. Springer, Berlin, pp 41–57. doi:10.1007/978-3-319-14379-8_3

Alpaydin E (2014) Introduction to machine learning, third, 3rd edn. MIT Press, Cambridge

Bellogin A, Cantador I, Diez F, Castells Chavarriaga E (2011) An empirical comparison of social, collaborative filtering, and hybrid recommenders. ACM TIST 4:14

Bennett J, Lanning S (2007) The netflix prize. In: Proceedings of the KDD Cup Workshop 2007, ACM, New York, pp 3–6. http://www.cs.uic.edu/~liub/KDD-cup-2007/NetflixPrize-description

Cantador I, Brusilovsky P, Kuflik T (2011) 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In: Proceedings of the 5th ACM conference on recommender systems, ACM, New York, RecSys 2011

de Wit JJ (2008) Evaluating recommender systems—an evaluation framework to predict user satisfaction for recommender systems in an electronic programme guide context. Master's thesis, University of Twente

Desrosiers C, Karypis G (2011) A comprehensive survey of neighborhood-based recommendation methods. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) Recommender systems handbook. Springer, New York, pp 107–144. doi:10.1007/978-0-387-85820-3_4

Golbeck JA (2005) Computing and applying trust in web-based social networks. PhD thesis, College Park, aAI3178583

Hu Y, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. In: Proceedings of the 2008 Eighth IEEE international conference on data mining, IEEE Computer Society, Washington, ICDM '08, pp 263–272. doi:10.1109/ICDM.2008.22

Hubbard J (1959) Calculation of partition functions. Phys Rev Lett 3:77–78. doi:10.1103/PhysRevLett.3.77

Jamali M (2010) The flixster dataset. http://www.cs.sfu.ca/~sja25/personal/datasets/

Jamali M, Ester M (2009) Trustwalker: a random walk model for combining trust-based and item-based recommendation. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, KDD '09, pp 397–406. doi:10.1145/1557019.1557067

Jamali M, Ester M (2010) A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the Fourth ACM conference on recommender systems, ACM, New York, RecSys '10, pp 135–142. doi:10.1145/1864708.1864736

Konstas I, Stathopoulos V, Jose JM (2009) On social networks and collaborative recommendation. In: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval, ACM, New York, SIGIR '09, pp 195–202. doi:10.1145/1571941.1571977

Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, ACM, New York, KDD '08, pp 426–434. doi:10.1145/1401890.1401944

Lee D, Seung H (1999) Learning the parts of objects by non-negative matrix factorization. Nature. http://www.nature.com/nature/journal/v401/n6755/abs/401788a0.html

Lee DD, Seung HS (2000) Algorithms for non-negative matrix factorization. In: In NIPS, MIT Press, pp 556–562

Ma H, King I, Lyu MR (2009) Learning to recommend with social trust ensemble. In: Proceedings of the 32nd International ACM SIGIR conference on research and development in information retrieval, ACM, New York, SIGIR '09, pp 203–210. doi:10.1145/1571941.1571978

Massa P, Avesani P (2009) Trust metrics in recommender systems. In: Golbeck J (ed) Computing with social trust, human computer interaction series. Springer, London, pp 259–285. doi:10.1007/978-1-84800-356-9_10

Newman MEJ (2010) Networks: an introduction, 1st edn. Oxford University Press, Oxford

Nunez-Gonzalez JD, Grana M, Apolloni B (2015) Reputation features for trust prediction in social networks. Neurocomputing 166:1–7. doi:10.1016/j.neucom.2014.10.099

Park J, Newman M (2004a) Solution of the two-star model of a network. Phys Rev E 70(066):146. doi:10.1103/PhysRevE.70.066146

Park J, Newman M (2004b) Statistical mechanics of networks. Phys Rev E 70(6):066117. doi:10.1103/PhysRevE.70.066117 cond-mat/0405566

Park J, Newman MEJ (2005) Solution for the properties of a clustered network. Phys Rev E 72(2):026136. doi:10.1103/PhysRevE.72.026136 cond-mat/0412579

Psorakis I, Roberts S, Ebden M, Sheldon B (2011) Overlapping community detection using bayesian non-negative matrix factorization. Phys Rev E 83(066):114. doi:10.1103/PhysRevE.83.066114

Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on computer supported cooperative work, ACM, New York, CSCW '94, pp 175–186. doi:10.1145/192844.192905

Robins G, Pattison P, Kalish Y, Lusher D (2007) An introduction to exponential random graph (p*) models for social networks. Soc Netw 29(2):173–191. doi:10.1016/j.socnet.2006.08.002 (special section: advances in exponential random graph (p*) models)

Shi Y, Larson M, Hanjalic A (2010) List-wise learning to rank with matrix factorization for collaborative filtering. In: Proceedings of the fourth ACM conference on recommender systems, ACM, New York, RecSys '10, pp 269–272. doi:10.1145/1864708.1864764

Wang YX, Zhang YJ (2013) Nonnegative matrix factorization: a comprehensive review. IEEE Trans Knowl Data Eng 25(6):1336–1353. doi:10.1109/TKDE.2012.51

Weimer M, Karatzoglou A, Le QV, Smola A (2007) Cofirank maximum margin matrix factorization for collaborative ranking. In: Proceedings of the 20th international conference on neural information

processing systems, Curran Associates Inc., USA, NIPS'07, pp 1593–1600. http://dl.acm.org/citation.cfm?id=2981562.2981762

Yang X, Steck H, Guo Y, Liu Y (2012) On top-k recommendation using social networks. In: Proceedings of the sixth ACM conference on recommender systems, ACM, New York, RecSys '12, pp 67–74. doi:10.1145/2365952.2365969

Yang X, Guo Y, Liu Y, Steck H (2014) A survey of collaborative filtering based social recommender systems. Comput Commun 41:1–10. doi:10.1016/j.comcom.2013.06.009

Zhou Y, Wilkinson D, Schreiber R, Pan R (2008) Large-scale parallel collaborative filtering for the netflix prize. In: Proceedings of the 4th international conference on algorithmic aspects in information and management, Springer-Verlag, Berlin, Heidelberg, AAIM '08, pp 337–348. doi:10.1007/978-3-540-68880-8_32