Data Min Knowl Disc (2017) 31:1218–1241
DOI 10.1007/s10618-017-0499-9

Social regularized von Mises–Fisher mixture model for item recommendation

Aghiles Salah[1] · **Mohamed Nadif**[1]

Received: 25 September 2016 / Accepted: 20 February 2017 / Published online: 24 March 2017

Abstract Collaborative filtering (CF) is a widely used technique to guide the users of web applications towards items that might interest them. CF approaches are severely challenged by the characteristics of user-item preference matrices, which are often high dimensional and extremely sparse. Recently, several works have shown that incorporating information from social networks—such as friendship and trust relationships—into traditional CF alleviates the sparsity related issues and yields a better recommendation quality, in most cases. More interestingly, even with comparable performances, social-based CF is more beneficial than traditional CF; the former makes it possible to provide recommendations for cold start users. In this paper, we propose a novel model that leverages information from social networks to improve recommendations. While existing social CF models are based on popular modelling assumptions such as Gaussian or Multinomial, our model builds on the von Mises–Fisher assumption which turns out to be more adequate, than the aforementioned assumptions, for high dimensional sparse data. Setting the estimate of the model parameters under the maximum likelihood approach, we derive a scalable learning algorithm for analyzing data with our model. Empirical results on several real-world datasets provide strong support for the advantages of the proposed model.

Keywords Recommender systems · Collaborative filtering · Mixture models · von Mises–Fisher distribution · Directional statistics

---

Responsible editors: Kurt Driessens, Dragi Kocev, Marko Robnik Šikonja and Myra Spiliopoulou.

✉ Aghiles Salah
aghiles.salah@parisdescartes.fr

Mohamed Nadif
mohamed.nadif@parisdescartes.fr

1 LIPADE, Paris Descartes University, 75006 Paris, France

1 Introduction

Collaborative filtering (CF) is an important recommendation technique, which aims to guide the users of web applications towards items that may interest them, based on previous user preferences. CF has been adopted by many real-world applications such as Amazon, YouTube and Netflix. CF approaches can roughly be divided into two main categories:

Memory-based approaches: are based on computing similarities among users and/or items. User-based collaborative filtering (Bobadilla et al. 2013) makes recommendations to users according to their k nearest neighbors. Item-based collaborative filtering (Sarwar et al. 2001) provides users with recommendations according to the k nearest neighbors of items which they enjoyed in the past.

Model-based approaches: aim to fit a model to the user-item matrix in order to capture the hidden preference features and attributes of users and items, respectively. They then predict unknown preferences according to the fitted model. Among such approaches we can cite those based on clustering (Ungar and Foster 1998; Salah et al. 2016a), matrix factorization (MF) (Sarwar et al. 2000; Koren et al. 2009; Delporte et al. 2013) and probabilistic models (Barbieri et al. 2014).

Although the aforementioned approaches, often denoted as traditional CF methods, constitute an important contribution to CF and can offer a good recommendation accuracy, these techniques are still severely challenged by the characteristics of collaborative filtering data, i.e., high dimensionality and sparsity. Over the last few years, with the advent of social networks, social-based collaborative filtering has emerged as a new promising technique to alleviate the sparsity related issues. Such an approach consists in using information from online social networks—usually friendship and/or trust information—to improve recommendations. More intuitively, social-based CF approaches are based on the assumption that, for making a good recommendation, not only the user's expressed preferences are important, but also the user's social interactions. This is natural, since in real life people often turn to their friends to ask for a nice movie to watch, an interesting book to read, a good restaurant, etc. By taking into account this real-life behaviour, social-based CF approaches make more realistic recommendations and are, therefore, expected to offer better performances than traditional CF methods. More interestingly, even with comparable recommendation accuracy, social-based methods are more beneficial than traditional approaches, in that they can make recommendations for cold start users—who have expressed very few preferences or even none at all. This is possible by exploiting the social interactions of cold-start users.

Apart from being high dimensional and sparse, CF data are also directional in nature. Let us consider the *cosine* similarity and *Pearson Correlation Coefficient* (*PCC*), which are widely used measures in CF. The above two, or their variants, have been found to be superior to several other measures, such as euclidean distortions, to assess the similarities between users or items (Sarwar et al. 2001; Linden et al. 2003; Liu et al. 2014). Note that both measures focus on the directions of data vectors, or in other words the similarity between two objects—users or items—is measured relative to the angle between them. The success of these measures in CF suggests that the direction of a user/item preference-vector is relevant, not its magnitude.

It is worth noting that both the above measures are exactly the scalar product between objects lying on the surface of a unit-hypersphere, i.e., objects of unit length (L_2 norm). This is straightforward for the *cosine* similarity, now let us look at the *PCC*, between two users represented by the column vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^d$, given by: $PCC(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x}-\bar{\mathbf{x}})^\top(\mathbf{y}-\bar{\mathbf{y}})}{\|\mathbf{x}-\bar{\mathbf{x}}\|\|\mathbf{y}-\bar{\mathbf{y}}\|}$, where $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are two vectors of the appropriate dimensions, such that $\bar{x}_1 = \cdots = \bar{x}_d = \sum_j x_j/d$ and $\bar{y}_1 = \cdots = \bar{y}_d = \sum_j y_j/d$. Let $\mathbf{x}' = \frac{\mathbf{x}-\bar{\mathbf{x}}}{\|\mathbf{x}-\bar{\mathbf{x}}\|}$ and $\mathbf{y}' = \frac{\mathbf{y}-\bar{\mathbf{y}}}{\|\mathbf{y}-\bar{\mathbf{y}}\|}$, then the PCC between $\mathbf{x}$ and $\mathbf{y}$ is exactly the scalar product (*cosine* similarity) between the two unit-length vectors $\mathbf{x}'$ and $\mathbf{y}'$.

Hence, from a probabilistic perspective using the *cosine* and *PCC* measures is equivalent to assuming that CF data are distributed on the surface of a unit-hypersphere, and to measuring the similarities between objects—relative to the angle between them—using the scalar product. The success of the above directional measures in CF constitutes, therefore, empirical evidence that CF data possesses intrinsic directional characteristics, which should be taken into account to reach better performances. In other words, it seems better to model CF data as directional[1] data distributed on the surface of a unit-hypersphere, i.e., L_2 normalized data, (Mardia and Jupp 2009; Banerjee et al. 2005).

Existing social CF models are, however, based on popular modelling assumptions, such as Gaussian or Multinomial, which are inadequate for directional data lying on the surface of a unit-hypersphere. Hence, it seems natural to question whether it is possible to leverage both the users' social interactions and the directional properties of CF data, simultaneously. In this paper, we provide an answer to this question: we develop a novel social CF model which is based on the von Mises–Fisher (vMF) distribution, which arises naturally for directional data lying on the surface of a unit-hypersphere. The proposed model successfully integrates a directional measure, namely the *cosine* similarity, into a social CF model. This makes it possible to achieve a high recommendation accuracy as illustrated in our experiments. To the best of our knowledge the work we present is the first social-based CF approach that accounts for the directional characteristics of collaborative filtering data.

The remainder of the paper is organized as follows. Section 2 is devoted to present related work. In Sect. 3, we review the vMF mixture model, then we propose to extend the above model to account for social interactions among users and derive a scalable Generalized Expectation-Maximization (GEM) algorithm for inference and parameter estimation (Sect. 4). Finally, we evaluate our contribution on real data sets.

2 Related work

The work we present here is related to two main topics, namely CF approaches accounting for social-network information, and vMF mixture models from directional statistics (Mardia and Jupp 2009; Banerjee et al. 2005).

Over the last few years, several social-based collaborative filtering approaches have been proposed, most of which are based on Probabilistic Matrix Factorization (PMF)

[1] In the rest of this paper we treat "direction data" and "L_2 normalized data" as synonyms.

(Ma et al. 2008, 2009; Jamali and Ester 2010; Ma et al. 2011). The key idea behind these approaches is to make the latent preference factor of each user close to that of his/her direct neighbors in the user–user social graph, so as to capture the influence between friends.

Ma et al. (2008) built a MF model that connects the user-item preference matrix with the user–user social graph through a shared user latent factor. Ma et al. (2009) proposed an approach that fuses a MF model on the user-item matrix with a MF on the user–user graph, then predicts unknown preferences by combining the ratings resulting from both models. Based on the above works, Jamali and Ester (2010) proposed `SocialMF` another matrix factorization-based method that accounts for trust propagation. Ma et al. (2011) proposed to add a regularization term into the traditional PMF (Salakhutdinov and Mnih 2008) so as to bring the latent factors of socially connected users closer to each other. Yang et al. (2013) developed three trust MF-based models that consider different aspects of trust information. The first variant reflects that the preference of a user for an item is influenced by the preferences of his/her trustees on that item, the second reflects that the behaviour of a user will influence that of his/her trusters and the third is a combination of the above two during the prediction phase. More recently, Guo et al. (2015) presented `TrustSVD` an extension of the well known SVD++ (Koren 2008) method that accounts for social trust information. Chaney et al. (2015) proposed Social Poisson Factorization. In addition to learn the user and item latent factors, as in traditional MF, this method introduces a third latent factor that reflects how much each user is influenced by his/her direct neighbors in the social network. Then, the preference of a user for an item is explained by combining the three factors above.

All these efforts demonstrated the importance of considering information from social networks in CF. Nevertheless, existing approaches to social CF are based on popular modelling assumptions such as Gaussian, Multinomial or Poisson, and therefore do not account for the aforementioned directional characteristics of CF data. In this paper, we aim to address this limitation by building a novel model that leverages the social interactions among users as well as the directional properties of CF data. More precisely, we rely on a mixture of von Mises–Fisher distributions.

The vMF distribution is a continuous probability distribution, on a unit-hypersphere, from directional statistics (Mardia and Jupp 2009). It focuses on the directions of objects and measures the distance between them using the *cosine* similarity. Most of the earlier works using vMF distributions focused on low dimensional data, i.e., 2- or 3-dimensional data (McLachlan and Peel 2004), due to the difficulties related to the estimation of the concentration parameter kappa, which involves the inversion of a ratio of Bessel functions. A notable contribution is the mixture of vMF distributions *movMFs* (Banerjee et al. 2005) for clustering high dimensional sparse data. Banerjee et al. (2005) derived an EM-based solution for inference and parameter estimation, and they proposed an accurate approximation to estimate the concentration parameter κ for a high dimensional vMF distribution. Since this contribution, different vMF-based models have been proposed in the context of high dimensionality. For instance, Reisinger et al. (2010) proposed a topic model based on a mixture of vMF distributions. More recently, for text data clustering, Gopal and Yang (2014) proposed a full Bayesian formulation of *movMFs* and developed two novel variants of *movMFs*, namely hierarchical and temporal. Le and Lauw (2014) developed a vMF-based model for the task of *semantic visualization*, i.e., jointly modeling topics and visualization, of text data. Salah et al. (2016b, c) proposed the block von Mises–Fisher mixture model and derived several vMF-based algorithms for co-clustering document-term matrices.

Notation Matrices are denoted with boldface uppercase letters and vectors with boldface lowercase letters. The L_2 norm is denoted by $\|.\|$. The $(d-1)$ dimensional unit sphere embedded in $\mathbb{R}^d$ is denoted by $\mathbb{S}^{d-1}$. Data is represented by a matrix $\mathbf{X} = (x_{ij})$ of size $n \times d$, $x_{ij} \in \mathbb{R}$, its ith row (user) is represented by a vector $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})^\top$, where $\top$ denotes the transpose. The partition of the set of rows $\mathcal{I}$ into g clusters can be represented by a classification matrix $\mathbf{Z}$ of elements z_{ih} in $\{0, 1\}$ satisfying $\sum_{h=1}^{g} z_{ih} = 1$. The notation $\mathbf{z} = (z_1, \ldots, z_n)^\top$, where $z_i \in \{1, \ldots, g\}$ represents the cluster label of i, will also be used. In the same way, the fuzzy classification matrix of $\mathcal{I}$ is denoted by $\tilde{\mathbf{Z}} = (\tilde{z}_{ih})$ where $\tilde{z}_{ih} \in [0, 1]$, satisfying $\sum_{h=1}^{g} \tilde{z}_{ih} = 1$, for all i in $\mathcal{I}$.

3 Preliminaries

A d dimensional vMF (d-vMF) distribution, i.e., $d \geq 2$ is a continuous probability distribution on the surface of a unit-hypersphere $\mathbb{S}^{d-1}$. The probability density function of a data point $\mathbf{x}_i \in \mathbb{S}^{d-1}$, following a d-vMF distribution, is given by:

$$f(\mathbf{x}_i|\boldsymbol{\mu}, \kappa) = c_d(\kappa) \exp^{\kappa \boldsymbol{\mu}^\top \mathbf{x}_i}, \tag{1}$$

where $\boldsymbol{\mu}$ is the mean direction or centroid parameter and κ is the concentration parameter, such that $\|\boldsymbol{\mu}\| = 1$ and $\kappa \geq 0$. The normalization term $c_d(\kappa)$ is equal to $c_d(\kappa) = \frac{\kappa^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\kappa)}$ where $I_r(\kappa)$ represents the modified Bessel function of the first kind and order r. In the vMF distribution the parameter κ controls the concentration of data points $\mathbf{x}_i$, following (1), around the mean direction $\boldsymbol{\mu}$. For more details on the vMF distribution, we recommend the book of Mardia and Jupp (2009).

In the mixture model context, the data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are supposed to be i.i.d. and generated from a mixture of g vMF distributions with a set of unknown parameters Θ. The log-likelihood function of this mixture takes the following form

$$L(\Theta; \mathbf{X}) = \sum_i \log \left(\sum_h \alpha_h f_h(\mathbf{x}_i|\boldsymbol{\mu}_h, \kappa_h) \right), \tag{2}$$

where $\Theta = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_g, \alpha_1, \ldots, \alpha_g, \kappa_1, \ldots, \kappa_g\}$, $\boldsymbol{\mu}_h$ and κ_h represent the centroid and the concentration parameters of the hth component, respectively. Each parameter α_h denotes the proportion of points $\mathbf{x}_i$ generated from the hth component, such that $\sum_h \alpha_h = 1$ and $\alpha_h > 0$, $\forall h \in \{1, \ldots, g\}$. As the optimization of the above function is intractable, we rely on the "complete" data likelihood given by

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\alpha}, \kappa; \mathbf{X}, \mathbf{z}) = \prod_i \alpha_{z_i} (c_d(\kappa_{z_i}) \exp^{\kappa_{z_i} \boldsymbol{\mu}_{z_i}^\top \mathbf{x}_i}), \tag{3}$$

where $\mathbf{z}$ is the latent variable which is assumed to be known, i.e., $z_i = h$ if $\mathbf{x}_i$ is generated from the hth component. Using the classification matrix $\mathbf{Z}$, the corresponding complete data log-likelihood takes the following form:

$$L_c(\Theta; \mathbf{X}, \mathbf{Z}) = \sum_h z_{.h} \log \alpha_h + \sum_h z_{.h} \log c_d(\kappa_h) + \sum_{i,h} z_{ih} \kappa_h \boldsymbol{\mu}_h^\top \mathbf{x}_i$$

where $z_{.h}$ denotes the cardinality of cluster h.

As the latent variable $\mathbf{z}$ is unknown in practice, the authors in Banerjee et al. (2005) proposed to use the EM algorithm (Dempster et al. 1977) to obtain the maximum likelihood estimates for the parameters Θ. The E-step finds the conditional expectation of the missing variable $\mathbf{z}$ given the current estimated parameters $\Theta^{(t)}$ and the observed data, i.e., $\tilde{z}_{ih} = \mathbb{E}(z_{ih} = 1|\mathbf{x}_i, \Theta^{(t)})$. The M-step finds the new parameters $\Theta^{(t+1)}$ maximizing the expectation of the complete data log-likelihood (4) subject to the constraints $\sum_h \alpha_h = 1$, $\|\boldsymbol{\mu}_h\| = 1$ and $\kappa_h > 0$. This procedure leads to the soft vMF clustering algorithm (Banerjee et al. 2005), denoted in this paper as `movMF`.

Note that if we impose the following constraints on the parameters: equality of proportions $\alpha_1 = \cdots = \alpha_h$ and concentration $\kappa_1 = \cdots = \kappa_h$ parameters, the maximization of $L_c(\Theta; \mathbf{X}, \mathbf{Z})$ reduces to the maximization of the spherical k-means criterion (Dhillon and Modha 2001; Banerjee et al. 2005)

$$\sum_{i,h} z_{ih} \kappa_h \boldsymbol{\mu}_h^\top \mathbf{x}_i = \sum_{i,h} z_{ih} < \boldsymbol{\mu}_h, \mathbf{x}_i >= \sum_{i,h} z_{ih} \cos(\delta_{ih})$$

where $<,>$ denotes the scalar product and δ_{ih} is the angle between vectors $\mathbf{x}_i$ and $\boldsymbol{\mu}_h$. So, when relying on a vMF mixture model, the *cosine* similarity is underlying. In fact, the well known spherical k-means algorithm, using the *cosine* similarity instead of euclidean distortions, arises as special case from the `movMF` algorithm, considered in this paper, when we enforce some restrictive constraints (Banerjee et al. 2005).

4 Social regularized von Mises–Fisher mixture model

We now propose *Social-movMFs* a novel model that leverages, simultaneously, the benefits of the vMF modeling assumption and the use of social network information so as to alleviate the sparsity related issues in CF. Specifically, we propose to extend the mixture of vMF distribution *movMFs*, presented above, to account for the social interactions among users. The intuition behind our model is to bring the distributions over clusters of socially connected users closer to each other in order to capture the influences between friends. To this end, inspired by previous works on manifold learning (Zhu and Lafferty 2005; Belkin et al. 2006; Cai et al. 2008; Mei et al. 2008; He et al. 2011), we propose to smooth the posterior probabilities $\tilde{z}_{ih}$ based on the user–user social network. Posterior smoothness can be achieved by using any adequate function, here we adopt the quadratic energy function—also denoted as the graph harmonic function—(Zhu and Lafferty 2005) defined as follows in our case

$$R(\mathcal{T}) = \frac{1}{2}\sum_{h}\sum_{i}\sum_{j}\tau_{ij}(\tilde{z}_{ih} - \tilde{z}_{jh})^2 \tag{4}$$

where $\mathcal{T} = (\tau_{ij})$ is the adjacency matrix of the user–user social graph, i.e., $\tau_{ij} = 1$ if users i and j are socially connected and $\tau_{ij} = 0$ otherwise. The above function is minimized if all socially connected users exhibit similar posterior distributions over clusters. Recall that our purpose is to force socially connected users to exhibit similar distributions over clusters. This objective can be achieved by regularizing the log-likelihood (2) by function (4), which yields the following regularized log-likelihood function:

$$L_r(\Theta; \mathbf{X}, \mathcal{T}) = L(\Theta; \mathbf{X}) - \lambda R(\mathcal{T}) \tag{5}$$

where $R(\mathcal{T})$ plays the role of the regularization term that enforces smoothness of the posterior probabilities on the social network, and λ is the regularization parameter which controls the extent of regularization. Observe that the complete data log-likelihood (4) of *movMFs* arises from (5) as a special case when $\lambda = 0$. The corresponding regularized complete data log-likelihood can be obtained by substituting $L_c(\Theta; \mathbf{X}, \mathbf{Z})$ for $L(\Theta; \mathbf{X})$ in (5).

It is worth noting that the regularized log-likelihood (5) is penalized by users who are socially connected and who exhibit substantially different distributions over clusters. Thus, as opposed to the *movMFs* model, *Social-movMFs* accounts for the interactions among users.

4.1 Maximum likelihood estimates

In order to obtain the maximum likelihood estimates of the model parameters we rely on the *Generalized* EM algorithm (Dempster et al. 1977; McLachlan and Krishnan 2007). The E-step is to estimate the posterior probabilities $\tilde{z}_{ih}$, given the current estimated parameters $\Theta^{(t)}$ and the observed data $\mathbf{X}$, as follows:

$$\tilde{z}_{ih} = \mathbb{E}(z_{ih} = 1|\mathbf{x}_i, \Theta^{(t)}) \propto \alpha_h^{(t)} f_h(\mathbf{x}_i|\boldsymbol{\mu}_h^{(t)}, \kappa_h^{(t)}). \tag{6}$$

The M-step finds the new parameters $\Theta^{(t+1)}$ maximizing or increasing the expectation of the complete data log-likelihood which is given by

$$\begin{aligned} Q(\Theta, \Theta^{(t)}) &= \mathbb{E}\left(L_c(\Theta; \mathbf{X}, \mathbf{Z})|\mathbf{X}, \Theta^{(t)}\right) \\ &= \sum_{i,h} \tilde{z}_{.h} \log \alpha_h + \sum_{h} \tilde{z}_{.h} \log c_d(\kappa_h) + \sum_{i,h} \tilde{z}_{ih}\kappa_h \boldsymbol{\mu}_h^\top \mathbf{x}_i \end{aligned} \tag{7}$$

where $\tilde{z}_{.h} = \sum_i \tilde{z}_{ih}$. The above optimization scheme yields the `movMF` algorithm providing parameter estimation for the classical *movMFs* model (Banerjee et al. 2005). In our case, the purpose is to take into account social interactions among users. Thus, instead of maximizing expression (7), we maximize the following regularized expected complete data log-likelihood (or equivalently the expectation of the regularized complete data log-likelihood):

$$\begin{aligned} Q_r(\Theta, \Theta^{(t)}) &= Q(\Theta, \Theta^{(t)}) - \lambda R(\mathcal{T}) \\ &= \sum_{i,h} \tilde{z}_{.h} \log \alpha_h + \sum_h \tilde{z}_{.h} \log c_d(\kappa_h) + \sum_{i,h} \tilde{z}_{ih} \kappa_h \boldsymbol{\mu}_h^\top \mathbf{x}_i \\ &\quad - \frac{\lambda}{2} \sum_h \sum_i \sum_j \tau_{ij} (\tilde{z}_{ih} - \tilde{z}_{jh})^2 \end{aligned} \tag{8}$$

Note that the direct maximization of expression (8) is intractable due to the introduction of the regularization term—the M-step of EM does not have a closed-form solution. Fortunately, in the GEM algorithm it is sufficient to find a better Θ at each iteration, i.e., we choose $\Theta^{(t+1)}$ so that $Q_r(\Theta^{(t+1)}, \Theta^{(t)}) \geq Q_r(\Theta^{(t)}, \Theta^{(t)})$. Hence, following the strategy described in Cai et al. (2008) and He et al. (2011), which is closely related to the optimization scheme proposed in (Zhu and Lafferty 2005) in the context of semi-supervised learning, we derive an efficient M-step that is guaranteed to increase (8) at each iteration. The key idea is to optimize the different parts of (8) separately so as to increase $Q_r(\Theta^{(t)}, \Theta^{(t)})$.

We first minimize the regularization term as it depends only on the posterior probabilities $\tilde{z}_{ih}$. This step yields the smoothed posteriors on the user–user graph. It is obvious that the smoothed posterior $\tilde{z}_{ih}$ minimizing $R(\mathcal{T})$ is given by $\tilde{z}_{ih} = \frac{\sum_j \tau_{ij} \tilde{z}_{jh}}{\sum_j \tau_{ij}}$. This minimization scheme, however, can lead to a strong smoothing, where the new posteriors are substantially far from the original ones. Hence, for a better control of the smoothing process one should decrease $R(\mathcal{T})$ gradually, instead of its direct minimization. This can be done by the Newton–Raphson method as follows:

$$\tilde{z}_{ih} := \tilde{z}_{ih} - \gamma \frac{R'(\mathcal{T})}{R''(\mathcal{T})} = (1-\gamma)\tilde{z}_{ih} + \gamma \frac{\sum_j \tau_{ij} \tilde{z}_{jh}}{\sum_j \tau_{ij}} \tag{9}$$

where R', R'' denote the first and second derivative of the regularization term relative to $\tilde{z}_{ih}$, and $\gamma \in [0, 1]$ is the Newton–Raphson's step parameter. In our context, we can think of γ being the level of smoothing. If $\gamma = 0$ then no smoothing is performed, and if $\gamma = 1$ then the smoothed posterior distribution of each user is completely specified by the posterior distributions of his neighbors in the social graph.

Once the smoothing step is done, the next step consists in maximizing the expectation of the complete data log-likelihood Q relative to the parameters Θ, which yields the following update formulas (Banerjee et al. 2005):

$$\hat{\alpha}_h = \frac{\sum_i \tilde{z}_{ih}}{n}, \tag{10a}$$

$$\hat{\boldsymbol{\mu}}_h = \frac{\mathbf{r}_h}{\|\mathbf{r}_h\|} \quad \text{where} \quad \mathbf{r}_h = \sum_i \tilde{z}_{ih} \mathbf{x}_i, \tag{10b}$$

$$\hat{\kappa}_h \approx \frac{\bar{r}_h d - \bar{r}_h^3}{1 - \bar{r}_h^2} \quad \text{where} \quad \bar{r}_h = \frac{I_{d/2}(\hat{\kappa}_h)}{I_{d/2-1}(\hat{\kappa}_h)} = \frac{\|\mathbf{r}_h\|}{\sum_i \tilde{z}_{ih}}. \tag{10c}$$

Notice that, an exact estimation of the concentration parameter $\hat{\kappa}_h$ implies to inverse a ratio of Bessel functions, which has not a close form expression. To overcome this difficulty Banerjee et al. (2005) proposed the efficient approximation (10c), which is suitable for high dimensional datasets. Furthermore, Tanabe et al. (2007) showed theoretically that the above approximation lies in the interval in which the exact ML estimates of κ_h exists. More accurate approximation of κ_h can be reached by using iterative methods (Tanabe et al. 2007; Sra 2012), the latter, however, are less suitable in high dimensions, since they involve an extensive computation of a ratio of Bessel functions.

It is worth nothing that the M-step described above does not necessarily increase the regularized log-likelihood function (5) due to the smoothing step. In order to address this issue, we adopt a strategy similar to that described in Cai et al. (2008) and He et al. (2011). After each M-step we check if the regularized log-likelihood function has been decreased, then we decrease the smoothing parameter γ and perform again the M-step. Alternating the above E-step and optimization scheme (M-step) constitutes our soft social movMF algorithm—denoted as `Soc-movMF` in the rest of the paper—which is described in more details by Algorithm 1.

It can be shown that the computational complexity of `Soc-movMF` is $O(g \cdot nr + g \cdot ns)$ per iteration, which scales linearly with the number of observed relations ns in the social network and observed ratings nr in the user-item matrix. In practice, we have $nr \ll n \times d$ and $ns \ll n \times n$, thereby `Soc-movMF` is very efficient and suitable for large datasets.

Fitting the parameters of *Social-movMFs* to the user-item matrix, using `Soc-movMF`, constitutes the training component of our CF system. Once this step is done, the missing ratings of the ith user can be easily predicted as follows

$$\hat{\mathbf{x}}_i = \frac{\sum_h \tilde{z}_{ih} \boldsymbol{\mu}_h}{\| \sum_h \tilde{z}_{ih} \boldsymbol{\mu}_h \|}. \tag{11}$$

Observation It is worth noting that the proposed model *Social-movMFs* allows the propagation of social information, between users, through the smoothing process (see step 3.1 in Algorithm 1). A property which is desirable since it may help to alleviate the sparsity related issues in both the rating matrix and social network, as it has already been emphasized in previous works (Ma et al. 2008, 2009; Jamali and Ester 2010).

5 Experimental study

To show the benefits of our approach, we conduct extensive experiments on several real-world datasets, in which we benchmark `Soc-movMF` against several strong competing social-CF methods, namely `SoRec` (Ma et al. 2008), `RSTE` (Ma et al. 2009), `SocialMF` (Jamali and Ester 2010), `SoReg` (Ma et al. 2011), `TrustMF` (Yang et al. 2013) and `TrustSVD` (Guo et al. 2015). In order to illustrate the advantage of

Algorithm 1: `Soc-movMF`.

Input: $\mathbf{X}$ ($\mathbf{x}_i \in \mathbb{S}^{d-1}$) the rating matrix of size $(n \times d)$,
$\mathcal{T}$ the adjacency matrix of the social network, g the number of clusters, λ the regularization parameter.
Output: $\tilde{\mathbf{Z}}$, Θ;
1. Random initialization: $\Theta \leftarrow \Theta^{(0)}$; $t \leftarrow 0$;
repeat
  2. Expectation step of GEM:
  for $i = 1$ **to** n **do**
    for $h = 1$ **to** g **do**
      $\tilde{z}_{ih}^{(t)} \leftarrow \frac{\alpha_h f_h(\mathbf{x}_i|\boldsymbol{\mu}_h,\kappa_h)}{\sum_\ell \alpha_\ell f_\ell(\mathbf{x}_i|\boldsymbol{\mu}_\ell,\kappa_\ell)}$
    end for
  end for
  3. Maximization step of GEM:
  smooth $\leftarrow$ TRUE; $\gamma \leftarrow 0.9$;
  while smooth **do**
    3.1 Smooth the posterior probabilities
    $\tilde{z}_{ih}^{(t+1)} \leftarrow \tilde{z}_{ih}^{(t)};\ \forall i, h$
    $\tilde{z}_{ih}^{(t+1)} \leftarrow (1-\gamma)\tilde{z}_{ih}^{(t+1)} + \gamma \frac{\sum_j \tau_{ij}\tilde{z}_{jh}^{(t+1)}}{\sum_j \tau_{ij}};\ \forall i, h$
    3.2. Compute the new parameters $\Theta^{(t+1)}$
    for $h = 1$ **to** g **do**
      $\hat{\alpha}_h \leftarrow \frac{\sum_i \tilde{z}_{ih}^{(t+1)}}{n}$
      $\hat{\boldsymbol{\mu}}_h \leftarrow \frac{\mathbf{r}_h}{\|\mathbf{r}_h\|}$ with $\mathbf{r}_h = \sum_i \tilde{z}_{ih}^{(t+1)}\mathbf{x}_i$
      $\hat{\kappa}_h \leftarrow \frac{\bar{r}_h d - \bar{r}_h^3}{1-\bar{r}_h^2}$ with $\bar{r}_h = \frac{I_{d/2}(\hat{\kappa}_h)}{I_{d/2-1}(\hat{\kappa}_h)} = \frac{\|\mathbf{r}_h\|}{\sum_i \tilde{z}_{ih}^{(t+1)}}$.
    end for
    3.3. Compute the regularized log-likelihood
    $L_r(\Theta^{(t+1)}; X) \leftarrow L(\Theta^{(t+1)}; X) - \lambda R(\mathcal{T})$
    if $L_r(\Theta^{(t+1)}; X) < L_r(\Theta^{(t)}; X)$ **then**
      3.4 Decrease the smoothing parameter γ
      $\gamma \leftarrow \gamma \times \gamma$
    else
      smooth $\leftarrow$ FALSE; $t \leftarrow t + 1$;
    end if
  end while
until convergence

leveraging information from social networks, we also compare `Soc-movMF` with the traditional *movMFs*-based clustering algorithm `movMF` (Banerjee et al. 2005). Notice that `movMF` arises as a special case from `Soc-movMF` when $\lambda = 0$. For all competing methods we used LibRec[2]—a Java library which implements numerous state-of-the-art recommendation algorithms—, except for `movMF` we used our implementation.

Note that previous works, in the context of social CF, established empirically that the aforementioned competing methods perform better than several traditional CF

[2] http://www.librec.net/.

Table 1 Description of datasets

Characteristics	Datasets				
	FilmTrust	CiaoDVD	Ciao-280k	Flixster	Epinions
#Users	1508	17,615	7375	147,612	40,163
#Items	2071	16,121	106,797	48,794	139,738
#Ratings	35,497	72,665	284,052	8,196,077	664,824
Density	1.14%	0.026%	0.04	0.114%	0.01%
ratings-scale	[0.5,4]	[1,5]	[1,5]	[0.5,5]	[1,5]
#links	1853	22,484	111,781	2,538,746	442,979
links-type	Trust	Trust	Trust	Friendship	Trust
Network-density	0.08%	0.01%	0.2%	0.011%	0.029%

approaches, such as `SVD++` (Koren 2008), Matrix Factorization- and probabilistic MF-based methods, therefore we do not consider these approaches in our experiments.

5.1 Datasets

We selected five popular benchmark real-world datasets, including both the user-item preferences and social relationships between users, namely FilmTrust, CiaoDVD, Ciao-280k, Flixster and Epinions.

- **Flixster**[3]: is a social rating dataset crawled by Jamali and Ester (2010) from the flixster.com website. The latter is a social movie website, where users can watch, buy, share, rate and review movies. Each user can create his/her own social network by adding users into his/her friend list.
- **FilmTrust**[4]: is a dataset crawled by Guo et al. (2013) from the FilmTrust website. FilmTrust, similar to Flixster, is a movie rating and sharing community. Unlike the Flixster datasets, the social interactions between users are directed in FilmTrust.
- **Ciao-280k**[5]: this dataset is crawled by Tang et al. (2012) from the product review website Ciao (ciao.co.uk). In the above site users can rate and review products as well as add users to their trust network.
- **CiaoDVD**[4]: is crawled by Guo et al. (2014) from the Ciao site from the category of DVDs.
- **Epinions**[5]: is crawled from the Epinions website (epinions.com). As Ciao, Epinions is a consumer review site where users can rate and review products, and add members to their trust list.

The characteristics of the above datasets are reported in Table 1. The trust links, in the social network, are directed and give rise to asymmetric networks, while friendship are undirected and give rise to symmetrical networks.

[3] http://www.cs.ubc.ca/~jamalim/datasets/.

[4] http://www.librec.net/datasets.html.

[5] http://www.jiliang.xyz/trust.html.

5.2 Evaluation metrics

Evaluating CF approaches still remains a challenging task. In our experiments, we adopt a commonly used approach to evaluate such systems, that consists in assessing the recommendation accuracy on a set of held-out items—the test set. To this end, we retain four widely used measures, from information retrieval, namely the Normalized Discount Cumulative Gain (nDCG), Mean Reciprocal Rank (MRR), Precision@k (Prec@k) and Recall@k (Rec@k), where k is the number of items in the recommendation list.

- **MRR**: The Reciprocal Rank (RR) for a recommendation list is the multiplicative inverse of the rank of the first "good" item. The mean reciprocal rank is the average of the RR's of all the recommendation lists.

$$MRR = \frac{1}{n}\sum_{i}\frac{1}{rank_i}$$

  where n is the number of users who receive recommendations, i.e., the number of recommendation lists, and $rank_i$ is the rank of the first correct item in the recommendation list of user i. Intuitively, the RR measures how far a user should go in the recommendation list to find a good item.
- **nDCG**[6]: the DCG is used to measure the gain of each item relative to its position in a ranked list of items. Formally the DCG for a user i is given by

$$DCG_i = \sum_{j\in D_i}\frac{1}{\log(rank_j + 1)}$$

  where D_i denotes the set of held-out items for user i, and $rank_j$ is the rank of item j. The normalized DCG_i is given by

$$nDCG_i = \frac{DCG_i}{idealDCG_i}$$

  where the $idealDCG_i$ is the best achievable DCG_i, i.e., the value of the measure if the ranking was perfect. The nDCG is high if the most relevant items appear early in the ranked list. To evaluate an entire model we compute the average nDCG over all users: $nDCG = \frac{1}{n}\sum_i nDCG_i$.
- **Precision@**k: for each user the Precision@k denotes the proportion of good items in his/her top-k recommendation list. To evaluate an entire CF system we compute the average Precision@k over all users.
- **Recall@k** the Recall@k for a user is the proportion of good items, in the user's top-k recommendation list, from the number of relevant held-out items for that user. As for the above measures, we can compute the average Recall@k over all users to evaluate an entire model.

[6] Several variants of nDCG exist, here we adopt the same as in LibRec for fairness purpose.

The nDCG measures the raking quality of a model, while Precision@k and Recall@k assess the quality of a user's top-k recommendation list. All the above measures vary from 0.0 to 1.0, the higher these measures, the better is the recommendation quality.

Notice that we do not consider prediction metrics, such as MAE and RMSE, in our experiments. As it has been already established by previous works (Cremonesi et al. 2010; Amatriain et al. 2012; Loiacono et al. 2014; Chaney et al. 2015), low MAE and RMSE do not necessarily equate to best user satisfaction. As emphasized in (Cremonesi et al. 2010; Amatriain et al. 2012), the purpose of a CF system is to provide users with a set of relevant items, as a ranked list. In most commercial systems, users do not receive the predicted rating values, but rather lists of few selected items and ordered according to these values. Thus, the task of item recommendation is by nature a ranking problem. It is therefore more adequate to evaluate CF systems according to the quality of lists of items they recommend. Furthermore, in our case each approach makes predictions in its own range. For instance our method normalizes data so that it lies on a unit hypersphere, some retained competing methods map the original ratings to the interval [0, 1]. So, it is not consistent to compare these approaches in terms of prediction accuracy, by using measures such as MAE and RMSE which are strongly sensitive to the range in which the predicted ratings lie.

5.3 Experimental settings

In our experiments we adopt the fivefold cross validation strategy. Each dataset was randomly split into fivefolds. At each run, fourfolds—80% of data—are used for training and the remaining fold is used for testing. On each dataset we perform five runs in order to test all folds. The average performance over the five runs is reported as the final result.

In order for comparisons to be fair and assess the impact of the social network information, we use the same random parameters Θ^0 to initialize both `Soc-movMF` and `movMF`, in all our experiments.

Our approach `Soc-movMF` and the traditional `movMF`, require as an input the number of clusters g which is analogous the number of latent factors k in matrix factorization-based approaches. The number of latent factors k, in the retained baselines, is usually set to 5 or 10 in previous works. In our experiments we found that the different baselines achieve better performances with $k = 5$ in almost all situations, we choose therefore $k = 5$ for all MF-based approaches. Concerning the number of clusters, we empirically found that both our approach and `movMF` provide high performances with a small number of clusters—usually $g \leq 10$. Thus, for fairness purpose, and due to the analogy between g and k, we set $g = 5$ in all our experiments. Nevertheless, we illustrate the impact of g in our experiments.

Another input required by all approaches including the proposed one and competing methods, is the regularization parameter λ. In our case we set this parameter to 1 so as to give equal importance to both the preference and social information. The impact of λ is, however, illustrated in our experiments. The MF-based methods considered here often require several regularization parameters that are usually set to $\lambda = 0.001$. The other settings for the regularization parameters, determined either by our experiments or by previous works (Guo et al. 2015), are as follows: TrustMF and socialMF $\lambda_T = 1.0$, SoRec $\lambda_c = 0.001$ for Flixster and 1.0 for the others, SoReg $\beta = 1.0$ for Flixster and 0.1 for the others, RSTE $\alpha = 0.4$, TrustSVD $\lambda_t = 0.9,\ 1.0,\ 1.0,\ 0.5,\ 0.5$ and $\lambda = 1.2,\ 0.5,\ 0.5,\ 0.9,\ 0.8$ for FilmTrust, CiaoDVD, Ciao-280k, Epinions and Flixster, respectively.

5.4 Empirical results and discussion

The average performances of each approach over the different datasets are reported in Table 2. In order to ease interpretation, Figs. 1, 2, 3 and 4 provide another representation of the results reported in Table 2. As these results show clearly, the proposed Soc-movMF performs substantially better than all competing methods, over all datasets. Note that even a small improvement in nDCG may result in an important improvement in terms of the other recommendation quality measures, this is due to the log factor in nDCG.

Beyond the fact that the proposed approach outperforms competing methods, several issues are raised below to better understand the effectiveness of *Social-movMFs* and characterize the circumstances where it provides the most significant improvements.

Is it beneficial to model CF data as directional data distributed on the surface of a unit-hypersphere?
We observe that, even if the traditional vMF mixture model (movMF) does not take into account the social interactions between users, it is superior to the other competing methods in almost all situations, except on Flixster where SoReg offers the best performance among the competing methods. This provides strong empirical support for the advantage of modeling collaborative filtering data as directional data lying on a unit-hypersphere.

What is the impact of the social component of Soc-movMF?
Recall that movMF arises as a special case from Soc-movMF when $\lambda = 0$, i.e., movMF is exactly the Soc-movMF without the social component. Column "Improve", in Table 2, shows that the proposed Soc-movMF noticeably improves the performances of movMF. This constitutes empirical evidence that accounting for social interactions among users helps to improve the recommendation quality.

Why does the improvement rate, in the recommendation quality, differ substantially from one dataset to another?
The improvement rate on Flixster is significantly more important than on the other datasets, while the improvement on FilmTrust is quite low. We believe that this behaviour is due to the characteristics of the different datasets in terms of the number of observed ratings and social relations per user. In Fig. 5 we report the proportion of

Table 2 Comparison of average recommendation accuracy over different datasets. "Improve" indicates the improvement reached by the proposed social model `Soc-movMF` relative to performance of the traditional `movMF` model

Data	Measure	TrustSVD	TrustMF	SocialMF	RSTE	SoRec	SoReg	movMF	Soc-movMF	Improve (%)
FilmTrust	nDCG	0.1764	0.1679	0.2040	0.4612	0.1833	0.4658	0.6291	0.6626	5.05
	MRR	0.0142	0.0142	0.0799	0.3926	0.0829	0.3664	0.5843	0.6311	7.41
	Prec@5	0.0034	0.0124	0.0176	0.2160	0.0168	0.2380	0.3857	0.4030	4.29
	Rec@5	0.0029	0.0069	0.0188	0.1935	0.0190	0.2024	0.3771	0.4040	6.66
	Prec@10	0.0029	0.0090	0.0117	0.1807	0.0087	0.2069	0.3218	0.3385	4.93
	Rec@10	0.0048	0.0219	0.0242	0.3215	0.0194	0.3391	0.5785	0.6215	6.92
CiaoDVD	nDCG	0.1085	0.1023	0.1149	0.1236	0.1149	0.1283	0.1451	0.1527	4.98
	MRR	0.0062	0.0062	0.0124	0.0214	0.0122	0.0139	0.0252	0.0335	24.8
	Prec@5	0.0013	0.0017	0.0028	0.0061	0.0027	0.0028	0.0065	0.0088	26.1
	Rec@5	0.0029	0.0057	0.0081	0.0195	0.0079	0.0106	0.0211	0.0267	21.0
	Prec@10	0.0012	0.0014	0.0024	0.0059	0.0025	0.0024	0.0059	0.0077	23.4
	Rec@10	0.0059	0.0090	0.0141	0.0384	0.0143	0.0172	0.0351	0.0455	22.9
Ciao-280k	nDCG	0.1206	0.1117	0.1158	0.1094	0.1172	0.1157	0.1690	0.1820	7.14
	MRR	0.0079	0.0009	0.0053	0.0005	0.0068	0.0065	0.0653	0.0907	28.0
	Prec@5	0.0017	0.0001	0.0012	0.0001	0.0014	0.0012	0.0249	0.0320	22.2
	Rec@5	0.0009	0.0001	0.0008	0.0001	0.0011	0.0016	0.0149	0.0231	35.5
	Prec@10	0.0012	0.0001	0.0009	0.0001	0.0012	0.0006	0.0199	0.0256	22.3
	Rec@10	0.0016	0.0002	0.0013	0.0002	0.0019	0.0016	0.0243	0.0380	36.1
Epinions	nDCG	0.1071	0.1004	0.1021	0.0953	0.0944	0.1113	0.1438	0.1508	4.64
	MRR	0.0037	0.0094	0.0045	0.0016	0.0021	0.0047	0.0263	0.0409	35.7
	Prec@5	0.0005	0.0030	0.0008	0.0003	0.0005	0.0008	0.0071	0.0119	40.3
	Rec@5	0.0005	0.0038	0.0009	0.0005	0.0006	0.0013	0.0082	0.0148	44.6

Table 2 continued

Data	Measure	TrustSVD	TrustMF	SocialMF	RSTE	SoRec	SoReg	movMF	Soc-movMF	Improve (%)
	Prec@10	0.0006	0.0021	0.0009	0.0002	0.0004	0.0005	0.0069	0.0097	28.9
	Rec@10	0.0013	0.0053	0.0022	0.0006	0.0010	0.0017	0.0169	0.0229	26.2
Flixster	nDCG	0.1800	0.1277	0.1399	0.1289	0.1397	0.2768	0.2072	0.3547	41.6
	MRR	0.0376	0.0005	0.0157	0.0004	0.0157	0.1237	0.0497	0.2382	79.1
	Prec@5	0.0114	0.0001	0.0031	0.0001	0.0031	0.0358	0.0135	0.1078	87.5
	Rec@5	0.0037	0.0000	0.0104	0.0000	0.0105	0.0608	0.0032	0.1109	97.1
	Prec@10	0.0137	0.0002	0.0016	0.0001	0.0016	0.0313	0.0113	0.0948	88.1
	Rec@10	0.0116	0.0000	0.0104	0.0000	0.0105	0.0924	0.0045	0.1635	97.2

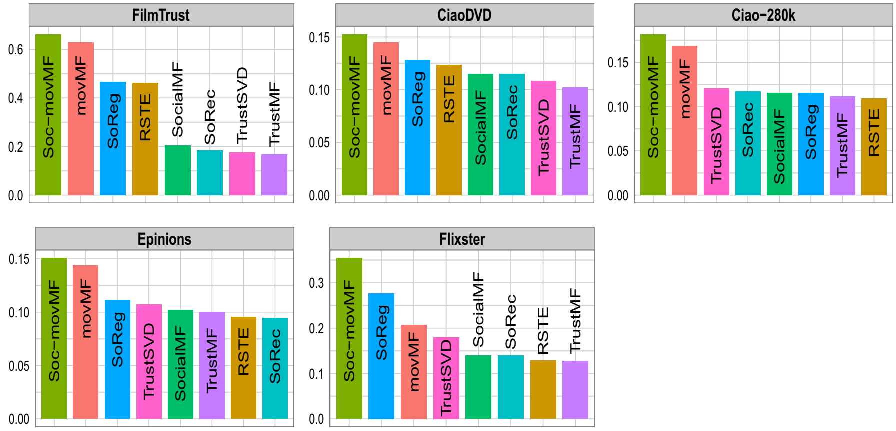

Fig. 1 Comparison of average nDCG over different datasets

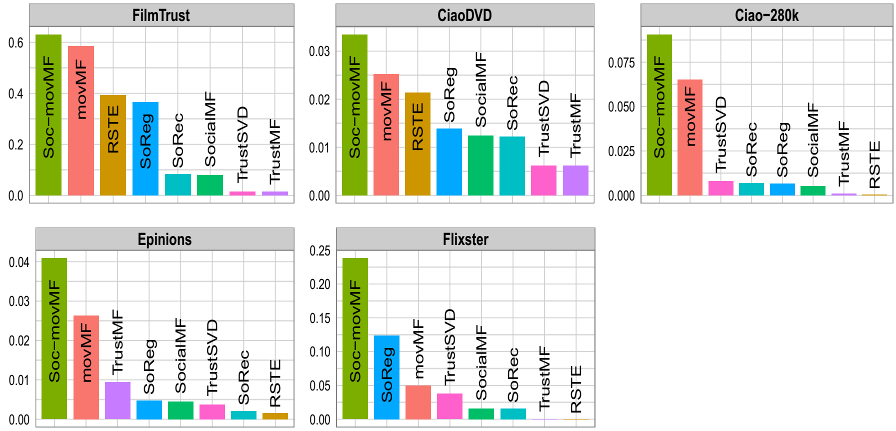

Fig. 2 Comparison of average MRR over different datasets

cold start users[7], who have expressed very few ratings (resp., social-relations), five or fewer, in the user-item matrix (resp., social network).

From Fig. 5 we note that Flixster exhibits different characteristics in comparison to the other datasets. In fact, in Flixster, while most users—more than 50%—are cold start users in the preference matrix, only few are cold start in the social network as opposed to the other datasets. This suggests that the social interactions in Flixster play a key role in handling the cold start users in the preference matrix, which allowed `Soc-movMF` to improve the performances of `movMF` by a noticeable amount.

[7] Cold start users are users who have expressed only few rating/social-interactions. Following previous works (Jamali and Ester 2010; Guo et al. 2015) we consider users who have expressed less than five ratings as cold start users in the preference matrix. Similarly, users who have less than five social relations are considered as cold start users in the social network.

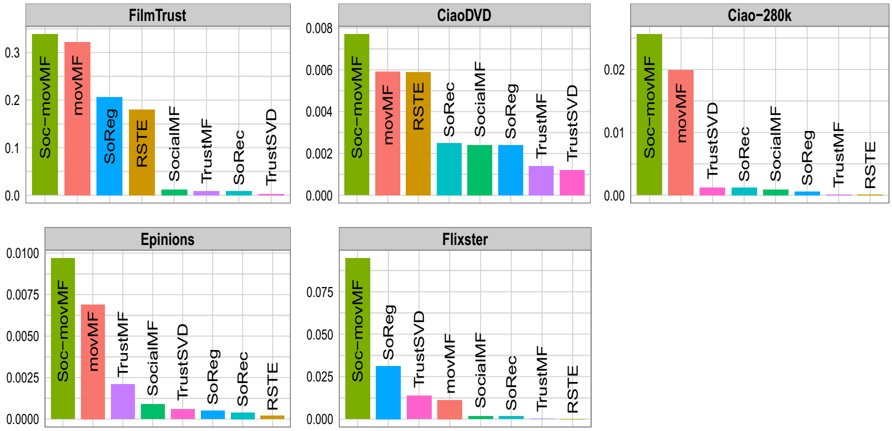

Fig. 3 Comparison of average Precision@10 over different datasets

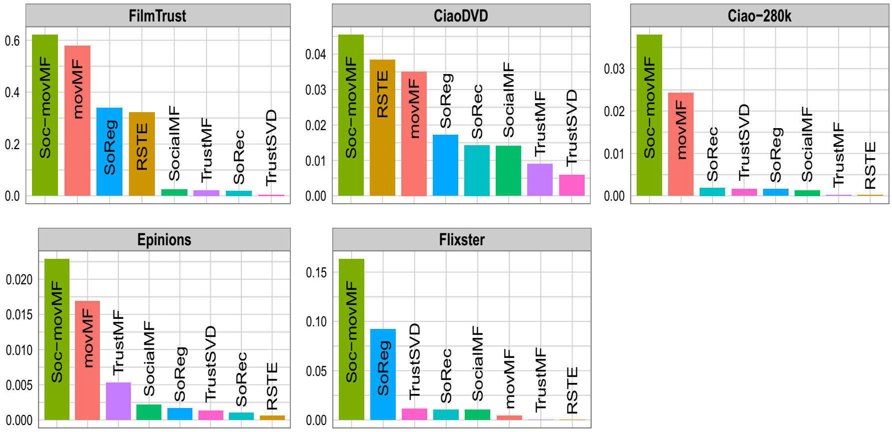

Fig. 4 Comparison of average Recall@10 over different datasets

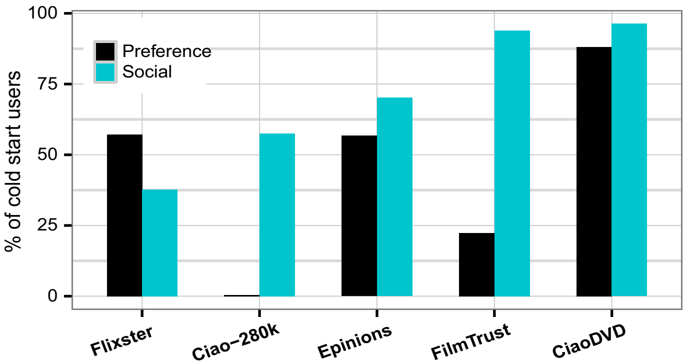

Fig. 5 Proportion of cold start users, in the user-item matrix (Preference) and social network (Social)

In FilmTrust, we note only few cold start users in the preference matrix, against a lot of cold start users—more than 90%—in the social network. Hence, most information in FilmTrust is contained in the user-item matrix, which may explain the low improvement in the performances of `Soc-movMF` relative to `movMF`.

In the Epinions and CiaoDVD datasets we observe a high rate of cold start users in both the user-item matrix and the social network. Although most users expressed only few social interactions in the above two, the social information seems to play an important role, in that it allowed `Soc-movMF` to reach better performances than `movMF`. We believe that this behaviour is due to the propagation of social interactions in `Soc-movMF`, see the observation below Algorithm 1.

Finally, the high performances of `Soc-movMF`, relative to `movMF`, on Ciao-280k suggest that, even when only few users are cold start users, in the preference matrix, the social information is still of great interest to make more accurate recommendations.

To sum up, the results from Table 2 suggest that, to make more accurate recommendations, not only the social interactions between users should be taken into account, but also the intrinsic directional properties of CF data. It seems better to model collaborative filtering data as directional data. Moreover, accounting for the social interactions among users seems to be of particular interest when most users have expressed very few ratings. In the next section, we shall investigate the latter result further.

5.5 movMFs versus social-movMFs on cold start users

Cold start is a major challenge in CF, because in many real-world applications most users express very few ratings. In order to complete the results from the previous section, we shall investigate in greater depth the impact of the social information on users who expressed very few ratings. We conduct, therefore, another series of experiments in which we benchmark our social model `Soc-movMF` against the traditional `movMF` model, on cold start users.

From Table 3 we can clearly observe that `Soc-movMF` still provides a high recommendation accuracy and substantially improves the performances of `movMF`. More interestingly, on the FilmTrust, Ciao-280*k* and Flixster data sets we observe a greater improvement, in the performance of `Soc-movMF` relative to `movMF`, compared with Table 2 (see column "Improve").

In order to understand why the improvement rate may differ substantially from one dataset to another, we report in Fig. 6 the distribution of out degrees per cold start user, i.e., the number of social relations expressed by a cold start user, over the different datasets. From Table 3 and Fig. 6, we observe that the improvement rate in the performances of `Soc-movMF`, relative to `movMF`, goes from high to low as the distribution of the number of social interactions, per cold start user, decreases. For instance on Flixster, we observe that most cold start users expressed more than five social relationships, which may explain the strong superiority of social `Soc-movMF` in comparison to `movMF`.

These results, constitute empirical evidence, that accounting for social interactions among users is of great interest and helps to alleviate the cold start issue.

Table 3 Comparison of average recommendation accuracy on cold start users—with 5 or fewer ratings; "Improve" indicates the improvement reached by `Soc-movMF` relative to the performance of `movMF`

Datasets	Measures	`movMF`	`Soc-movMF`	Improve (%)
FilmTrust	nDCG	0.4534	0.4663	2.75
	MRR	0.3457	0.3572	3.21
	Prec@5	0.1202	0.1336	10.1
	Rec@5	0.3974	0.4334	8.29
	Prec@10	0.0731	0.0843	13.3
	Rec@10	0.4760	0.5471	13.0
CiaoDVD	nDCG	0.1409	0.1459	3.40
	MRR	0.0271	0.0317	14.5
	Prec@5	0.0068	0.0077	11.2
	Rec@5	0.0289	0.0316	8.28
	Prec@10	0.0058	0.0062	6.62
	Rec@10	0.0487	0.0519	6.09
Ciao-280k	nDCG	0.1203	0.1344	10.5
	MRR	0.0201	0.0431	53.3
	Prec@5	0.0059	0.0110	45.9
	Rec@5	0.0132	0.0282	53.1
	Prec@10	0.0041	0.0073	42.9
	Rec@10	0.0185	0.0375	50.6
Epinions	nDCG	0.1102	0.1127	2.17
	MRR	0.0125	0.0161	22.0
	Prec@5	0.0026	0.0035	26.4
	Rec@5	0.0094	0.0123	23.4
	Prec@10	0.0022	0.0032	31.1
	Rec@10	0.0152	0.0218	30.1
Flixster	nDCG	0.1142	0.3106	63.2
	MRR	0.0047	0.1848	97.5
	Prec@5	0.0004	0.0538	99.2
	Rec@5	0.0016	0.2233	99.3
	Prec@10	0.0003	0.0394	99.2
	Rec@10	0.0023	0.3209	99.3

5.6 Impact of the number of clusters and the regularization parameters

We investigate the impact of the two parameters g and λ on the performances of `Soc-movMF`. We illustrate their behavior on FilmTrust, CiaoDVD, Ciao-280k and Epinions in terms of nDCG. In Fig. 7 the values of nDCG are depicted as a function of the number of clusters g, over the different datasets. We observe that a small number of clusters ($<$10) seems to be enough to reach high recommendation performances.[8] Figure 8 illustrates the impact of the regularization parameter λ. As it is clear from

[8] We observed the same behaviour on the Flixster dataset, not reported here for presentation purpose.

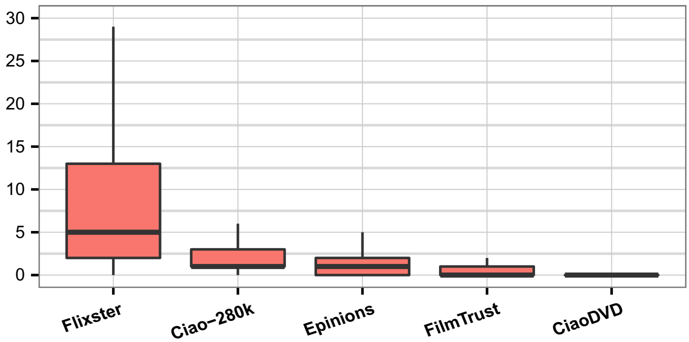

Fig. 6 Cold start users: distribution of out degree, i.e., the number of social interactions per cold start user

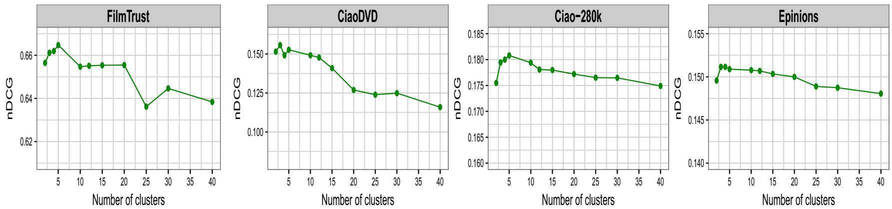

Fig. 7 Impact of the number of clusters g

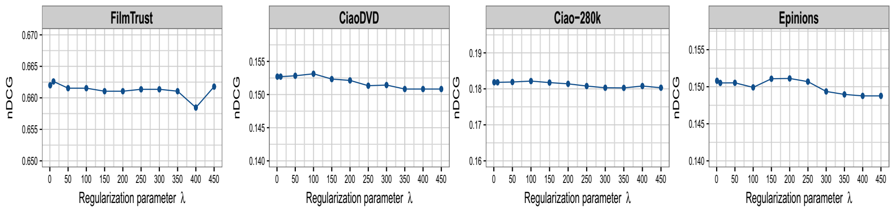

Fig. 8 Impact of the regularization parameter λ

this figure, `Soc-movMF` is highly stable relative to the variations of the regularization parameter λ and seems to provide slightly better performances with a small value of λ, which facilitates the setting of this parameter.

6 Conclusion

We proposed *Social-movMFs*, a novel model that accounts for social network information to improve item recommendations. *Social-movMFs* simultaneously seeks groups of users who tend to express similar preferences and brings the distributions, over clusters, of socially connected users closer to each other so as to capture the influences between friends. While existing approaches to social CF are based on popular modelling assumptions, such as Gaussian, our approach builds on the vMF distribution which arises naturally for data distributed on the surface of a unit-hypersphere. From our experiments, it seems that CF datasets possess intrinsic directional characteristics that are consistent with the vMF modeling assumption. Moreover, incorporating social information into a vMF mixture model turns out to be very beneficial and makes it possible to alleviate the sparsity related issues, such as the cold start problem.

In terms of performance the proposed model improves noticeably the recommendation accuracy of several strong competing methods, including the traditional `movMF` and several social CF models, as illustrated in our experiments. Our empirical results suggest that, for making good recommendations, not only the social interactions among users should be taken into account, but also the intrinsic "directional" properties of CF data.

The good performances of `Soc-movMF` motivates future investigations, that may include incorporating time into *Social-movMFs* and building online variants so as to handle the frequent changes in social CF: new ratings, social relations, items and users. Another line of future work is to extend *Social-movMFs* to the context of co-clustering, by relying on the block vMF mixture model (Salah et al. 2016b, c), so as to partition the sets of users and items simultaneously. Such an extension would allow us to alleviate the sparsity problem even better since the co-clustering (Dhillon et al. 2003; Nadif and Govaert 2010; Govaert and Nadif 2013, 2016) has proven to be very effective in the context of high dimensional sparse data.

References

Amatriain X, Castells P, de Vries A, Posse C (2012) Workshop on recommendation utility evaluation: beyond RMSE–RUE 2012. In: ACM conference on recommender systems (RecSys), pp 351–352

Banerjee A, Dhillon IS, Ghosh J, Sra S (2005) Clustering on the unit hypersphere using von Mises–Fisher distributions. J Mach Learn Res 6:1345–1382

Barbieri N, Manco G, Ritacco E (2014) Probabilistic approaches to recommendations. Synth Lect Data Min Knowl Discov 5(2):1–197

Belkin M, Niyogi P, Sindhwani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7:2399–2434

Bobadilla J, Ortega F, Hernando A, Gutirrez A (2013) Recommender systems survey. Knowl Based Syst 46:109–132

Cai D, Mei Q, Han J, Zhai C (2008) Modeling hidden topics on document manifold. In: Proceedings of the ACM conference on information and knowledge management, pp 911–920

Chaney AJ, Blei DM, Eliassi-Rad T (2015) A probabilistic model for using social networks in personalized item recommendation. In: ACM conference on recommender systems (RecSys), pp 43–50

Cremonesi P, Koren Y, Turrin R (2010) Performance of recommender algorithms on top-n recommendation tasks. In: ACM conference on recommender systems (RecSys), pp 39–46

Delporte J, Karatzoglou A, Matuszczyk T, Canu S (2013) Socially enabled preference learning from implicit feedback data. In: Joint european conference on machine learning and knowledge discovery in databases (ECML PKDD), Springer, Berlin, pp 145–160

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Methodol) 39:1–38

Dhillon IS, Modha DS (2001) Concept decompositions for large sparse text data using clustering. Mach Learn 42(1–2):143–175

Dhillon IS, Mallela S, Modha DS (2003) Information-theoretic co-clustering. In: Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining, pp 89–98

Gopal S, Yang Y (2014) Von Mises–Fisher clustering models. In: Proceedings of the international conference on machine learning (ICML), pp 154–162

Govaert G, Nadif M (2013) Co-Clustering. Wiley, New York

Govaert G, Nadif M (2016) Mutual information, phi-squared and model-based co-clustering for contingency tables. Adv Data Anal Classif. doi:10.1007/s11634-016-0274-6

Guo G, Zhang J, Yorke-Smith N (2013) A novel Bayesian similarity measure for recommender systems. In: Proceedings of the international joint conference on artificial intelligence (IJCAI), pp 2619–2625
Guo G, Zhang J, Thalmann D, Yorke-Smith N (2014) ETAF: an extended trust antecedents framework for trust prediction. In: IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), pp 540–547
Guo G, Zhang J, Yorke-Smith N (2015) TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: Proceedings of the international joint conference on artificial intelligence (AAAI), pp 123–129
He X, Cai D, Shao Y, Bao H, Han J (2011) Laplacian regularized gaussian mixture model for data clustering. IEEE Trans Knowl Data Eng (TKDE) 23(9):1406–1418
Jamali M, Ester M (2010) A matrix factorization technique with trust propagation for recommendation in social networks. In: ACM conference on recommender systems (RecSys), pp 135–142
Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, ACM, pp 426–434
Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37
Le T, Lauw HW (2014) Semantic visualization for spherical representation. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp 1007–1016
Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput 7(1):76–80
Liu H, Hu Z, Mian A, Tian H, Zhu X (2014) A new user similarity model to improve the accuracy of collaborative filtering. Knowl Based Syst 56:156–166
Loiacono D, Lommatzsch A, Turrin R (2014) An analysis of the 2014 RecSys challenge. In: ACM conference on recommender systems (RecSys), p 1
Ma H, Yang H, Lyu MR, King I (2008) Sorec: social recommendation using probabilistic matrix factorization. In: Proceedings of the ACM international on conference on information and knowledge management (CIKM), pp 931–940
Ma H, King I, Lyu MR (2009) Learning to recommend with social trust ensemble. In: Proceedings of the international ACM SIGIR conference on research and development in information retrieval, ACM, pp 203–210
Ma H, Zhou D, Liu C, Lyu MR, King I (2011) Recommender systems with social regularization. In: Proceedings of the ACM WSDM international conference on web search and data mining, pp 287–296
Mardia K, Jupp P (2009) Directional statistics. Wiley Series in Probability and Statistics. Wiley, New York
McLachlan G, Krishnan T (2007) The EM algorithm and extensions, vol 382. Wiley, New York
McLachlan G, Peel D (2004) Finite mixture models. Wiley, New York
Mei Q, Cai D, Zhang D, Zhai C (2008) Topic modeling with network regularization. In: Proceedings of the international conference on world wide web (WWW), pp 101–110
Nadif M, Govaert G (2010) Model-based co-clustering for continuous data. In: Proceedings of international conference on machine learning and applications (ICMLA), pp 175–180
Reisinger J, Waters A, Silverthorn B, Mooney RJ (2010) Spherical topic models. In: Proceedings of the international conference on machine learning (ICML), pp 903–910
Salah A, Rogovschi N, Nadif M (2016a) A dynamic collaborative filtering system via a weighted clustering approach. Neurocomputing 175:206–215
Salah A, Rogovschi N, Nadif M (2016b) Model-based co-clustering for high dimensional sparse data. In: Proceedings of the 19th international conference on artificial intelligence and statistics (AISTATS), pp 866–874
Salah A, Rogovschi N, Nadif M (2016c) Stochastic co-clustering for document-term data. In: Proceedings of the SIAM SDM international conference on data mining, pp 306–314
Salakhutdinov R, Mnih A (2008) Probabilistic matrix factorization. Adv Neural Inf Process Syst (NIPS) 20:1257–1264
Sarwar B, Karypis G, Konstan J, Riedl J (2000) Application of dimensionality reduction in recommender system-a case study. Technical Report, DTIC Document
Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the international conference on world wide web (WWW), ACM, pp 285–295

Sra S (2012) A short note on parameter approximation for von Mises–Fisher distributions: and a fast implementation of I s (x). Comput Stat 27(1):177–190

Tanabe A, Fukumizu K, Oba S, Takenouchi T, Ishii S (2007) Parameter estimation for von Mises–Fisher distributions. Comput Stat 22(1):145–157

Tang J, Gao H, Liu H (2012) mTrust: discerning multi-faceted trust in a connected world. In: Proceedings of the ACM WSDM international conference on web search and data mining, pp 93–102

Ungar LH, Foster DP (1998) Clustering methods for collaborative filtering. AAAI workshop on recommendation systems, vol 1, pp 114–129

Yang B, Lei Y, Liu D, Liu J (2013) Social collaborative filtering by trust. In: Proceedings of the international joint conference on artificial intelligence (AAAI), pp 2747–2753

Zhu X, Lafferty J (2005) Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In: Proceedings of the international conference on machine learning (ICML), pp 1052–1059