IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 10, 2017, accepted December 20, 2017, date of publication December 29, 2017,
date of current version February 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2788138

Recommender Systems Clustering Using Bayesian Non Negative Matrix Factorization

JESÚS BOBADILLA[1], RODOLFO BOJORQUE[2], ANTONIO HERNANDO ESTEBAN[1], AND REMIGIO HURTADO[2]

[1]Department of Information Systems, Universidad Politécnica de Madrid, 28031 Madrid, Spain
[2]Department of Computer Science, Universidad Politécnica Salesiana, 010102 Cuenca, Ecuador

Corresponding author: Jesús Bobadilla (jesus.bobadilla@upm.es)

ABSTRACT Recommender Systems present a high-level of sparsity in their ratings matrices. The collaborative filtering sparse data makes it difficult to: 1) compare elements using memory-based solutions; 2) obtain precise models using model-based solutions; 3) get accurate predictions; and 4) properly cluster elements. We propose the use of a Bayesian non-negative matrix factorization (BNMF) method to improve the current clustering results in the collaborative filtering area. We also provide an original pre-clustering algorithm adapted to the proposed probabilistic method. Results obtained using several open data sets show: 1) a conclusive clustering quality improvement when BNMF is used, compared with the classical matrix factorization or to the improved KMeans results; 2) a higher predictions accuracy using matrix factorization-based methods than using improved KMeans; and 3) better BNMF execution times compared with those of the classic matrix factorization, and an additional improvement when using the proposed pre-clustering algorithm.

INDEX TERMS Bayesian NMF, collaborative filtering, hard clustering, matrix factorization, pre-clustering, recommender systems, sparse data.

I. INTRODUCTION

A. MATRIX FACTORIZATION BASED RECOMMENDER SYSTEMS

The Collaborative Filtering (CF) Recommender Systems (RS), in their early days, mainly used memory-based methods, such as K Nearest Neighbours (KNN). These methods proved to be too sensitive to the degree of sparsity presented by RS datasets. Currently, model-based RSs provide more accurate results, are more scalable, and better address the data sparsity problem [1]. The mostly adopted model by modern RS is the Matrix Factorization (MF) [2], and recently its Nonnegative Matrix Factorization (NMF) variant. NMF was first introduced and popularized as a low-rank matrix approximation technique [3]. It is one of the most typical decomposition tools for extracting key features from the source matrix. Currently, NMF is widely used into machine learning processes and it is been applied in numerous applications: Language processing [4], image retrieval [5], computational biology [6], large-scale networks [7], audio signals [8], collaborative filtering [9], etc.

New RS applications and approaches are emerging every day, using factorization techniques to provide new solutions in the CF field or to improve existing ones [10]. The following is a series of recent studies using Matrix Factorization (MF) or NMF in the field of CF RS: The MF based framework [11] provides a better use the intrinsic structure of user-item rating matrix and content-based information. Using the MF based CF as the best option for group RS has been reported in [12]. The maximum margin MF techniques when the rating matrix contains only two levels (e.g. like/dislike) is used in [13]. MF in context-aware CF RS has been used [14], specifically using a dataset with geographic information and another dataset with musical information. A generic framework to preserving privacy MF by utilizing differential privacy [15] has been addressed. High performance healthcare recommendation model [16] has been provided. It uses the topic model based approach to discover user preference distribution and they incorporate topic model and emotional offset into the MF. A non-negative multiple MF with social similarity for RS [17] considers

the similarities between users, the relationships of users-resources and the relationships of tag-resources. The novel hybrid web service QoS prediction approach [18] systematically combines the memory-based CF and model-based CF, using NMF and Expectation-Maximization techniques.

NMF is used in a large number of applications as a machine learning method. NMF is a blind source separation technique decomposing multivariate data sets into meaningful non-negative factors. A really promising approach is to incorporate regular statistical support into the NMF model. In this way it is possible to associate probabilities with the predictions and recommendations themselves, as well as to make use of the inherent properties of the probabilistic approach. In short, incorporating Bayesian theory into the NMF model enriches its results and also its interpretation.

The following studies show the current use of the Bayesian approach into the NMF factorization model, and its application to the CF RS: Two novel latent factor models [19] incorporate both socially-influenced feature value discrepancies and socially-influenced conditional feature value discrepancies. This method models the user preferences as a Bayesian Network. The real log canonical threshold of NMF is tudied in [20]; they give an upper bound of the generalization error in Bayesian learning. Results show that the generalization error of the MF can be made smaller than regular statistical models if Bayesian learning is applied. Uniqueness is an open NMF problem addressed in [21], where authors propose a Bayesian optimality criterion for NMF solutions. It can be delivered in the absence of prior knowledge. The NMF model order determination problems [22] determine the capability and accuracy of data structure discovering. Authors propose a method based on hierarchical Bayesian inference model, maximum a posteriori criterion and non-informative parameters. A probabilistic MF scaling linearly with the number of observations has been introduced in [23]. It includes an adaptive prior on the model parameters.

B. RECOMMENDER SYSTEMS CLUSTERING

Data clustering [24] presents three main aspects:

1) Methods: techniques commonly used.
2) Domains: raw data types (text, multimedia, ratings, streams, biological, etc.).
3) Variations: cluster validation, cluster ensembles, etc.

RS research mainly uses ratings data as source information. RS can incorporate a large amount of additional information to ratings (demographic, content-based, context-aware, social, etc.); however, the common denominator for all modern RS is its ratings matrix. Effective clustering based on the ratings matrix will be useful for any RS, and therefore this approach can be considered as universal. The data type (domain) determines, to a large extent, the set of clustering methods that best fit that domain. The main types of clustering used in the field of RS are: a) distance based algorithms and b) dimensionality reduction methods. Feature selection methods and density-based approaches are much less widely used methods in the RS field.

The following are some representative papers in the field of CF clustering: Two clustering based CF algoritms [25] are proposed: Item-based fuzzy clustering and trust-aware clustering; they obtain an increased value of coverage without affecting recommendation quality. By integrating the user clustering regularization term [26], the standard MF can be optimized. This work reports improvements in the recommendations accuracy, compared with standard algorithms. To dimension the number of clusters (K) is a process that requires experience, knowledge of the data and a trial and error mechanism to choose the most appropriate values. The correct choice of this parameter determines the quality of the resulting clusters, as well as the predictions and recommendations made; [27] dynamically sets the parameter: with more data coming in, the incremental clustering algorithm determines whether to increase the number of clusters or merging the existing clusters. Authors report encouraging prediction accuracy. Hu *et al.* [28] use a co-clustering method to divide the raw rating matrix into clusters, and then it employs NMF to make improved predictions of unknown ratings.

User preferences change over time. A method to model the change of preferences is based on taking into account the temporal features evolutions. An evolutionary clustering algorithm [29] is proposed. This algorithm performs an optimization of conflicting parameters instead of using the traditional evolutionary algorithms like genetic ones. An efficient incremental CF system [30] is proposed, based on weighted clustering approach; this method provides a very low computation cost. The recommendations of e-commerce products can be improved by clustering similar products [31]; recommendation work is then done with the resulting clusters. The [32] research paper applies the users' implicit interaction records with items to efficiently process massive data by employing association rules mining. The clustering technique has been employed to reduce the size of data and dimensionality of the item space as the performance of association rules mining. A network model [33] is made feeding with items and users regarded as heterogeneous individuals. According to the constructed network model, states of individuals evolve over time. Individuals with higher scores cluster together and individuals with lower scores got away.

Centroid selection in k-means based RS [34] can improve performance as well as being cost saving. The proposed centroid selection method has the ability to exploit underlying data correlation structures. [35] provides a k-means survey, where they summarize the existing methods, discuss the major challenges and point out some of the emerging research.

It is possible to formulate clustering as a matrix decomposition problem [36]. According to [24] and [37], MF has important advantages when used as a clustering method: 1) It can model widely varying data distributions due to the flexibility of matrix factorization [38], [39]; 2) It is able to perform simultaneous clustering of the rows (users) and the columns (items) of the input data matrix; and 3) It can simultaneously achieve both hard and soft clustering [40].

This paper delves into improving, simultaneously, the RS hard clustering results and the predictions accuracy.

A set of representative and current MF and NMF clustering papers is presented below: Latent information [38] in high dimensional data, using 3-factors NMF as a coclustering method can be extracted. They discover a clean correlation structure between document and terms clusters; this correlation structure can be used, as starting point, for some other machine learning method. Co-clustering has been proposed into the collaborative filtering field to discover subspace correlations from big data environments [41], [42]. An NMF-based constrained clustering framework [37] is proposed in which there are constraints in the form of must-link and cannot-link (semi-supervised co-clustering [43]).

When datasets incorporate some specific constraint information, it is possible to fuse both the data distribution information and the constraint information to be processed using a constrained clustering algorithm. When the data incorporates constraints, the constrained clustering algorithms provide better performance; however its use is not universal, because it only makes sense to apply them in datasets that meet the constraint restrictions. The regularized NMF field introduces additional constraints to the standard NMF formulation, such as local learning regularizations based on neighbourhoods [44]. The clustering method we propose is not based on any specific restrictions and does not use additional information, so it can be used on any type of CF ratings matrix. A RS clustering based on genres [45] uses NMF to obtain predictions and recommendations. They report improvements both in prediction and recommendation quality. To combine each cluster result is the challenge they have faced using weighting strategies.

C. MOTIVATION AND HYPOTHESIS

Organizing data into sensible groupings is one of the most fundamental modes of understanding and learning [35]. Grouping, or clustering, is made according to measured intrinsic characteristics or similarity. The research in the field of CF RS clustering has focused on the objective of detecting groups of users or groups of items, in order to perform a differentiated process of CF in each group of users or items. The advantages that are presented as results of these studies are: a) accuracy improvement, b) performance improvement [27]–[29], [31], [46]. These papers usually test the quality of the clusters in an indirect way: verifying the quality improvement of the predictions and the recommendations.

The use of RS model-based methods, specifically the MF and the NMF, has reduced the importance of the CF method performance, due to the model speed prediction when it has already been trained. However, due to the increasing demand for big data results [41], [42], [47], the process of RS clustering takes on importance in itself as a source of data analytics information. Consequently, the process of clustering CF ratings matrices is an increasingly important field, not only as a means to improve quality recommendations, but also to provide reliable analytics, which can be incorporated into value-added business actions.

The MF techniques provide great clustering results when applied to sparse data, such as those in the RS datasets [37], [42], [43]. NMF can model widely varying data distributions due to the flexibility of MF as compared to the rigid spherical clusters that the K-means clustering objective function attempts to capture. When the data distribution is far from a spherical clustering, NMF may have advantages. NMF scalability can be effectively addressed through different schemes: Shrinking, partitioning [48], incremental [49] and parallel [50], making it possible to perform clustering of big data CF matrices.

Current research points to the MF methods as the most suitable for performing clustering in sparse datasets, such as the RS ones. However, the assignment of each user (or item) to its corresponding cluster is not always sufficiently precise: the problem is that the different values of the hidden factors may be too similar; In such situations you can not make a convincing decision about which cluster to assign to each user or item.

The preliminary hypothesis of this paper is that the quality of clustering in RS will improve if Bayesian NMF is used: this approach enriches the NMF model, providing a probabilistic basis. In this way, we can use the probabilistic parameters to fine-tune the allocation of each user (or item) to each cluster, and thus to improve the whole RS clustering.

As far as we know, there is not a relevant paper providing a comprehensive study of CF clustering based on Bayesian NMF. However, there are indeed several Bayesian NMF methods published [21], [22], [51]–[53]. Our paper takes as a reference the Bayesian MF method [54]. It is based on factorizing the ratings matrix into two non-negative matrices whose components have an understandable probabilistic meaning. The mathematical foundations of BNMF and its good behaviour as CF method are established in [54]. We focus on checking its superiority as a hard clustering method when applied to CF datasets.

The rest of the paper is structured as follows: section II explains the Bayesian non-Negative Matrix Factorization (BNMF) method and presents a running example to easily understand the choice of the proposed method and the customized pre-clustering algorithm. Section III focuses on pre-clustering: the centroid selection and the BNMF initialization of their learning parameters. Section IV introduces the experiments designed for the paper and it explains the obtained results. Finally, section V summarizes the conclusions of the paper and it proposes promising future works.

II. METHOD

This section is divided into two subsections:

1) Method design and formulation: Summary of the fundamentals of the BNMF method [54], the meaning of its most important parameters, the reason why it adequately clusters RS and the mathematical formulation that implements it.

2) Motivation: We use a data-toy running example to illustrate the main concepts. This running example shows the weakness of existing clustering methods when applied to RS datasets. We also show the improvement of results by using the BNMF proposed method. The motivation sequence starts from the simplest and best known clustering method (Kmeans) and continues by selecting the most promising centroids (LogUser Power method); then, the Matrix Factorization model is used. Finally, results are obtained by applying the proposed method. The running example shows the gradual improvements that each of the tested methods provide, and the superior quality that is achieved when using the proposed one.

A. METHOD DESIGN AND FORMULATION

Here we will summarize the probabilistic model discussed in [54] and the algorithm based on this model for finding out clusters of users sharing the same tastes.

The algorithm for finding out clusters is based on a generative probabilistic model; this model is graphically showed into Fig. 1: Circles represent random variables, arrows between two variables indicate dependence between them. A grey circle indicates that the value of this random variable is observed. The small black squares represent parameters of the model.

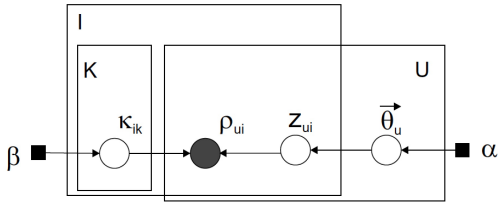

FIGURE 1. Graphical model representation of our probabilistic approach.

The probabilistic model considers that users can be clustered in K sets according to their tastes. This probabilistic model allows us to simulate the ratings that users make over the items by means of the following procedure:

1) For each user u, we sample a K dimensional vector random variable ϕ_u from a Dirichlet distribution:

$$\vec{\phi}_u \sim Dir(\alpha_1, \ldots, \alpha_K) \tag{1}$$

- The vector $(\phi_{u,1}, \ldots, \phi_{u,K})$ represents the probability that a user belongs to each cluster.
- α is a parameter of the model. It indicates the prior knowledge of the overlapping degree between clusters.

2) For each item i and each factor k, we sample a random variable $K_{i,k}$ from a Beta distribution (which takes values from 0 to 1):

$$K_{i,k} \sim Beta(\beta, \beta) \tag{2}$$

- The value $K_{i,k}$ represents the probability that users in cluster k like the item i.
- β is a parameter of the model and it indicates the amount of evidence that the algorithm requires to deduce that a group of users likes an item.

3) For the user u and each item i such that $r_{u,i} \neq \bullet$ (not voted), we sample the following random variables:

- The random variable $z_{u,i}$ from a categorical distribution (which takes values in the range $\{1, \ldots, K\}$).

$$z_{u,i} \sim Cat(\vec{\phi}_u) \tag{3}$$

- A value k in $z_{u,i}$ represents that the user u behaves like users in cluster k when consuming the item i.
- The random variable $\rho_{u,i}$ from a Binomial distribution.

$$\rho_{u,i} \sim Bin(R, K_{i,z_{u,i}}) \tag{4}$$

Parameter R is highly related to the ratings that the user u has made over item i. Indeed, the normalized rating $(r^*_{u,i})$ that the user u has made over the item i is $\rho_{u,i}/R$.

$$R = MaxRatingRange - IntervalRatingRange \tag{5}$$

e.g. using Movielens and Netflix: $R = 5 - 1 = 4$, whereas using FilmTrust: $R = 4 - 0.5 = 3.5$ (FilmTrust ratings go from 0 to 4 step 0.5; Movielens & Netflix ratings go from 1 to 5, step 1).

Following this procedure, we can simulate samples of plausible ratings. We observe the value of the random variables $\rho_{u,i}$ (grey circle into Fig. 1). The rest of variables (white circles in Fig. 1) are unknown for us. The important issue about the model lies in finding out the posterior distribution of the unknown random variables $\vec{\phi}_u$ (which are sometimes called latent variables), indicating the posterior probability that the user belongs to each cluster. Like almost all probabilistic graphical models, the probabilistic model considered does not allow to computationally calculate the exact conditional distributions in an analytical way. An algorithm based on the variational inference technique is adopted to calculate this posterior probability distribution. The description of the algorithm as a black box is:

- Input. The input of the algorithm is a matrix of ratings, since we are focusing on CF RS. We will use the following notation related to this rating matrix:
  - N: Number of users in the ratings matrix.
  - M: Number of items in the ratings matrix.
  - $r_{u,i}$: Rating that user u has made over the item i. So that our algorithm can be used for any RS with different scales, we will consider the normalized ratings $r_{u,i}$ which lie within [0, 1]. We will use the notation $r_{u,i} = \bullet$ to indicate that user u has not rated the item i yet.
  - Parameters. Besides the rating matrix, the technique also takes into account some parameters fixed

by the RS. Specifically it considers the following parameters:

* K. This parameter indicates the number of groups of users (clusters) that the algorithm is going to find out.
* $\alpha \in (0, 1)$. This parameter is related to the possibility of obtaining overlapping groups of users sharing the same tastes. A value very close to 0 involves that users tend to only belong to one group. A greater value indicates that we are allowing that the user may be in more than one group.
* $\beta > 1$: This parameter is related to the amount of evidence that the algorithm requires to deduce that a group of users likes an item. The higher β, the more evidence the model requires to deduce that a group of users likes or dislikes an item.

– Output. From the input data and the parameters values, the algorithm outputs the NxK matrix $(a_{u,k})$ associated to users. Each term $a_{u,k}$ of the matrix informs about the probability that user u belongs to each cluster k of users. The algorithm calculates some auxiliary variables $\gamma_{u,k}$, $\epsilon^-_{i,k}$, $\epsilon^+_{i,k}$ to finally calculate the values $a_{u,k}$. Steps:

* Initialize randomly $\gamma_{u,k}$
* Initialize randomly $\epsilon^-_{i,k}$
* Initialize randomly $\epsilon^+_{i,k}$
* Repeat until changes are not significant

  For each user u:

  For each item i rated by the user u in the training set:

  For each factor k: update $\lambda_{u,i,k}$ according to equations:

$$\lambda'_{u,i,k} = exp\left(\Psi(\gamma_{u,k}) + r^+_{u,i} \cdot \Psi(\epsilon^+_{i,k}) + r^-_{u,i} \cdot \Psi(\epsilon^-_{i,k}) - R \cdot \Psi(\epsilon^+_{i,k} + \epsilon^-_{i,k})\right)$$

$$\lambda_{u,i,k} = \frac{\lambda'_{u,i,k}}{\lambda'_{u,i,1} + \cdots + \lambda'_{u,i,K}} \tag{6}$$

  For each user u:

  For each factor k: update $\lambda_{u,i,k}$ according to equation:

$$\gamma_{u,k} = \alpha + \sum_{\{i|r_{u,i}\neq\bullet\}} \lambda_{u,i,k} \tag{7}$$

  For each item i rated by the user u in the training set: For each factor k: update $\epsilon^+_{i,k}$ according to equation:

$$\epsilon^+_{i,k} = \beta + \sum_{\{u|r_{u,i}\neq\bullet\}} \lambda_{u,i,k} \cdot r^+_{u,i} \tag{8}$$

  For each factor k: update $\epsilon^-_{i,k}$ according to equation:

$$\epsilon^-_{i,k} = \beta + \sum_{\{u|r_{u,i}\neq\bullet\}} \lambda_{u,i,k} \cdot r^-_{u,i} \tag{9}$$

where

Ψ is the digamma function defined as the logarithmic derivative of the gamma function:

$$\Psi(x) = (ln\,\Gamma(x))' = \frac{\Gamma'(x)}{\Gamma(x)} \tag{10}$$

$$r^+_{u,i} = \rho_{u,i} = R \cdot r^*_{u,i} \tag{11}$$

$$r^-_{u,i} = R - \rho_{u,i} = R \cdot \left(1 - r^*_{u,i}\right) \tag{12}$$

$R = maxRatingRange - 1$, $r^+_{u,i}$ & $r^-_{u,i} \in \{minRatingRange - 1, , maxRatingRange - 1\}$ e.g.: for datasets where ratings belongs to $\{1, 2, 3, 4, 5\}$: $minRatingRange = 1$, $maxRatingRange = 5$, $r^+_{u,i}$ & $r^-_{u,i} \in \{0, 1, 2, 3, 4\}$,$R = 4$. Output $a_{u,k}$:

$$a_{u,k} = \frac{\gamma_{u,k}}{\gamma_{u,1} + \ldots + \gamma_{u,K}} \tag{13}$$

Output $b_{k,i}$:

$$b_{k,i} = \frac{\epsilon^+_{i,k}}{\epsilon^+_{i,1} + \ldots + \epsilon^+_{i,K}} \tag{14}$$

It is important to highlight that this probabilistic method fulfils an important restriction about the users hidden factors:

$$\sum_{k=1}^{K} a_{u,k} = 1 \tag{15}$$

The mathematical proof of these equations can be found into the "Appendix" section of the BNMF paper [54]

B. MOTIVATION

In this section we intend to show, in a simple and intuitive way, the operation of the proposed method. We also explain, with examples, the improvements made to the most significant existing methods. We will support a didactic running example. This running example is based on a data-toy designed to show the clustering quality variations obtained in each case of study. The original data-toy is based on Fig. 2 rating matrix (12 users, 12 items):

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}
U_1		4	5			4			1		4	
U_2	5	5	5		5			1				4
U_3	4	4	5									
U_4		1		1	1	1		1			5	
U_5	2			2		2			2			4
U_6				1	1	1						
U_7	1						4	4	5			
U_8		2			4		5	5	5	4	5	
U_9				5			4		4			
U_{10}										1	2	2
U_{11}	1		1		4			1		2	1	
U_{12}		1				5			2	2	2	1

FIGURE 2. Data-toy ratings matrix.

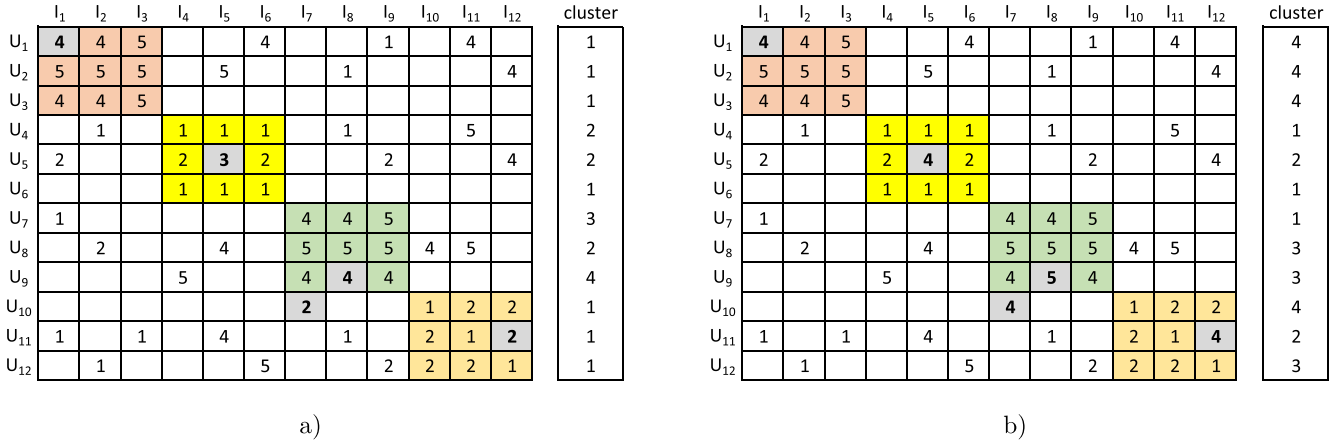

FIGURE 3. KMeans clustering and prediction results: a) KMeans, b) KMeans and Log power-users pre-clustering (KMeans+). Each different color into the figure corresponds to a cluster found using this method.

Into the data-toy we can see, intuitively, the existence of four groups: $G1 = U_1, U_2, U_3$, $G2 = U_4, U_5, U_6$, $G3 = U_7, U_8, U_9$, $G4 = U_{10}, U_{11}, U_{12}$. $G1$ users share a similar taste for items I_1, I_2, I_3 since $r_{u,i} > 4$. U_1 and U_8 have a similar taste for item 11; However, U_8 does not belong to G1 because it dislikes I_2. The same reasoning can be used to explain the existence of $G2$, $G3$ and $G4$. In the following sections, we will use several clustering methods to test their ability to detect the four predicted groups.

A major problem in most clustering methods is their sensitivity to initialization of parameters and to the choice of start centroids. Research literature reports clustering improvements when initialitation methods are applied to the iterative refinement clustering algorithms; the most cited algorithms to be initialized are: K-prototypes (mainly k-Means and k-Modes), expectation-maximitation (EM), fuzzy c-Means, k-Means++, spherical K-Means, and Classification EM (CEM). Poor choice of pre-clustering parameters and centroids can lead to local minima, poor clustering and insufficient prediction quality. This paper addresses both clustering and pre-clustering improvements.

1) KMeans AND PRE-CLUSTERING IMPROVED KMeans (KMeans+)

Fig. 3a shows a result obtained using the KMeans method. The column "cluster" shows the cluster in which each user has been classified: $G1 = U_1, U_2, U_3, U_6, U_{10}, U_{11}, U_{12}$, $G2 = U_4, U_5, U_8$, $G3 = U_7$, $G4 = U_9$. Users have not been classified as expected. Predictions $p_{1,1}$, $p_{9,8}$ and $p_{11,12}$ are correct, but prediction $p_{5,5}$ shows as "indifferent" (3) a value that was expected "not like" (1 or 2). The results obtained vary considerably according to which are the initial centroids with which the KMeans is executed.

Fig. 3b) shows the results obtained using KMeans clustering and "Log power-users" pre-clustering [34]. The users who have rated a large number of items in a RS are referred to as power users. Zahra *et al.* [34] use the power users to select the pre-clustering centroids, obtaining good RS results. The "Log" power users version is based on an specific probability function to choose centroids. [34] presents three probability versions: "Power", "ProbPower" and "Log Power", The latter is the one that has given us the best results.

Fig. 3b) shows an improvement in the choice of clusters: now, cluster 4 is perfectly done. However, the quality of clustering is still poor, and some prediction results are incorrect (e.g. $p_{11,12}$). Traditional clustering methods have little precision when applied to extremely sparse datasets, such as those in RS.

2) MATRIX FACTORIZATION

RS have evolved from using memory-based methods, such as the KNN to the use of model-based methods, such as the Matrix Factorization (MF). MF methods provide better prediction and recommendation results, total coverage and scalability. MF methods also provide clustering results, by analyzing their hidden factors.

Fig. 4a) shows the users hidden factors values in a MF of dimension $K = 4$. We assigned each user to the cluster (F1, F2, F3, F4) whose hidden value is higher; e.g: U_1 has been assigned to cluster 3, because its hidden value 1.87 is greater than its other three hidden values (0.87, 0.75, 0.04). U_4 has been assigned to cluster 4, for its hidden value 2.07, greater than -0.33, 0.47 and 0.06, and so on.

Fig. 4b) summarizes the users assignment to clusters (right column) and also shows some relevant predictions. We can see a significant improvement in the clustering results with respect to the KMeans+ solutions: in this case U_1 to U_9 users are correctly classified (U_{10} and U_{12} are not). Predictions are correct, except for $p_{9,8}$ which shows an "indifferent" value (3) instead of a "like" one (4 or 5).

Overall, MF clustering method improves the KMeans+ one when applied to sparse datasets; Nevertheless, there is

	F1	F2	F3	F4
U_1	0.87656387	0.75455194	1.8776912	0.04900811
U_2	0.95886239	1.34506115	1.84018616	-0.2336056
U_3	1.18185195	1.27677104	1.34879504	0.75105649
U_4	2.07212866	-0.3327408	0.47941458	0.06431205
U_5	1.59006798	0.5009905	0.13830185	0.27043082
U_6	0.31993822	0.27879877	0.23073946	0.10754167
U_7	0.65684201	1.19207029	0.16343754	2.05769024
U_8	0.81377341	0.79679385	0.7575215	2.47741228
U_9	1.09921661	1.2825001	1.02449528	1.35185257
U_{10}	0.68915641	0.33925499	0.28167438	0.14010675
U_{11}	-0.1734621	1.75024794	0.21648498	0.44181459
U_{12}	-0.0883087	0.19619245	0.82535583	1.41878189

a)

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	cluster
U_1	**5**	4	5			4			1		4		3
U_2	5	5	5		5			1				4	3
U_3	4	4	5										3
U_4		1		1	1	1		1			5		1
U_5	2			2	**2**	2			2			4	1
U_6				1	1	1							1
U_7	1						4	4	5				4
U_8		2			4		5	5	5	4	5		4
U_9				5			4	**3**	4				4
U_{10}							**1**			1	2	2	1
U_{11}	1		1		4			1		2	1	**1**	2
U_{12}		1				5			2	2	2	1	4

b)

FIGURE 4. Matrix Factorization clustering and prediction results: a) Users factors matrix, b) Clustering results. Each different color into the figure corresponds to a cluster found using this method.

	F1	F2	F3	F4
U_1	0.0185208	0.94443802	0.01852059	0.0185206
U_2	0.01562588	0.95312107	0.01562675	0.0156263
U_3	0.02941302	0.9117617	0.02941305	0.02941223
U_4	0.0156263	0.01562631	0.95312063	0.01562676
U_5	0.01852107	0.01852198	0.94443519	0.01852175
U_6	0.02941215	0.02941216	0.91176354	0.02941215
U_7	0.93181278	0.02272815	0.02272902	0.02273005
U_8	0.95945492	0.01351491	0.0135152	0.01351497
U_9	0.91175799	0.02941371	0.02941395	0.02941435
U_{10}	0.02941348	0.02941383	0.02941324	0.91175945
U_{11}	0.01562697	0.01562711	0.01562665	0.95311927
U_{12}	0.01562587	0.01562685	0.01562707	0.95312021

a)

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	cluster
U_1	**4**	4	5			4			1		4		2
U_2	5	5	5		5			1				4	2
U_3	4	4	5										2
U_4		1		1	1	1		1			5		3
U_5	2			2	**2**	2			2			4	3
U_6				1	1	1							3
U_7	1						4	4	5				1
U_8		2			4		5	5	5	4	5		1
U_9				5			4	**4**	4				1
U_{10}							**3**			1	2	2	4
U_{11}	1		1		4			1		2	1	**3**	4
U_{12}		1				5			2	2	2	1	4

b)

FIGURE 5. Bayesian Non Negative Matrix Factorization clustering and prediction results: a) Users factors matrix, b) Clustering results. Each different color into the figure corresponds to a cluster found using this method.

margin for improvement in both clustering and prediction results.

3) BAYESIAN NON-NEGATIVE MATRIX FACTORIZATION (BNMF)

The Bayesian NMF version we propose [54] improves the existing NMF and provides a probabilistic basis that classifies each user or item among the K factorization factors. This fine tune is expected to improve both MF and NMF clustering results.

Fig. 5 shows the results of our running example using Bayesian non Negative Matrix Factorization (BNMF). As it can be seen into graph a), results are correct: each user is assigned to its clustering group. Graph b) shows the clustering groups and predictions: Predictions (u_1,i_1), (u_5,i_5), (u_9,i_8) are accurate, whereas prediction (u_{11},i_{12}) is not.

4) PRE-CLUSTERING AND BAYESIAN NON-NEGATIVE MATRIX FACTORIZATION (BNMF+)

Pre-clustering is used for improving clustering accuracy as well as System performance. We will use KMeansPlus LogPower centroid selection [34] to initialize the learning parameters of our BNMF method. We will call BNMF+ to the sequence: 1) centroid selection, and 2) Bayesian NMF. Section III explains the details of the proposed pre-clustering algorithm and its customization to the BNMF method.

To illustrate the advantages of pre-clustering, we have added a user (u_{13}) to the rating matrix of our running example, making clustering a bit more complicated (Fig. 6).

The top table into Fig. 7 shows the results of three different runs of the BNMF method. These runs do not use any type of pre-clustering. Usually, a very accurate result is obtained, as shown in "Experiment 1". However, in some

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}
U_1		4	5			4			1		4	
U_2	5	5	5		5			1				4
U_3	4	4	5									
U_4		1		1	1	1		1			5	
U_5	2			2		2			2			4
U_6				1	1	1						
U_7	1						4	4	5			
U_8		2			4		5	5	5	4	5	
U_9				5			4		4			
U_{10}										1	2	2
U_{11}	1		1		4			1		2	1	
U_{12}		1				5			2	2	2	1
U_{13}	5	5	5				1	5	5			

FIGURE 6. Extended running example ratings matrix.

cases, the method is based on initial values, taken randomly, that lead to unsuitable solutions. This is the case for the runs labelled as ''Experiment 2'' and ''Experiment 3'' where the algorithm has reached local minimums.

If we use pre-clustering, we can reduce the probability of reaching inadequate solutions and local minimums. The chosen pre-clustering method uses a similarity measure to calculate the distances between the RS items (or users), and thus determine which are the most suitable initial centroids. Usually, the similarity measure chosen is Euclidean distance, however, in RS datasets, it is more appropriate to use similarity measures adapted to sparse matrices [55]. [56] explains that Pearson correlation is better than the cosine and Euclidean distance.

The table at the bottom of Fig. 7 shows the results of using centroid selection pre-clustering to determine the initial values of the parameters that feed the iterative BNMF algorithm. Left side shows results using the Mean Jaccard Difference (MJD) similarity measure, central position shows the obtained results using JMSD and right side shows Pearson correlation results. It can be seen that, in our example, MJD [57] produces the expected results, whereas JMSD [58] and Pearson correlation do not. Overall, using pre-clustering, we will be able to minimize the situations in which we obtain inadequate clustering results. As we will see later, into this paper, the pre-clustering we have designed improves the clustering quality and it decreases the execution times of the matrix factorization.

III. BAYESIAN NON NEGATIVE MATRIX FACTORIZATION PRE-CLUSTERING (BNMF+)

The basic clustering methods start from initial centroids, or from initial parameters, taken at random. From them, by means of an iterative process, the centroids and their clusters are modified. The machine learning process finishes when a condition of completion is obtained or when a maximum number of iterations have been reached.

The clustering quality results depend, to a great extent, on the choice of the centroids or the initial parameters with which the iterative process is fed. A poor choice of initial values can lead the algorithm to a local minimum. For this reason there are several RS-based approaches to choosing the initial centroids rather than randomly; e.g: Kmeans-, Kmeans+, Kmeans++, KmeansUserPower, etc. [34]. The sparse nature of the RS datasets makes it difficult to choose the initial centroids and the constituent elements of each cluster. Since each user only votes a very small number of the available items, most of the pairs <user, item>, into the rating matrix, are empty. This is a circumstance that does not exist in most clustering situations. To fill in the rating matrix with zero values, or with intermediate values, is not an appropriate solution. Table 1 shows that the quality of the prediction results is very deficient following these approaches.

TABLE 1. Mean Absolute Error (MAE) of the RS when not rated items are filled in with: rating 0, with rating 3, with the average of each user's ratings. Ratings from 1 to 5.

ru,i=0	ru,i=3	ru,i=avg(ru,i)
2.1144	0.9757	0.875

Memory-based CF RS face the sparsity problem by using customized similarity measures [55], [57]–[59]. These similarity measures provide distances between RS items (or users) better than the classic statistical metrics (Euclidean distance, cosine, Pearson correlation, etc.) [1], [60]. In this way, we will use RS similarity measures to select the initial centroids in the proposed pre-clustering process.

The pre-clustering method proposed in this paper consists of two phases that are executed consecutively. Phase 1 is the KMeansPlusLogPower algorithm [34]. This is a KMeans variant, which utilizes ''power'' users to find K centroids. Power users are the users with maximum number of ratings in the training set. Thus, the first centroid corresponds to the user who has cast the most votes (or a random one, in case of a tie). Next centroids are selected looking for the greater distance from the existing ones. Using KMeansPlusLogPower, the probability that a training set user becomes one of the K initial centroids is calculated by the following equation:

$$P_u \in C = \sum_{c \in C} dist(u, c) + log\left(\frac{|I_c|}{|I_u|} + 1\right) \qquad (16)$$

Where:

C is the set of existing centroids; $C \in [1..K]$

$|Iu|$ is the number of items rated by user u.

To implement the distance $dist(u, c)$ into the previous equation, we have tested different RS similarity measures, and compared the quality of the results with that obtained without performing pre-clustering. Results suggest an improvement in the quality when applying an adequate similarity measure; in this case: Mean Jaccard Difference (MJD) [57]. MJD improves results compared with the traditional Pearson correlation, used to implement KMeansPlus algorithms [34], MJD has also provided better results than Jaccard Mean Squared Differences (JMSD) [58].

Phase 1 outputs the cluster $k \in [1..K]$ to which each RS user belongs. In this way, for each RS user $u \in U$, we can know if, initially, it belongs to a cluster k. Phase 2 initializes the BNMF learning parameters, taking into account the

BNMF without pre-clustering

Experiment 1					Experiment 2					Experiment 3				
F1	F2	F3	F4	cluster	F1	F2	F3	F4	cluster	F1	F2	F3	F4	cluster
0.019	0.944	0.019	0.019	2	0.02	0.02	0.71	0.25	3	0.94	0.02	0.02	0.02	1
0.016	0.953	0.016	0.016	2	0.02	0.19	0.78	0.02	3	0.95	0.02	0.02	0.02	1
0.029	0.912	0.029	0.029	2	0.03	0.03	0.91	0.03	3	0.91	0.03	0.03	0.03	1
0.016	0.016	0.953	0.016	3	0.02	0.02	0.02	0.95	4	0.17	0.79	0.02	0.02	2
0.019	0.019	0.944	0.019	3	0.02	0.02	0.02	0.94	4	0.02	0.02	0.02	0.94	4
0.029	0.029	0.912	0.029	3	0.03	0.03	0.03	0.91	4	0.03	0.91	0.03	0.03	2
0.932	0.023	0.023	0.023	1	0.93	0.02	0.02	0.02	1	0.02	0.93	0.02	0.02	2
0.959	0.014	0.014	0.014	1	0.46	0.01	0.51	0.01	3	0.01	0.96	0.01	0.01	2
0.912	0.029	0.029	0.029	1	0.91	0.03	0.03	0.03	1	0.03	0.03	0.91	0.03	3
0.029	0.029	0.029	0.912	4	0.91	0.03	0.03	0.03	1	0.91	0.03	0.03	0.03	1
0.016	0.016	0.016	0.953	4	0.02	0.95	0.02	0.02	2	0.59	0.38	0.02	0.02	1
0.016	0.016	0.016	0.953	4	0.03	0.94	0.02	0.02	2	0.02	0.02	0.95	0.02	3
0.331	0.637	0.016	0.016	2	0.02	0.02	0.95	0.02	3	0.64	0.33	0.02	0.02	1

BNMF with pre-clustering (BNMF+)

MJD					JMSD					PEARSON				
F1	F2	F3	F4	cluster	F1	F2	F3	F4	cluster	F1	F2	F3	F4	cluster
0.019	0.944	0.019	0.019	2	0.02	0.02	0.02	0.94	4	0.93	0.02	0.02	0.02	1
0.016	0.953	0.016	0.016	2	0.02	0.02	0.02	0.95	4	0.53	0.02	0.02	0.43	1
0.029	0.912	0.029	0.029	2	0.03	0.03	0.03	0.91	4	0.87	0.04	0.04	0.04	1
0.016	0.016	0.953	0.016	3	0.19	0.02	0.77	0.02	3	0.02	0.02	0.02	0.93	4
0.019	0.019	0.944	0.019	3	0.79	0.02	0.17	0.02	1	0.02	0.93	0.02	0.03	2
0.029	0.029	0.912	0.029	3	0.03	0.03	0.91	0.03	3	0.04	0.87	0.04	0.04	2
0.932	0.023	0.023	0.023	1	0.02	0.02	0.93	0.02	3	0.91	0.03	0.03	0.03	1
0.959	0.014	0.014	0.014	1	0.01	0.01	0.96	0.01	3	0.91	0.02	0.02	0.06	1
0.912	0.029	0.029	0.029	1	0.03	0.03	0.03	0.91	4	0.04	0.04	0.87	0.04	3
0.029	0.029	0.029	0.912	4	0.03	0.91	0.03	0.03	2	0.04	0.04	0.87	0.04	3
0.016	0.016	0.016	0.953	4	0.02	0.95	0.02	0.02	2	0.02	0.02	0.93	0.02	3
0.016	0.016	0.016	0.953	4	0.02	0.95	0.02	0.02	2	0.02	0.02	0.93	0.02	3
0.331	0.637	0.016	0.016	2	0.02	0.02	0.31	0.66	4	0.93	0.02	0.02	0.02	1

FIGURE 7. **Pre-clustering and Bayesian Non Negative Matrix Factorization (BNMF+): Users factors.**

clustering results obtained in phase 1. Specifically, learning parameters $\epsilon^+_{i,k}$ and $\epsilon^-_{i,k}$ are initialized as follows (see the end of section II-A):

$$\epsilon^+_{i,k} = \beta + \sum_{u \in K r_{u,i} \neq \bullet} r^+_{u,i} \tag{17}$$

$$\epsilon^-_{i,k} = \beta + \sum_{u \in K r_{u,i} \neq \bullet} r^-_{u,i} \tag{18}$$

In summary, making use of a pre-clustering phase contributes to improve the quality of the clustering results. As we will see in next section, using a pre-clustering phase also improves both the quality of RS predictions and the clustering execution times. The contributions of this paper to BNMF pre-clustering are: a) choice of the best similarity measure for the *KMeansPlus* methods, and b) the way to link KMeans pre-clustering results with the initial values of the BNMF learning parameters ($\epsilon^+_{i,k}$ and $\epsilon^-_{i,k}$).

IV. EXPERIMENTS AND RESULTS

In this section we design a set of experiments that shows the validity of the paper hypothesis: the use of Bayesian non-negative matrix factorization improves clustering results in collaborative filtering RS environments. Moreover, it is verified that the prediction accuracy of the RS is also improved, reducing the average error of prediction (MAE).

Sub-section IV-A explains the experiments that have been carried out: methods that have been executed, chosen parameters, baselines, tested datasets, and quality measures to check the improvements of the proposed MF. Sub-section IV-B selects the BNMF parameters used to make experiments. Sub-section IV-C focuses on testing the quality of clustering, measuring the data cohesion. Sub-section IV-D tests the prediction accuracy, analyzing the mean error of predictions. Finally, sub-section IV-E shows the runtime improvements through the use of the proposed pre-clustering.

A. EXPERIMENTS DESIGN

Table 2 summarizes the most relevant decisions that have been made in the experiments design:

- We test: 1) The Bayesian non-Negative Matrix Factorization method (BNMF), 2) the proposed pre-clustering, and 3) The combination of both (BNMF +).

TABLE 2. Experiments designs scheme.

Experiments design			
Proposed methods	**Baselines**	**Datasets**	**Quality measures**
BNMF [54]	Matrix Factorization	Movielens Netflix*	Intra-cluster (cluster cohesion)
Pre-clustering [34]	(MF) [11], [12]	FilmTrust	Mean Absolute Error (MAE)
BNMF+ (BNMF + pre-clustering)	KMeansPlus LogPower [34]		Performance (execution time)

We use a customized Netflix dataset: Netflix, containing 100.000 users randomly selected from the original dataset. This size reduction has been necessary in order to perform experiments involving large number of clusters.

- We use as baselines: Improved K-Means and Matrix Factorization.
- We obtain results in RS open datasets.
- We measure the quality of predictions, data cohesion and runtime performance results.
- Experiments design includes 3-fold cross-validation techniques.
- Testing a broad range of cluster sizes: from 6 to 400.

Within-cluster (cohesion) quality measure has been chosen because it is the most representative in the MF context: The very nature of the MF leads to a classification of the RS elements (users and items) into a K number of hidden factors. Consequently, the between-cluster (separation) quality of the results is high. Where the RS model-based machine learning methods present the greatest difficulties of clustering is in the cohesion quality. This is because once the classification is done according to the K hidden factors values, the set of items or users into each cluster i depends on the $k-1$ hidden factors different to i. This effect could be reduced by making successive matrix factorizations clustering from the elements of each cluster, but the scalability of the method would be compromised. The advantage of our BNMF probabilistic approach lies in its ability to adopt different solutions based on the values of its parameters. In this way, the hidden factors will be configured differently in each solution, and we can choose the combination of parameter values that better clustering results achieve.

Mean Absolute Error (MAE) prediction accuracy measure has been chosen because it is the metric that indicates the quality of the CF predictions. RMSE provides conceptually similar results to MAE. Precision, recall and F1 measures test the quality of RS recommendations. There is some relationship between the quality of the prediction results and the quality of the recommendation results. When you are testing the quality of a CF method, both types of quality measures are used: an increase in the MAE usually leads to an increase in precision/recall, but not in a linear way. Knowing this level of detail is relevant in the RS recommendation papers; nevertheless, to validate the clustering quality usually it is measured the prediction quality (combined with the clustering quality).

B. CHOOSING THE ALPHA AND BETA VALUES

The BNMF method allows us to define α and β parameters (section II-A). The higher the β is, the more evidence the model requires to deduce that a group of users likes or dislikes an item. In this way, the higher the value of β, the higher the resulting clustering quality. On the other hand, a high β value generates conservative predictions and therefore worsens accuracy. An α value very close to zero means that users tend to belong to one group. Higher α values indicate that each user may probabilistically belongs, simultaneously, to more than one group. Therefore, an α value close to zero takes us to a hard clustering approach, providing better within-cluster quality results. An α value closer to one leads us to a more flexible soft clustering approach.

The BNMF α parameter is particularly promising for establishing the clustering properties. This parameter determines the sparsity level of the users hidden factors. The lower the value of α the more likely there will be to assign most of the user's features to a single hidden factor, while α high values provide more flexibility to distribute each user characteristics among the set of hidden factors. Lower values of α make it easier to assign each user to a single cluster, while using higher α values it is more natural to probabilistically assign each user to a set of clusters (soft-clustering).

Using MF, NMF or BNMF, each user is represented by a number K of hidden factors. Each hidden factor represents and encodes a mixture of characteristics; as a simplified example: factor 1 could represent action films, but not horror ones, while factor 2 could be associated with humor films. In this way, if $K = 2$, a user containing a 0.85 value into factor 1 and a 0.15 value into factor 2, has been represented as a fan of action movies, who dislikes horror films and does not like very much humor movies. Using BNMF, when α is small, characteristics of each user are concentrated into a single "determinant" factor or into a reduced number of factors. When α is large, the amount of necessary evidence to characterize a user into various factors is lower.

Fig. 8 shows the results of the following experiment: using the Movielens 1M dataset, the BNMF process is performed for different α values. This experiment sets to 6 the number of clusters ($K = 6$). For each α, by way of example, we take the mean of the hidden factor 5 for those users whose factor 5 is greater than the other factors (black bars into Fig. 8). As expected, for small values of *alpha* a single factor (factor 5) determines almost all the behaviour of the user, whereas with large α values, the main factor loses weight and the rest of factors also define the users characteristics. Into Figure 8, when α is similar to 1, factor 5 determines less than half of the user's characteristics.

Table 3 shows all the averaged values corresponding to the BNMF $K = 6$ hidden factors. Each table has been obtained using a different α value. Each column of each table shows the averaged values from factor 1 (f1) to factor 6 (f6). Each column factor is the one with the maximum value. For example, in the table corresponding to $\alpha = 0.8$, column f3 shows that factor 3 determines, on average, the 51%

TABLE 3. Users hidden factors distribution from Movielens 1M, BNMF, $K = 6$. Averaged results for all the users.

$\alpha \approx 0 \quad k = 6 \quad \beta = 5$

	f1	f2	f3	f4	f5	f6
f1	**0.81**	0.03	0.03	0.04	0.05	0.04
f2	0.05	**0.84**	0.02	0.03	0.03	0.02
f3	0.04	0.02	**0.85**	0.03	0.02	0.04
f4	0.06	0.03	0.02	**0.76**	0.09	0.04
f5	0.05	0.03	0.3	0.05	**0.81**	0.02
f6	0.05	0.02	0.02	0.05	0.01	**0.85**

$\alpha = 0.2 \quad k = 6 \quad \beta = 5$

f1	f2	f3	f4	f5	f6
0.77	0.04	0.06	0.05	0.03	0.05
0.06	**0.78**	0.04	0.04	0.03	0.04
0.08	0.05	**0.76**	0.03	0.06	0.02
0.05	0.03	0.03	**0.77**	0.06	0.06
0.04	0.04	0.06	0.10	**0.71**	0.04
0.05	0.03	0.03	0.07	0.04	**0.78**

$\alpha = 0.4 \quad k = 6 \quad \beta = 5$

	f1	f2	f3	f4	f5	f6
f1	**0.73**	0.06	0.06	0.04	0.07	0.06
f2	0.07	**0.72**	0.04	0.03	0,06	0.08
f3	0.05	0.04	**0.70**	0.06	0.04	0.10
f4	0.04	0.04	0.10	**0.59**	0.13	0.11
f5	0.06	0.05	0.03	0.06	**0.70**	0.09
f6	0.04	0.07	0.07	0.05	0.07	**0.70**

$\alpha = 0.6 \quad k = 6 \quad \beta = 5$

f1	f2	f3	f4	f5	f6
0.58	0.12	0.07	0.10	0.05	0.09
0.08	**0.63**	0.08	0.10	0.06	0,05
0.09	0.13	**0.58**	0.06	0.06	0.08
0.07	0.10	0.04	**0.64**	0.08	0.07
0.06	0.09	0.05	0.12	**0.59**	0.10
0.09	0.07	0.07	0.10	0.09	**0.57**

$\alpha = 0.8 \quad k = 6 \quad \beta = 5$

	f1	f2	f3	f4	f5	f6
f1	**0.59**	0.09	0.10	0.10	0.06	0.06
f2	0.10	**0.57**	0.10	0.08	0.09	0.06
f3	0.13	0.12	**0.51**	0.09	0.05	0.09
f4	0.11	0.10	0.10	**0.52**	0.07	0.10
f5	0.07	0.14	0.07	0.08	**0.53**	0.10
f6	0.07	0.10	0.11	0.13	0.10	**0.50**

$\alpha \approx 1 \quad k = 6 \quad \beta = 5$

f1	f2	f3	f4	f5	f6
0.48	0.11	0.09	0.10	0.13	0.10
0.11	**0.52**	0.06	0.06	0.09	0.15
0.10	0.07	**0.49**	0.14	0.10	0.10
0.08	0.06	0.10	**0.52**	0.12	0.13
0.12	0.09	0.08	0.14	**0.47**	0.10
0.08	0.11	0.06	0.12	0.09	**0.53**

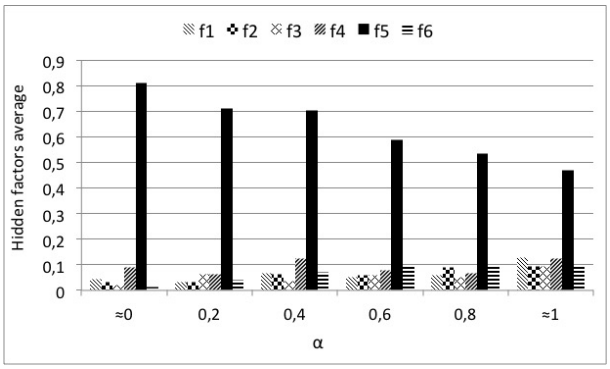

FIGURE 8. Users hidden factors distribution from Movielens 1M, BNMF, K = 6. Averaged results for users whose maximum hidden factor is f5.

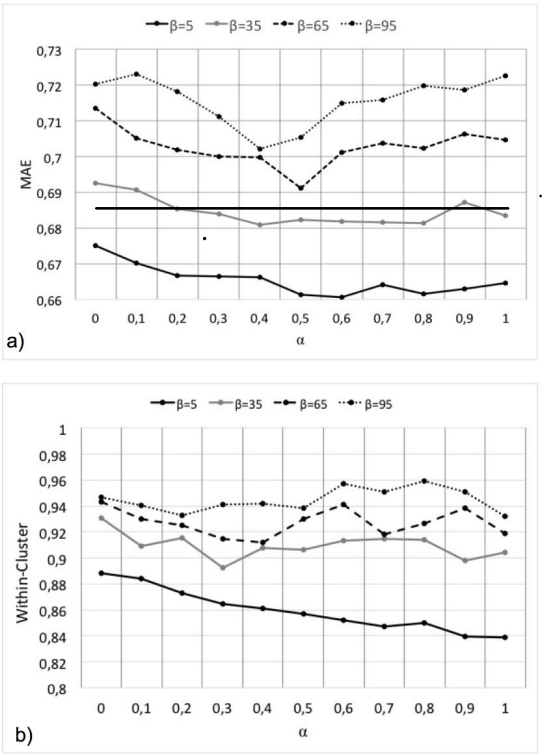

FIGURE 9. MAE and within-cluster quality results obtained when different values of the α and β parameters are combined. Dataset: Movielens 1M, Method: BNMF, Number of clusters (K): 10. a) Lower values are the better ones, b) Higher values are the better ones.

of the characteristics from the users most characterized by factor 3, whereas those same users are determined 11% by factor 6.

Once the significance of the alpha parameter has been detailed, figure 9 shows the MAE and the within-cluster quality results when different BNMF α and β values are applied to the Movielens 1M dataset. These results were obtained using 10 clusters ($K = 10$). Fig. 9a shows the mean absolute error impact when different values are applied to the α and β parameters. As expected, accuracy improves as the β value decreases. The horizontal line into Fig. 9a indicates the MAE value provided by the classical MF. Our objective is to test the BNMF method superiority to perform RS clustering, without worsening the predictions accuracies. In this way we select the value $\beta = 5$ which gives us a margin of improvement both in accuracy and clustering quality.

Fig. 9b shows the within-cluster impact when different values are applied to the α and β parameters. Within-cluster values (y axis) have been normalized to the interval [0..1].

As predicted, a low beta value ($\beta = 5$) produces worse clustering results than high beta values do. When the clustering process is independent of the recommendation process it is more appropriate to choose a high β value. In our case, we look for a RS balance: we want to improve the clustering quality without worsening the prediction quality, so we maintain our choice of $\beta = 5$.

Once we have chosen the value $\beta = 5$, from Fig. 9b we can see that α values close to zero give us better clustering qualities without compromising the accuracy quality (Fig. 9a). In this way we select the $\alpha \approx 0$ value and therefore we choose the hard clustering approach. It is important to indicate that the BNMF method allows values $\alpha \in (0, 1)$, so $\alpha \approx 0$ means an α value very close to zero.

C. CLUSTERING IMPROVEMENTS

In this section we present the quality results obtained from each clustering experiment, according to the design presented in Table 2. Among the existing clustering quality measures [61], [62] we have taken the most representative and popular: cohesion (intra-cluster distance). Usually, this measure is defined as the sum of the distances between each cluster element and its centroid. The following equation formalizes the concept:

$$cohesion = \sum_{k=1}^{K} \sum_{\forall u \in c_k} similarity(u, c_k) \tag{19}$$

where: K is the number of clusters, C_k is the k cluster, c_k is the k cluster centroid, u is a RS user, similarity is the chosen similarity measure (Pearson, MJD, Euclidean, etc.). The higher the cohesion value, the better the clustering quality. Since we use RS datasets, we have chosen Pearson correlation as similarity measure [1]: It is a classical CF similarity measure and it returns $[-1..1]$ bounded values. Additionally, in order to be able to compare results through different datasets, we have normalized cohesion to the $[-1..1]$ range dividing its value among the number of each RS users (U):

$$cohesion = \frac{1}{U} \sum_{k=1}^{K} \sum_{\forall u \in c_k} correlation(u, c_k) \tag{20}$$

Following Table 2 experiments design, we have tested the proposed methods along with the chosen baselines. Fig. 10 shows the Movielens, Netflix* and FilmTrust datasets results. As explained into the previous subsection, the chosen parameters values are: $\beta = 5$, $\alpha \approx 0$. As it can be seen into Fig. 10, BNMF and BNMF+ within-cluster results are similar.

Analyzing the three graphs into Fig. 10, the main conclusions that we can extract are:

- The proposed BNMF method greatly improves the clustering quality of the two baselines (KMeans+ and MF). Fig. 11 shows the clustering improvement percentages of BNMF over the baselines average.
- BNMF and BNMF+ provide the same clustering quality, so their curves are almost completely overlapping: we expect execution time improvements rather than quality ones.
- KMeans+ gets better results than MF when applied to RS datasets, confirming the effectiveness of the Plus-Log-Power initialization [34].
- As expected, the higher the number of clusters (K) the better the quality of clustering: there are a greater variety of centroids to assign each element.
- Clustering quality decreases when the size of the dataset increases; FilmTrust shows the best results, followed by MovieLens, and finally Netflix*: the larger the dataset the more elements, on average, must be assigned to each cluster.
- BNMF clustering improvement over baselines (MF and KMeans+) is higher when the number of clusters (K) is lower (Fig. 11): there is a bigger margin of improvement on lower K values (figure 10).

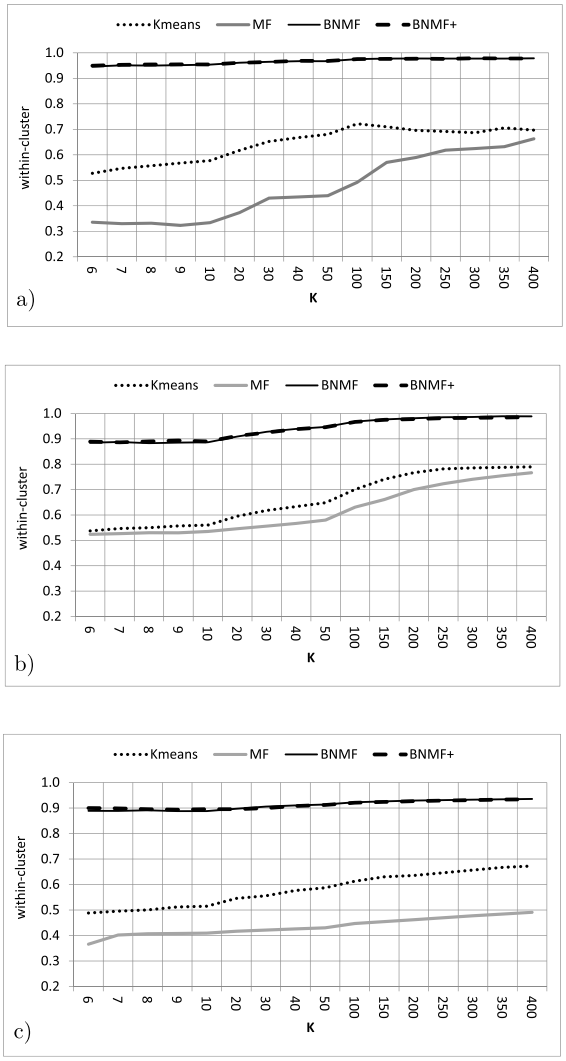

FIGURE 10. Clustering results using a) FilmTrust, b) Movielens, and c) Netflix*. y-axis: normalized cohesion (within-cluster) results; x-axis: Number of clusters (K). $\beta = 5$, $\alpha \approx 0$. Higher values are the better ones.

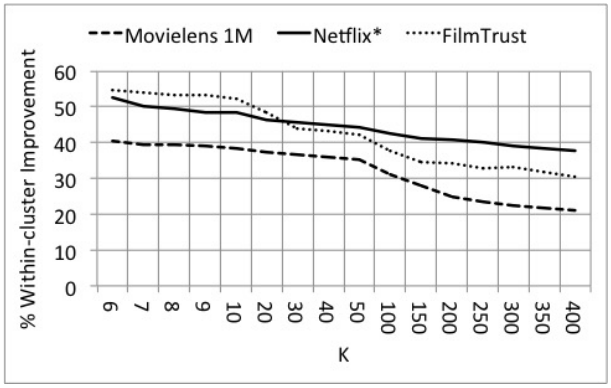

FIGURE 11. Clustering improvements of BNMF compared to the mean of the baselines (MF and KMeans+). y-axis: % improvement; x-axis: number of clusters (K). $\beta = 5$, $\alpha \approx 0$.

- From a significant number of clusters ($K > 20$), the improvement of BNMF with respect to the baselines is greater in the larger datasets (Fig. 11): there is a bigger margin of improvement in large datasets (Fig. 10).

D. PREDICTION ACCURACY IMPROVEMENTS

In this section we determine the ability of each tested method (KMeans+, MF, BNMF, BNMF+) to generate correct predictions. Fig. 12 shows the predicted quality (MAE) results for each tested dataset.

The main conclusions from graphs into Fig. 12 are:

- The proposed method (BNMF) provides better accuracy results than the baselines do.
- The proposed pre-clustering use achieves an additional accuracy improvement.
- When the classical MF method is used to cluster, predictions quality is significantly affected when selecting high K values.

Comparing graphs into Fig. 10 and Fig. 12 we can determine an inverse relationship between clustering quality and prediction quality. Fig. 13 shows, in the range [0..1], the evolution of the BNMF+ accuracy (expressed as 1-MAE) and the evolution of the BNMF+ within-cluster; Datasets: Netflix* (NF) and Movielens (ML). The points where the accuracy and within-cluster curves intersect gives us an optimal $K = 35$ for Netflix and $K = 20$ for Movielens. These values are consistent with the usual need to increase the number of clusters as the size of the datasets increase.

E. PRE-CLUSTERING EXECUTION IMPROVEMENT

In the previous sub-sections we have verified how the use of pre-clustering implies a little improvement in the quality of the BNMF predictions (MAE). In this sub-section we focus on the execution time improvement that is achieved when using the proposed BNMF pre-clustering technique, as well as on the BNMF execution time superiority regarding the selected baselines.

We have executed both the proposed methods (BNMF and BNMF+) and the baseline ones (KMeans+ and MF).

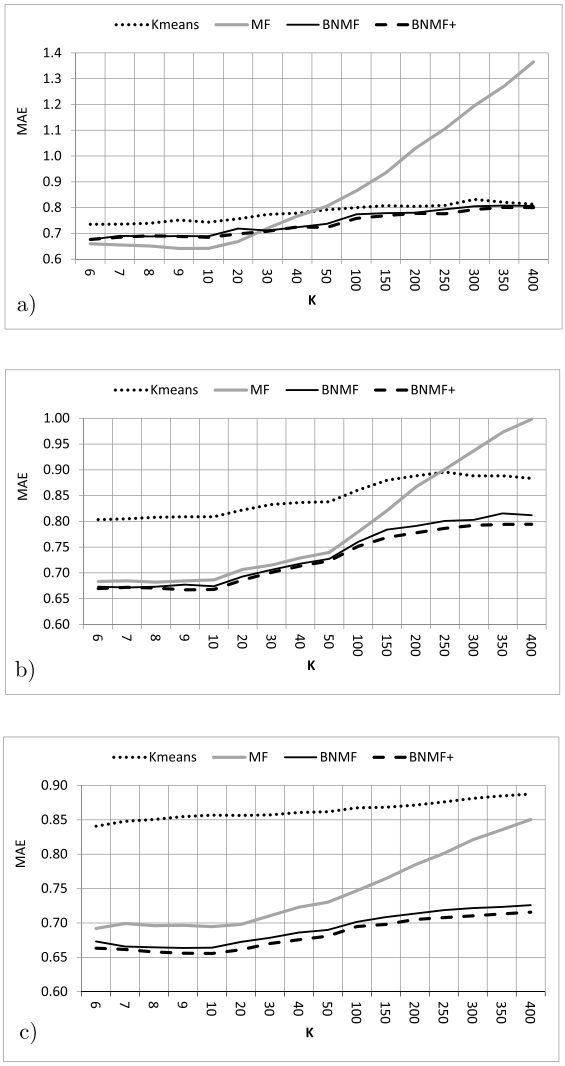

FIGURE 12. Mean Absolute Error results using a) FilmTrust, b) Movielens, and c) Netflix*. y-axis: MAE results; x-axis: number of clusters (K). $\beta = 5$, $\alpha \approx 0$. Lower values are the better ones.

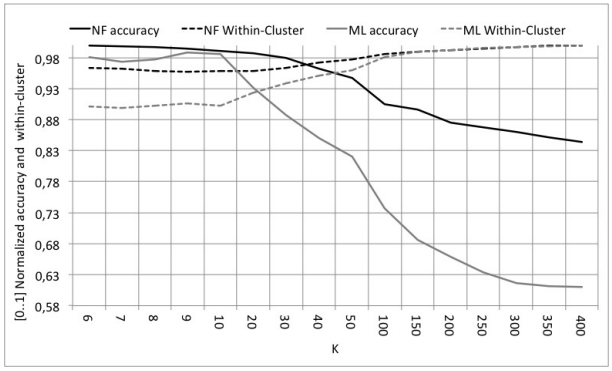

FIGURE 13. Accuracy (1-MAE) and within-cluster trends using BNMF+ on Movielens (ML) and Netflix* (NF). y-axis: (1-MAE) and within-cluster results normalized to the range [0..1]; x-axis: number of clusters (K) values. $\beta = 5$, $\alpha \approx 0$.

We show results choosing different values of K (from $K = 6$ to $K = 400$), $\beta = 5$, $\alpha \approx 0$. Fig. 14 shows the results; the most relevant conclusions are: a) BNMF execution time after

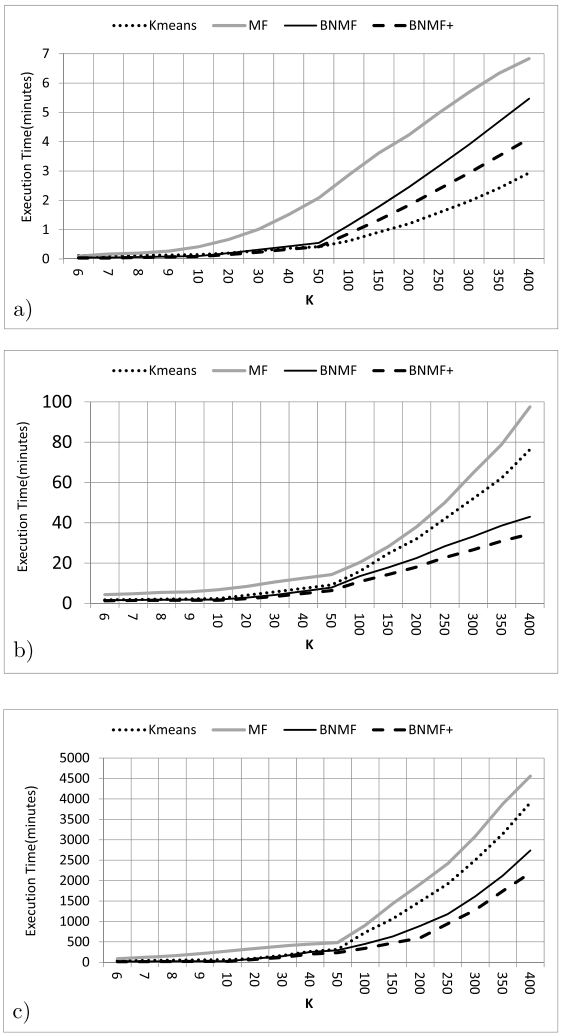

FIGURE 14. Execution performance on both the proposed methods and the baseline ones. Datasets: a) FilmTrust, b) Movielens, c) Netflix*; y-axis: time (minutes); x-axis: number of clusters (K). $\beta = 5$, $\alpha \approx 0$.

pre-clustering (BNMF+) is better than BNMF execution time without pre-clustering, b) Baseline methods are slower than BNMF ones, c) The performance differences are significant when the number of clusters is not small (bigger than 50), d) When using small datasets (e.g. FilmTrust), MF based methods are slower than KMeans+, and e) The MF baseline is slower than KMeans+ and BNMF, which emphasizes the suitability of BNMF (and BNMF +) as RS model-based method.

V. CONCLUSION

Beyond accuracy, recommender systems clustering is important; it allows to face several collaborative filtering challenges: recommendations explanation, data analytics, visualization and browsing through the dataset information, obtaining the characteristics that define each group of users or items, etc.

Model-based methods get accurate results and they accelerate the prediction process once the model has been trained. Among model-based methods, matrix factorization techniques are the most popular ones. Classical matrix factorization methods do not provide flexibility in their operation beyond the choice of the number of hidden factors. The Bayesian non-negative matrix factorization method gives flexibility factorizing data. It also provides the useful condition that each individual user hidden factor is a probability. Experiments show that the BNMF method: a) It improves, simultaneously, the baselines accuracy and their clustering quality results, b) It can be configured to increase clustering improvements reducing accuracy, or vice versa, 3) It obtains important execution times improvements, compared to baselines, especially when the number of clusters is not small.

This paper proposes an original BNMF pre-clustering algorithm (BNMF+) that provides clustering and performance improvements. Additionally, this work opens a hopeful future work: getting clustering improvements through a BNMF soft-clustering approach. BNMF can be parameterized so that each user or item probabilistically belongs to several clusters. Soft BNMF also opens up the possibility of improving recommender systems accuracy.

REFERENCES

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013.
[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[3] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
[4] E. Arabnejad, R. F. Moghaddam, and M. Cheriet, "PSI: Patch-based script identification using non-negative matrix factorization," *Pattern Recognit.*, vol. 67, pp. 328–339, Jul. 2017.
[5] R. Rad and M. Jamzad, "Image annotation using multi-view non-negative matrix factorization with different number of basis vectors," *J. Vis. Commun. Image Represent.*, vol. 46, pp. 1–12, Jul. 2017.
[6] K. Devarajan, "Nonnegative matrix factorization: An analytical and interpretive tool in computational biology," *PLoS Comput. Biol.*, vol. 4, no. 7, p. e1000029, 2008.
[7] X. Wang, X. Cao, D. Jin, Y. Cao, and D. He, "The (un) supervised NMF methods for discovering overlapping communities as well as hubs and outliers in networks," *Phys. A, Stat. Mech. Appl.*, vol. 446, pp. 22–34, Mar. 2016.
[8] S. Mirzaei, H. V. Hamme, and Y. Norouzi, "Under-determined reverberant audio source separation using Bayesian non-negative matrix factorization," *Speech Commun.*, vol. 81, pp. 129–137, Jul. 2016.
[9] F. Zhang, Y. Lu, J. Chen, S. Liu, and Z. Ling, "Robust collaborative filtering based on non-negative matrix factorization and R_1-norm," *Knowl.-Based Syst.*, vol. 118, pp. 177–190, Feb. 2017.
[10] D. Bokde, S. Girase, and D. Mukhopadhyay, "Matrix factorization model in collaborative filtering algorithms: A survey," *Procedia Comput. Sci.*, vol. 49, pp. 136–146, Jan. 2015.
[11] Y. Li, D. Wang, H. He, L. Jiao, and Y. Xue, "Mining intrinsic information by matrix factorization-based approaches for collaborative filtering in recommender systems," *Neurocomputing*, vol. 249, pp. 48–63, Aug. 2017.
[12] F. Ortega, A. Hernando, J. Bobadilla, and J. H. Kang, "Recommending items to group of users using matrix factorization based collaborative filtering," *Inf. Sci.*, vol. 345, pp. 313–324, Jun. 2016.
[13] V. Kumar, A. K. Pujari, S. K. Sahu, V. R. Kagita, and V. Padmanabhan, "Proximal maximum margin matrix factorization for collaborative filtering," *Pattern Recognit. Lett.*, vol. 86, pp. 62–67, Jan. 2017.
[14] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proc. 5th ACM Conf. Recommender Syst. (RecSys)*, New York, NY, USA, 2011, pp. 301–304.
[15] A. Friedman, S. Berkovsky, and M. A. Kaafar, "A differential privacy framework for matrix factorization recommender systems," *User Model. User-Adapted Interact.*, vol. 26, pp. 425–458, Dec. 2016.

[16] Y. Zhang, M. Chen, D. Huang, D. Wu, and Y. Li, "iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization," *Future Generat. Comput. Syst.*, vol. 66, pp. 30–35, Jan. 2017.

[17] G. Zhang, M. He, H. Wu, G. Cai, and J. Ge, "Non-negative multiple matrix factorization with social similarity for recommender systems," in *Proc. IEEE/ACM 3rd Int. Conf. Big Data Comput. Appl. Technol. (BDCAT)*, Dec. 2016, pp. 280–286.

[18] K. Su, L. Ma, B. Xiao, and H. Zhang, "Web service QoS prediction by neighbor information combined non-negative matrix factorization," *J. Intell. Fuzzy Syst.*, vol. 30, no. 6, pp. 3593–3604, 2016.

[19] F. Zafari and I. Moser, "Modelling socially-influenced conditional preferences over feature values in recommender systems based on factorised collaborative filtering," *Expert Syst. Appl.*, vol. 87, pp. 98–117, Jun. 2017.

[20] N. Hayashi and S. Watanabe. (Dec. 2016). "Upper bound of Bayesian generalization error in non-negative matrix factorization." [Online]. Available: https://arxiv.org/abs/1612.04112

[21] R. Schachtner, G. Pöppel, A. Tomé, C. G. Puntonet, and E. W. Lang, "A new Bayesian approach to nonnegative matrix factorization: Uniqueness and model order selection," *Neurocomputing*, vol. 138, pp. 142–156, Aug. 2014.

[22] Q. Sun, J. Lu, Y. Wu, H. Qiao, X. Huang, and F. Hu, "Non-informative hierarchical Bayesian inference for non-negative matrix factorization," *Signal Process.*, vol. 108, pp. 309–321, Mar. 2015.

[23] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2007, pp. 1–8.

[24] C. C. Aggarwal and C. K. Reddy, *Data Clustering: Algorithms and Applications*, 1st ed. London, U.K.: Chapman & Hall, 2013.

[25] C. Birtolo and D. Ronca, "Advances in clustering collaborative filtering by means of fuzzy C-means and trust," *Expert Syst. Appl.*, vol. 40, pp. 6997–7009, Dec. 2013.

[26] C.-X. Zhang, Z.-K. Zhang, L. Yu, C. Liu, H. Liu, and X.-Y. Yan, "Information filtering via collaborative user clustering modeling," *Phys. A, Stat. Mech. Appl.*, vol. 396, pp. 195–203, Feb. 2014.

[27] X. Wang and J. Zhang, "Using incremental clustering technique in collaborative filtering data update," in *Proc. IEEE 15th Int. Conf. Inf. Reuse Integr. (IEEE IRI)*, Aug. 2014, pp. 420–427.

[28] H. Wu, Y.-J. Wang, Z. Wang, X. L. Wang, and S. Z. Du, "Two-phase collaborative filtering algorithm based on co-clustering," *J. Softw.*, vol. 21, no. 5, pp. 1042–1054, 2010.

[29] C. Rana and S. K. Jain, "An evolutionary clustering algorithm based on temporal features for dynamic recommender systems," *Swarm Evol. Comput.*, vol. 14, pp. 21–30, Feb. 2014.

[30] A. Salah, N. Rogovschi, and M. Nadif, "A dynamic collaborative filtering system via a weighted clustering approach," *Neurocomputing*, vol. 175, pp. 206–215, Jan. 2016.

[31] C.-L. Liao and S.-J. Lee, "A clustering based approach to improving the efficiency of collaborative filtering recommendation," *Electron. Commerce Res. Appl.*, vol. 18, pp. 1–9, Aug. 2016.

[32] M. K. Najafabadi, M. N. Mahrin, S. Chuprat, and H. M. Sarkan, "Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data," *Comput. Hum. Behav.*, vol. 67, pp. 113–128, Feb. 2017.

[33] J. Chen, Uliji, H. Wang, and Z. Yan, "Evolutionary heterogeneous clustering for rating prediction based on user collaborative filtering," *Swarm Evol. Comput.*, vol. 38, pp. 35–41, Feb. 2017.

[34] S. Zahra, M. A. Ghazanfar, A. Khalid, M. A. Azam, U. Naeem, and A. Prugel-Bennett, "Novel centroid selection approaches for Kmeans-clustering based recommender systems," *Inf. Sci.*, vol. 320, pp. 156–189, Nov. 2015.

[35] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[36] J. Kim and H. Park, "Sparse nonnegative matrix factorization for clustering," School Comput. Sci. Eng., Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep., 2008. [Online]. Available: http://hdl.handle.net/1853/20058

[37] X. Zhang, L. Zong, X. Liu, and J. Luo, "Constrained clustering with nonnegative matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 7, pp. 1514–1526, Jul. 2016.

[38] N. D. Buono and G. Pio, "Non-negative matrix tri-factorization for co-clustering: An analysis of the block matrix," *Inf. Sci.*, vol. 301, pp. 13–26, Apr. 2015.

[39] J. Yoo and S. Choi, "Weighted nonnegative matrix co-tri-factorization for collaborative prediction," in *Advances in Machine Learning*. Berlin, Germany: Springer, 2009, pp. 396–411.

[40] C. Buchta, M. Kober, I. Feinerer, and K. Hornik, "Spherical k-means clustering," *J. Stat. Softw.*, vol. 50, no. 10, pp. 1–22, 2012.

[41] T. George and S. Merugu, "A scalable collaborative filtering framework based on co-clustering," in *Proc. 5th IEEE Int. Conf. Data Mining (ICDM)*, Washington, DC, USA, Nov. 2005, pp. 625–628.

[42] J. Yang, W. Wang, H. Wang, and P. Yu, "δ-Clusters: Capturing subspace correlation in a large data set," in *Proc. 18th Int. Conf. Data Eng.*, Feb./Mar. 2002, pp. 517–528.

[43] T. Li, Y. Zhang, and V. Sindhwani, "A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge," in *Proc. Joint Conf. 47th Annu. Meeting ACL 4th Int. Joint Conf. Natural Lang. Process. (AFNLP)*, vol. 1. Stroudsburg, PA, USA, 2009, pp. 244–252.

[44] Q. Gu and J. Zhou, "Local learning regularized nonnegative matrix factorization," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 1046–1051.

[45] S. Frémal and F. Lecron, "Weighting strategies for a recommender system using item clustering based on genres," *Expert Syst. Appl.*, vol. 77, pp. 105–113, Jul. 2017.

[46] R. Hu, W. Dou, and J. Liu, "ClubCF: A clustering-based collaborative filtering approach for big data application," *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 3, pp. 302–313, Sep. 2014.

[47] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 171–209, Apr. 2014.

[48] R. Du, D. Kuang, B. Drake, and H. Park, "DC-NMF: Nonnegative matrix factorization based on divide-and-conquer for fast clustering and topic modeling," *J. Global Optim.*, vol. 68, no. 4, pp. 1–22, 2017.

[49] C. Zhang, H. Wang, S. Yang, and Y. Gao, "Incremental nonnegative matrix factorization based on matrix sketching and k-means clustering," in *Proc. Int. Conf. Intell. Data Eng. Autom. Learn.*, 2016, pp. 426–435.

[50] R. Kannan, G. Ballard, and H. Park, "A high-performance parallel algorithm for nonnegative matrix factorization," in *Proc. 21st ACM SIGPLAN Symp. Principles Pract. Parallel Program. (PPoPP)*, New York, NY, USA, 2016, pp. 9-1–9-11.

[51] B. Lakshminarayanan, G. Bouchard, and C. Archambeau, "Robust Bayesian matrix factorisation," in *Proc. 14th Int. Conf. Artif. Intell. Stat.*, 2011, pp. 425–433.

[52] M. Luo, F. Nie, X. Chang, Y. Yang, A. G. Hauptmann, and Q. Zheng, "Probabilistic non-negative matrix factorization and its robust extensions for topic modeling," in *Proc. AAAI*, 2017, pp. 2308–2314.

[53] T. Virtanen, A. T. Cemgil, and S. Godsill, "Bayesian extensions to non-negative matrix factorisation for audio signal modelling," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2008, pp. 1825–1828.

[54] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model," *Knowl.-Based Syst.*, vol. 97, pp. 188–202, Apr. 2016.

[55] J. Bobadilla and F. Serradilla, "The effect of sparsity on collaborative filtering metrics," in *Proc. 20th Austral. Conf. Austral. Database*, vol. 92. 2009, pp. 9–18.

[56] B. Sun and L. Dong, "Dynamic model adaptive to user interest drift based on cluster and nearest neighbors," *IEEE Access*, vol. 5, pp. 1682–1691, 2017.

[57] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowl.-Based Syst.*, vol. 26, pp. 225–238, Feb. 2012.

[58] J. Bobadilla, F. Serradilla, and J. Bernal, "A new collaborative filtering metric that improves the behavior of recommender systems," *Knowl.-Based Syst.*, vol. 23, no. 6, pp. 520–528, 2010.

[59] Suryakant and T. Mahara, "A new similarity measure based on mean measure of divergence for collaborative filtering in sparse environment," *Procedia Comput. Sci.*, vol. 89, pp. 450–456, Dec. 2016.

[60] J. Sanchez, F. Serradilla, E. Martinez, and J. Bobadilla, "Choice of metrics used in collaborative filtering and their impact on recommender systems," in *Proc. 2nd IEEE Int. Conf. Digit. Ecosyst. Technol. (DEST)*, Feb. 2008, pp. 432–436.

[61] U. F. Siddiqi and S. M. Sait, "A new heuristic for the data clustering problem," *IEEE Access*, vol. 5, pp. 6801–6812, 2017.

[62] Q. Zhao, M. Xu, and P. Fränti, "Sum-of-squares based cluster validity index and significance analysis," in *Adaptive and Natural Computing Algorithms* (Lecture Notes in Computer Science), vol. 5495. Springer, 2009, pp. 313–322, doi: 10.1007/978-3-642-04921-7_32.

JESÚS BOBADILLA received the B.S. degree in computer science from the Universidad Politécnica de Madrid and the Ph.D. degree in computer science from the Charles III University of Madrid. He is currently a Lecturer with the Department of Applied Intelligent Systems, Universidad Politécnica de Madrid. He is currently a Lecturer with the Department of Intelligent Systems, Universidad Politécnica de Madrid. He has been a Visitor Researcher with the International Computer Science Institute, Berkeley University. He is also in charge of the FilmAffinity.com research team, which is involved in the collaborative filtering kernel of the website. He is a habitual author of programming languages books with McGraw-Hill, Ra-Ma, and Alfa Omega publishers. His research interests include information retrieval, recommender systems, and speech processing.

RODOLFO BOJORQUE was born in 1982. He received the B.S degree in systems engineering from Universidad Politécnica Salesiana, Ecuador, in 2012, the master's degree in security information and communications technologies from the Universidad Autónoma de Barcelona, Spain, in 2015, and the master's degree in computer science and technology from the Universidad Politécnica de Madrid, Spain, in 2017.

He is currently a Lecturer with Universidad Politécnica Salesiana, Ecuador. His research interests include machine learning techniques and intelligent systems. He is also a member of the Research Team in Artificial Intelligence and Assistive Technology, and his team is collaborating with the Universidad Politécnica de Madrid.

ANTONIO HERNANDO ESTEBAN was born in 1976. He received the B.A. degree in computer science in 1999, the B.A. degree in mathematics in 2005, and the Ph.D. degree in artificial intelligence in 2007. He has been an Assistant Lecturer with the Complutense University in Madrid, and he is currently a Lecturer with the Technical School, Universidad Politécnica de Madrid. His main research lines relate to artificial intelligence and computer algebra.

REMIGIO HURTADO was born in 1989. He received the B.S degree in systems engineering from Universidad Politécnica Salesiana, Ecuador, in 2012, the master's degree in information and software technology from the Instituto Tecnológico y de Estudios Superiores de Monterrey, México, in 2014, and the master's degree in computer science and technology from the Universidad Politécnica de Madrid, Spain, in 2017.

He is currently a Lecturer with Universidad Politécnica Salesiana, Ecuador. His research interests include recommender systems and natural language processing. He is also a member of the Research Team in Artificial Intelligence and Assistive Technology, and his team is collaborating with the Universidad Politécnica de Madrid.