Item Cold-Start Recommendations: A Non-negative Matrix Factorization Approach

Anonymized

ABSTRACT

Recommender systems suggest to users items that they might like (*e.g.*, news articles, songs, movies) and, in doing so, they help users deal with information overload and enjoy a personalized experience. One of the main problems of these systems is the item cold-start, *i.e.*, when a new item is introduced in the system and no past information is available, then no effective recommendations can be produced. The item cold-start is a very common problem in practice: modern online platforms have hundreds of new items published every day. To solve the problem, we introduce a new hybrid recommender system based on a *Joint Non-negative Matrix Factorization*. This results in a unified matrix factorization framework that exploits both the properties of the items and past user preferences by jointly decomposing the content and the collaborative matrices. Additionally, we propose a graph-based regularization which accounts for the local geometric structure of the data. We present learning algorithms based on multiplicative update rules, that are efficient and easy to implement. Experiments on two item cold-start use cases: news recommendation and email recipient recommendation, demonstrate the effectiveness of the two approaches and show that they significantly outperform four state-of-the-art methods for item cold-start recommendations.

1. INTRODUCTION

Recommender systems are aimed to help users of online platforms to deal with the large volumes of information and to provide them a personalized experience. This is achieved by suggesting items of interest to the users based on their explicit and implicit preferences. Recommender systems use a number of different technologies, but may be broadly classified into two groups: content-based and collaborative filtering systems. Content-based systems examine the properties of the items and recommend items which are similar to the ones the user preferred in the past. They model the taste of a user by building a user profile based on the properties of the items the user liked, and use the profile to compute the similarity with new items. Items which are most similar to the user's profile are recommended. Collaborative filtering systems, on the other hand, ignore the properties of the items and base their recommendations on community preferences. They recommend items that users with similar tastes and preferences liked in the past. Two users are considered similar if they have many items in common.

One of the main problems for recommender systems is the cold-start problem, *i.e.*, when a new item or user is introduced in the system. In this study we focus on the problem of producing effective recommendations for new items: the item cold-start. Collaborative filtering systems suffer from this problem as they rely on the previous ratings of the users. Content based approaches, on the other hand, may still produce recommendations using the description of the items and are the default solution to the item cold-start. However, they tend to achieve lower accuracy and, in practice, they are seldom the only choice.

The problem of item cold-start is of great practical importance because of two main reasons. First, modern online platforms have hundreds of new items everyday. Hence, effectively recommending them is essential for keeping the users continuously engaged. Second, collaborative filtering methods are at the core of most recommendation engines, as they tend to achieve the state-of-the-art accuracy [16]. However, to be effective and produce recommendations at the expected accuracy, they require that items are rated by a sufficient number of users. Therefore, it is crucial for every collaborative recommender to reach this state as soon as possible. Having methods that produce accurate recommendations for new items will allow enough feedback to be collected in a short amount of time, making effective collaborative recommendations possible.

Dispute it's practical importance the item cold-start problem has not been widely studied in the literature. The majority of the techniques focus on using the user profiles to make a link between the users and the item descriptions (*e.g.*, [30, 32]). However, they fail to make explicit use of the collaborative filtering information. A series of more sophisticated approaches addressing this limitation have been proposed in the link prediction literature [25], however this line of work does not target the item cold-start scenario. In this study, we focus on the item cold-start problem while fully exploiting the collaborative matrix as well as the items content by relying on a joint matrix factorization technique.

Recently, matrix factorization techniques have been extensively used in recommendation systems and topic mod-

WSDM '14 New York City

elling literature. Many collaborative filtering systems approximate the collaborative matrix by applying techniques such as Singular Value Decomposition (SVD) or UV decomposition [21]. Similar matrix factorization techniques have been used to discover topics occurring in a document collections by decomposing the content. *i.e.*, document-term matrix. Non-negative matrix factorization (NMF) is one such approach that aims at factorizing the document-term matrix in two non-negative, low-rank matrices, where one matrix corresponds to the topics that occur in the collection, while the other represents the extent to which each document belongs to these topics. Due to the non-negativity constraints, NMF produces a so-called "additive parts-based" representation of the data [11] that increases the sparsity and the interpretability of the hidden factors.

In this paper, we propose a new hybrid approach that exploits both the properties of the items and the similarity of the user preferences. We introduce *Joint* NMF (JNMF), a matrix factorization technique based on NMF, that jointly decomposes the content and the collaborative matrices in a common low-dimensional space. Given the description of a new item (*e.g.*, the content of a news article), we may project it in the common low-dimensional space and infer the users which are most likely to be interested in it. By doing so, we are able to overcome the item cold-start problem. We also propose an extension of the model, *Joint* NMF *with Graph Regularization* (JNMF-GR), that exploits the local geometric structure of the data to discover better low-dimensional space. Finally, we perform an extensive experimental evaluation of the models on two item cold-start use cases: email recipient recommendation (based on explicit feedback) and news recommendation (based on implicit feedback). We show that the proposed models outperform four state-of-the-art baseline approaches.

Our contributions in this paper can be summarized as:

- We introduce a new method for recommendation, *Joint* NMF, that combines the content and collaborative information in a unified matrix factorization framework. We propose a simple and efficient learning algorithm, based on multiplicative update rules and prove its convergence.
- We develop an extension of the proposed method, *Joint* NMF with *Graph Regularization*, that accounts for the local geometric structure of the data, and we derive an effecient learning algorithm, with proof of its convergence.
- We conduct an extensive experimental study and we show that the proposed methods outperform four state-of-the-art methods for item-cold start recommendation.

The paper is organized as follows. In Section 2 we discuss the related work and position our paper with respect to it. In Section 3 we define the problem we consider and in Section 4 we give a brief review of NMF. Sections 5 and 6 present in detail the proposed models: JNMF and JNMF-GR. Our experimental evaluation is described in Section 7, followed by conclusions and future work in Section 8.

2. RELATED WORK

In this section, we briefly describe several hybrid recommender systems that are capable of addressing the item cold-start scenario, as well as matrix factorization techniques that are based on the idea of joint factorization.

Soboroff [32] proposed a technique based on Latent Semantic Indexing (LSI) for combining the collaborative filtering input and the document content for recommendation of textual items. The method builds a content profile for each user as a linear combination of the preferred documents. LSI is then applied to the user profiles to discover topics in the collection and implicitly learn commonalities among the user profiles. Incoming documents are projected into the LSI space and compared to user profiles. The documents are recommended to the users who have the most similar profiles. The author argues that applying LSI on the user profiles instead of the documents allows one to take into account the collaborative input and consequently improves the recommendation performance. However, the system is not evaluated in the cold-start scenario. In section 7, we compare this technique against the methods proposed.

Schein *et al.* [30] propose a probabilistic model for cold-start recommendations that is very similar to the one proposed by Soboroff. Their approach extends the work of Hoffman and Puzicha [18] which models the joint distribution of users and items through an aspect model that clusters users and items in a latent space. In order to deal with new items, instead of modelling the joint distribution of users and items, the authors propose to model the joint distribution of users and content features. At query time a "folding-in"' technique [17] is used to embed new items into the latent space so that items can be recommended. After careful analysis one may notice that the technique essentially boils down to building user profiles and applying pLSA to discover latent factors. Taking into account that previous studies have shown the correspondence between pLSA and NMF [15], one may clearly distinguish between this approach and our proposal. Instead of explicitly building user profiles and finding latent features, we discover a latent space common to both the content and collaborative information that allows us to link one to the other. The high memory requirements of this method prohibited us to include it in the comparison.

Rosen *et al.* [27] introduce the *author-topic model*, a generative model which extends Latent Dirichlet Allocation (LDA) to include authorship information. They associate each author with a multinomial distribution over topics, and each topic with multinomial distribution over words. Thus, a document with multiple authors can be modeled as a distribution over topics that is a mixture of the distribution associated with the authors. The model may be used to answer a range of interesting queries including: which topics an author writes about or who may be the authors of an unobserved document. Notice the similarity of the problem of author-topic modelling and the cold-start recommendations. One may associate the documents to the users who showed interest in them, instead of their authors. Thus, by using the same model, one may predict which users may be interested in a new document.

Gantner *et al.* [14] describe a method that maps item attributes to latent features of a matrix factorization model. This model can be used for item recommendation from implicit, positive-feedback only. In this paper, we are also considering the case of negative feedback as for the email recipient recommendation. If an item (*i.e.*, a contact) is not part of the recipient field, it is an indication of a negative feedback for this specific item.

Singh and Gordon [31] propose the idea of *collective matrix factorization*, a general framework for multi-relational factorization models. They subsume models on any number of relations as long as their loss function is a twice differentiable decomposable loss. In their work, they address both rating prediction and item recommendation. The matrix factorization approach proposed in this work is based on a similar idea of joint (collective) factorization. However, we impose non-negativity constraints on the factorization to obtain sparse and interpretable factors, and we consider the specific scenario of cold-start recommendations.

The majority of studies on NMF and its variations focus on the use of NMF as a clustering technique, rather than a prediction model. Badea *et al.* [2] introduce an extension of NMF, similar to the one proposed in this work, for clustering data from multiple sources. They apply the method on two gene expression datasets to uncover gene regulatory programs that are common to the two phenotypes. A similar model has been used by Akata and Thurau [1] in the context of tag prediction of Flickr images. The authors show anecdotal examples where the model predicts the correct tags, but no systematic evaluation or comparison with other methods is performed. The model introduced in this work differs in that we introduce: (1) a graph regularization to account for the intrinsic geometric structure of the data, and (2) Thikonov regularization to obtain more robust prediction models.

The idea of exploiting the local geometric structure of the data to discover better low-dimensional representations in NMF has been first proposed by Cai *et al.* [8]. Inspired by the success of using the nearest neighbour graph for label propagation in semi-supervised learning, they propose a clustering technique based on NMF. The algorithm favours factorizations for which similar instances have similar low-dimensional representations. The authors show that, by imposing this constraint, they outperform NMF and classical clustering algorithms. In this work, we build upon their findings by extending the model to multiple data sources, *i.e.*, the content and collaborative data matrices.

Finally, it is worth noting that there exists a line of research (e.g., [33] and [22]) that aims at addressing the cold-start scenario under the assumption that additional information is available, such as the social connections between the users or their tagging activities. However, we do not assume such additional information as input in our work.

3. PROBLEM STATEMENT

The scenario we consider is the item cold-start recommendation, where we would like to suggest new items — for which no interests has been expressed so far — to potentially interested users. Given a new item, it's corresponding description and the patterns of past activities of the users, we want to retrieve users who would likely manifest interest in this item.

More formally, we can define the problem as follows.

> At training time, we are given a collection of n items described by: (1) a set of m properties stored in a matrix $\mathbf{X_s} \in \mathbb{R}^{n\times m}$, where a row corresponds to an item and a column to an item property; and (2) a set of u users stored in a matrix $\mathbf{X_u} \in \mathbb{R}^{n\times u}$, where a cell (i, j) indicates if the user j has shown interest in item i. At test time, we are given a new item $\mathbf{q}$ with the corresponding description $\mathbf{q}_s \in \mathbb{R}^{1\times m}$, and our goal is to predict $\mathbf{q}_u \in \mathbb{R}^{1\times u}$, *i.e.*, how likely is a user to show interest in the new item.

To develop a deeper understanding of this class of problems in this study we consider two particular use cases: news article recommendation and email recipient recommendation.

News Recommendation. Online news platforms generally offer the possibility to users to engage in specific news by posting comments. In this case items are news articles and their description are the terms that appear in the news, *i.e.*, the content information. We also have information of which users commented on which news articles in the past, *i.e.*, the collaborative information. However, when a new article is published on the web site the collaborative information is not available as none of the users has commented on the article yet. Thus, given the content of the article and the past commenting patterns of the users, we would like to recommend the article to users who are most likely to comment.

Email Recipient Recommendation. Another item cold-start use case is when people write emails. They do not necessarily fill in the destination address first, but might rather start by writing the content. Moreover, they tend to write about specific topics to specific people. For instance, emails to employers may contain reports and discuss work activities, while emails to friends are more likely to be informal and discuss fun and personal issues. Thus, given the content of the email and the communication habits of the user, we would like to predict the most likely recipients and suggest them to the user. In this case items are the emails described by the words which appear in the message and the recipients of message represent the collaborative information.

4. A BRIEF REVIEW OF NMF

Non-negative Matrix Factorization (NMF) aims at decomposing a matrix $\mathbf{X}$ in two non-negative, lower dimensional matrices $\mathbf{W}$ and $\mathbf{H}$, such that their product can well approximate the original matrix $\mathbf{X}$, *i.e.*, $\mathbf{X} \approx \mathbf{WH}$. Unlike other matrix factorization techniques, such as SVD, it imposes non-negativity constraints on the resulting matrices. These constraints result in an additive effect that leads to a so-called "additive parts-based" representation of the data [11]. The discovered factors are sparse and easily interpretable, *i.e.*, the basis vectors naturally correspond to conceptual properties of the data. Moreover, the sparsity of the factors results in ligther model and easier to apply on new data [7].

5. JOINT NON-NEGATIVE MATRIX FACTORIZATION

In this section, we propose a new NMF formulation that jointly factorizes two data matrices and establishes a link between the two factorizations (Figure 1).

Given the problem defined, items are associated with a description and a set of users who consumed them. In the case of news, each news article is described by the set of words it contains and by all the users who commented on it. This information is then represented with two matrices, a document-term matrix $\mathbf{X_s} \in \mathbb{R}^{n\times v}$, and a document-user matrix $\mathbf{X_u} \in \mathbb{R}^{n\times u}$, where n is the number of documents, v is the vocabulary size and u is the number of users. The document-term matrix ($\mathbf{X_s}$) may be a boolean matrix or may represent the TF-IDF score of the words in the document. On the other hand, the entries of the document-user matrix ($\mathbf{X_u}$) reflect whether a given user commented on a

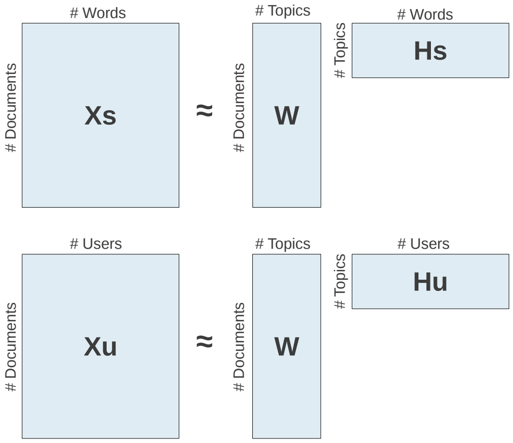

Figure 1: Graphical representation of the NMF model defined. The matrices $\mathbf{X_s}$ and $\mathbf{X_u}$ are observed, while the low-rank matrices $\mathbf{W}$, $\mathbf{H_s}$ and $\mathbf{H_u}$ are optimization parameters. Notice that the matrix $\mathbf{W}$ is common to both factorizations.

given article. As both matrices are non-negative, NMF can be applied to each of them. If decomposed separately, each factorization will represent a different hidden space, one for the users and one for the words. The idea of our approach is that both, documents and users, should be represented in a common latent space (Figure 1). In this "interpretable" hidden space, detected topics are supported by communities. In other words, each latent variable can be described by a set of words (i.e. a topic) but also a set of users (i.e. a community). To achieve this, we have to factorize both $\mathbf{X_s}$ and $\mathbf{X_u}$ jointly and enforce a low-dimensional representation in a common space. We call this method *Joint* NMF (JNMF).

Optimization Problem. More formally, given the matrices $\mathbf{X_s}$ and $\mathbf{X_u}$, we define the following optimization problem:

$$\min : J = \frac{1}{2}[\alpha||\mathbf{X_s} - \mathbf{W}\mathbf{H_s}||^2 + (1-\alpha)||\mathbf{X_u} - \mathbf{W}\mathbf{H_u}||^2 + \\ + \lambda(||\mathbf{W}||^2 + ||\mathbf{H_s}||^2 + ||\mathbf{H_u}||^2)] \quad (1)$$

$$s.t. \quad \mathbf{W} \geq \mathbf{0}, \mathbf{H_s} \geq \mathbf{0}, \mathbf{H_u} \geq \mathbf{0}$$

The first and the second term correspond to the factorization of the matrices $\mathbf{X_s}$ and $\mathbf{X_u}$, respectively. The common space representation is achieved by imposing one unique matrix $\mathbf{W}$ for both decompositions of $\mathbf{X_s}$ and $\mathbf{X_u}$. $\alpha \in [0, 1]$ is a hyper-parameter that controls the importance of each factorization. Setting $\alpha = 0.5$ gives equal importance to both factorizations, while values of $\alpha > 0.5$ (or $\alpha < 0.5$) give more importance to the factorization of $\mathbf{X_s}$ (or $\mathbf{X_u}$). The remaining terms are Tikhonov (Frobenius norm) regularization of $\mathbf{W}$, $\mathbf{H_u}$, and $\mathbf{H_s}$, controlled by the hyper-parameter $\lambda \geq 0$. It is used to enforce smoothness of the solution and avoid overfitting.

Optimization Algorithm. Similar to the classical NMF, the optimization problem defined is non-convex in terms of all parameters ($\mathbf{W}$, $\mathbf{H_s}$, $\mathbf{H_u}$) together. Thus, it is unrealistic to expect an algorithm to find the global minimum. In what follows, we derive an iterative algorithm based on multiplicative update rules which can achieve a stationary point.

The partial derivatives of J with respect to $\mathbf{W}$, $\mathbf{H_s}$, and $\mathbf{H_u}$ are:

$$\nabla_{\mathbf{W}} J = \alpha\mathbf{W}\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}} - \alpha\mathbf{X_s}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)\mathbf{W}\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}} - \\ - (1-\alpha)\mathbf{X_u}\mathbf{H_u}^{\mathrm{T}} + \lambda\mathbf{W}, \quad (2)$$

$$\nabla_{\mathbf{H_s}} J = \alpha\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{H_s} - \alpha\mathbf{W}^{\mathrm{T}}\mathbf{X_s} + \lambda\mathbf{H_s} \quad (3)$$

$$\nabla_{\mathbf{H_u}} J = (1-\alpha)\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{H_u} - (1-\alpha)\mathbf{W}^{\mathrm{T}}\mathbf{X_u} + \lambda\mathbf{H_u} \quad (4)$$

Applying the Karush-Kuhn-Tucker (KKT) first-order optimality conditions to J [10], we derive:

$$\begin{array}{llll} \mathbf{W} \geq \mathbf{0}, & \mathbf{Hs} \geq \mathbf{0}, & \mathbf{Hu} \geq \mathbf{0}, & (5) \\ \nabla_{\mathbf{W}} J \geq \mathbf{0}, & \nabla_{\mathbf{H_s}} J \geq \mathbf{0}, & \nabla_{\mathbf{H_u}} J \geq \mathbf{0}, & (6) \\ \mathbf{W} \odot \nabla_{\mathbf{W}} J = \mathbf{0}, & \mathbf{Hs} \odot \nabla_{\mathbf{H_s}} J = \mathbf{0}, & \mathbf{Hu} \odot \nabla_{\mathbf{H_u}} J = \mathbf{0}, & (7) \end{array}$$

where $\odot$ corresponds to the element-wise matrix multiplication operator.

Substituting the derivatives of J from Equations (2), (3) and (4) in Equation (7) leads to the following update rules:

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{[\alpha\mathbf{X_s}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)\mathbf{X_u}\mathbf{H_u}^{\mathrm{T}}]}{[\alpha\mathbf{W}\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)\mathbf{W}\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}} + \lambda\mathbf{W}]}, \quad (8)$$

$$\mathbf{H_s} \leftarrow \mathbf{H_s} \odot \frac{[\alpha\mathbf{W}^{\mathrm{T}}\mathbf{X_s}]}{[\alpha\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{H_s} + \lambda\mathbf{H_s}]}, \quad (9)$$

$$\mathbf{H_u} \leftarrow \mathbf{H_u} \odot \frac{[(1-\alpha)\mathbf{W}^{\mathrm{T}}\mathbf{X_u}]}{[(1-\alpha)\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{H_u} + \lambda\mathbf{H_u}]}, \quad (10)$$

where $\div$ denotes the element-wise matrix division operator. Regarding these update rules, we have the following theorem:

THEOREM 1. *The objective function J in Equation* (1) *is nonincreasing under the update rules in Equations* (8)*,* (9)*, and* (10)*. The objective function J is invariant under these updates if and only if $\mathbf{H_u}$, $\mathbf{H_s}$ and $\mathbf{W}$ are at a stationary point of the function.*

A detailed proof of the above theorem is given in the Appendix A. Notice, that the update rules in Equations (9) and (10) are the same as for the classical NMF formulation and have been proven in [12].

Inference. Once the model has been trained to learn $\mathbf{W}$, $\mathbf{H_s}$ and $\mathbf{H_u}$, we can use these factors for prediction. For instance, given the bag-of-words vector of a new news article $\mathbf{q}_s$, we can predict the users which are most likely to leave a comment, *i.e.*, $\mathbf{q}_u$. To do so, we project the document vector $\mathbf{q}_s$ to the common hidden space by solving the overdetermined system $\mathbf{q}_s = \mathbf{w}\mathbf{H_s}$ using the least squares method (with a projection to 0 of the negative values, see [7]). The vector $\mathbf{w}$, computed online, captures the factors — in the common hidden space — that explain mostly the observed news article $\mathbf{q}_s$. Then, by using this low dimensional vector $\mathbf{w}$ we may infer the missing part of the query: $\mathbf{q}_u \leftarrow \mathbf{w}\mathbf{H_u}$. Each element of $\mathbf{q}_u$ represents a score of how likely it is that the user will comment the new article. Then, given these scores, we may rank the users.

6. JOINT NMF WITH GRAPH REGULARIZATION

In the previous section, we have introduced JNMF, an NMF formulation that allows us to jointly factorize two data matrices. In this section, we extend this model by adding an additional term that takes into account the local geometric structure of the data.

Recall that when performing JNMF factorization, we attempt to find a common low-dimensional space that is optimized for the linear approximation of the data from both views. We also suppose that the data from both views are drawn from a common distribution P. One may hope that additional knowledge of the distribution P can be exploited for a better discovery of the low-dimensional space. A natural assumption could be that if two data points x_i and x_j, in any view, are close in the intrinsic geometry of the distribution, then the representations of these two data points in the low-dimensional space should also be close to each other. This assumption is commonly referred to as *manifold assumption* and plays an essential role in algorithms for dimensionality reduction [5] and semi-supervised learning [6,34].

In reality the geometric structure of the distribution P is not known and cannot be directly used. However, recent studies on spectral graph theory [9] and manifold learning [4] have demonstrated that the local geometric structure can be effectively modeled through a nearest neighbor graph on a scatter of data points. Consider a graph with n nodes where each node represents a data point. For each point we find the p nearest neighbors and we connect the corresponding nodes in the graph. The edges may be binary (1 if one of the nearest neighbors, 0 otherwise) or may be weighted (e.g., cosine similarity). This results in a matrix $\mathbf{A}$ which can later be used to measure the local closeness of two points x_i and x_j.

Recall that the Joint NMF maps each data point x_i into a low-dimensional representation w_i (a row of the matrix $\mathbf{W}$). A natural way to measure the distance between two low dimensional representations, given the choice of a loss function, is by computing the Euclidean distance: $||w_i - w_j||^2$. Using the above defined weight matrix $\mathbf{A}$ we may measure the smoothness of the low dimensional representation as:

$$\begin{aligned} S &= \frac{1}{2}\sum_{i,j=1}^{n} ||w_i - w_j||^2 \mathbf{A}_{ij} \\ &= \sum_{i=1}^{n}(w_i^T w_i)\mathbf{D}_{ij} - \sum_{i,j=1}^{n}(w_i^T - w_j)\mathbf{A}_{ij} \\ &= \mathrm{Tr}(\mathbf{W}^{\mathrm{T}}\mathbf{D}\mathbf{W}) - \mathrm{Tr}(\mathbf{W}^{\mathrm{T}}\mathbf{A}\mathbf{W}) = \mathrm{Tr}(\mathbf{W}^{\mathrm{T}}\mathbf{L}\mathbf{W}), \end{aligned}$$

where $\mathbf{D}$ is a diagonal matrix whose entries are the row sums of $\mathbf{A}$ (or column, as $\mathbf{A}$ is symmetric), *i.e.*, $\mathbf{D}_{ii} = \sum_i \mathbf{A}_{ij}$. $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is called the Laplacian matrix of the graph [9] and $\mathrm{Tr}(\bullet)$ is the trace operator.

New Optimization Problem. Given the above, we modify the JNMF formulation as to enforce smooth low-dimensional representations of the data. This leads to JNMF with *Graph Regularization* (JNMF-GR):

$$\begin{aligned} \min : J = \frac{1}{2}[\alpha||\mathbf{X_s} - \mathbf{W}\mathbf{H_s}||^2 + (1-\alpha)||\mathbf{X_u} - \mathbf{W}\mathbf{H_u}||^2 + \\ + \beta\mathrm{Tr}(\mathbf{W}^{\mathrm{T}}\mathbf{L}\mathbf{W}) + \lambda(||\mathbf{W}||^2 + ||\mathbf{H_s}||^2 + ||\mathbf{H_u}||^2)] \\ s.t. \quad \mathbf{W} \geq \mathbf{0}, \mathbf{H_s} \geq \mathbf{0}, \mathbf{H_u} \geq \mathbf{0}, \end{aligned} \tag{11}$$

where $\mathbf{L}$ is the Laplacian matrix of the graph, and β is a hyper-parameter which controls the extent to which smoothness is enforced. It is easy to check that when $\beta = 0$ the formulation is equivalent to JNMF.

Optimization algorithm. As in the JNMF formulation, the optimization problem of Equation (11) is non-convex with respect to all optimization parameters together ($\mathbf{W}$, $\mathbf{H_s}$ and $\mathbf{H_u}$) and thus we may not guarantee to find the global minimum. In the remainder of this section, we follow a similar procedure to derive an iterative algorithm that can achieve a stationary point.

The partial derivatives with respect to $\mathbf{H_s}$ and $\mathbf{H_u}$ remain the same as in the Joint NMF formulation (Equations (3) and (4), respectively), while the partial derivative with respect to $\mathbf{W}$ becomes:

$$\begin{aligned} \nabla_{\mathbf{W}} J = \alpha\mathbf{W}\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}} - \alpha\mathbf{X_s}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)\mathbf{W}\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}} - \\ - (1-\alpha)\mathbf{X_u}\mathbf{H_u}^{\mathrm{T}} + \beta\mathbf{L}\mathbf{W} + \lambda_W\mathbf{W}. \end{aligned} \tag{12}$$

Applying the KKT first-order optimality conditions results in the same equations as in the Joint NMF formulation, *i.e.*, Equations (5), (6), and (7). Substituting the partial derivatives of J in Equation (7) results in the same update rules for $\mathbf{H_s}$ and $\mathbf{H_u}$ (Equations (9) and (10), respectively), while the update rule for $\mathbf{W}$ becomes:

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{[\alpha\mathbf{X_s}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)\mathbf{X_u}\mathbf{H_u}^{\mathrm{T}} + \beta\mathbf{A}\mathbf{W}]}{[\alpha\mathbf{W}\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)\mathbf{W}\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}} + \beta\mathbf{D}\mathbf{W} + \lambda\mathbf{W}]}, \tag{13}$$

Similarly, we have the following theorem:

THEOREM 2. *The objective function J in Equation* (11) *is nonincreasing under the update rules in Equations* (9), (10), *and* (13). *The objective function J is invariant under these updates if and only if* $\mathbf{H_u}$, $\mathbf{H_s}$ *and* $\mathbf{W}$ *are at a stationary point of the function.*

See Appendix B for a detailed proof of the above theorem.

7. EXPERIMENTAL EVALUATION

In this section we present a series of experiments to evaluate the performance of JNMF and JNMF-GR in the item cold-start scenario. We first describe the baselines and then we compare the performance of JNMF and JNMF-GR with the baseline approaches in two item cold-start use cases: email recipient recommendation (based on explicit feedback) and news recommendation (based on implicit feedback). Finally, we analyse the parameter settings and the running time of the proposed models.

7.1 Baselines for Comparison

We compare the two methods proposed, JNMF and JNMF-GR, to four other approaches: pure content-based recommender, content-topic-based recommender, LSI applied on the author profiles and the author topic-model.

Content-based Recommender (CB). We build a profile of each user based on the properties of the items preferred in the past. Experimentally we find that weighting each item inversely proportional to the number of users that interacted with the item leads to an improved performance. Thus, in the user profile, very popular items are given less importance, while less popular items are given more importance. More formally, a user profile U is defined as: $U = \sum_{i \in I}(\vec{v_i}/freq_i)$, where I is the set of items the user interacted with in the past, $\vec{v_i}$ is the description of item i and $freq_i$ is the number of users that interacted with i. At test time, we rank the items by computing the cosine similarity between the new items and the user profiles.

Content-topic-based Recommender (CTB). We extract topics from the content of the items by applying NMF and we describe each item as a mixture of the topics extracted. We then build a topical profile for each user based on the topics of the items the user interacted with in the past. At test time, we infer the topics of the new items and we rank the items based on the cosine similarity between the item's topics and the users' topical profiles. The CTB recommender allows us to investigate the importance of performing joint factorization of both the content and collaborative matrix, instead of factorizing only the content matrix.

LSI on the User Profiles (UP-LSI). We apply the hybrid recommendation system proposed in [32] (see Section 2). The approach combines the content and collaborative information by building user profiles and applying Latent Semantic Indexing (LSI) to discover latent factors. At test time, the new items are projected in the latent space and compared to the user profiles. Finally, the items are recommended to the users with the most similar profiles.

Author-topic Model (ATM). The author-topic model [27] is a generative probabilistic model which extends LDA to include authorship information (see Section 2). It associates each author with a multinomial distribution over topics, and each topic with multinomial distribution over words. As the authors point out, the model may not only be used to find the topics associated with the authors, but also to predict the authors of unobserved documents. In the email recipient recommendation experiment we model the recipients as authors, while in the news recommendation scenario we model the users as authors. As recommended by the authors, we set the parameters as: $\alpha = 50/k$, where k is the number of topics, $\beta = 0.01$, and we perform 500 iterations of the Gibbs Sampler.

7.2 Email Recipient Recommendation

When people write emails they do not necessarly start by filling in the recipient address (*i.e.*, the "to" field), but may start by writing the body of the message. Given the content of the message and the messaging habits of the user, *i.e.*, with whom the user exchanged messages with similar content in past, we would like to predict the most likely recipients of the new message. In this experiment, we test the accuracy of the recipient recommendations produced by the proposed methods against the four baseline techniques.

Dataset. The data is composed of email messages released during investigation of the Federal Energy Regulatory Commission against the Enron Corporation. We consider the 10 largest mailboxes and within each mailbox only the emails sent by the owner. The total number of emails is 36,010, sent to 4,984 recipients. The size of the vocabularies for each mailbox ranges from 12,375 to 56,193 unique tokens. The messages have been preprocessed by removing the headers (from/to/cc fields), converting all tokens to lower case and removing numbers, stop-words and infrequent (appearing < 5 time) tokens.

Evaluation Metrics. The output of the algorithms is a ranking of the past recipients of how likely they are to be recipients of the new email. The feedback from the users is explicit, *i.e.*, we have ground truth of who are the recipients of the email as specified by the user. To evaluate the ranking produced by each algorithm we use the state-of-the-art metrics from Information Retrieval: Micro and Macro F1, Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG) [3].

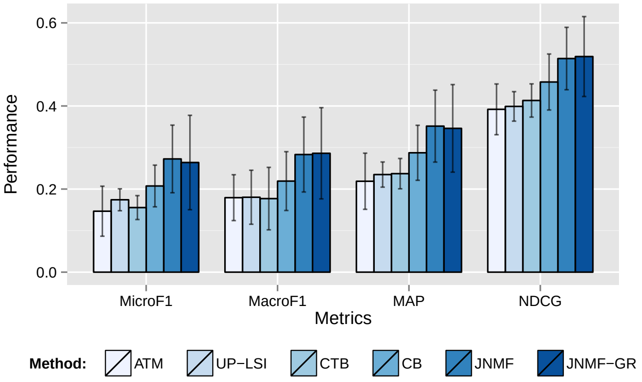

Figure 2: Comparison of the different methods on the Enron dataset.

Evaluation Protocol. As the data is intrinsically influenced by the time, we sort the 10 mailboxes messages chronologically. We divide the messages in 80% training and 20% testing, resulting in 10 independent train/test subsets. Only the recipients which appear in the training period are considered as potential receivers. We tune the hyper-parameters of each method on an independent validation set, 10% of the training set. For the JNMF-GR method, we build a binary nearest neighbour graph, where we connect the each email massage with the two other most similar messages. Finally, we evaluate the statistical significance of the differences in performance by using a Wilcoxon signed ranks test [13].

Results. Figure 2 shows the average performance and the standard deviation of each method across the 10 mailboxes. JNMF and JNMF-GR perform better than the other methods in all measures with differences ranging from 4%-13%. All differences are statistically significant ($p < 0.005$). On the other hand, JNMF and JNMF-GR achieve similar results, without statistically significant difference. This indicates that the graph regularization does not bring any additional information in the case of emails. One explanation may be that emails are a user generated content and as such contain a lot of noise, such as misspellings or informal expressions, that leads to inaccurate nearest neighbour graphs.

7.3 News Recommendation

To improve the user experience, online news platforms allow users to engage with the articles by posting comments. Moreover, to encourage the user engagement on the platform the users are recommended articles that they may be interested in. We consider the item cold-start scenario, *i.e.*, when a new article is published and none of the users have commented on it yet. Thus, given the content of the new articles and the past commenting patterns we would like to recommend to the users the articles that they are most likely to comment.

Dataset. We consider a random sample of news articles and the corresponding comments posted on the Yahoo! News website during a period of 40 days. The dataset contains

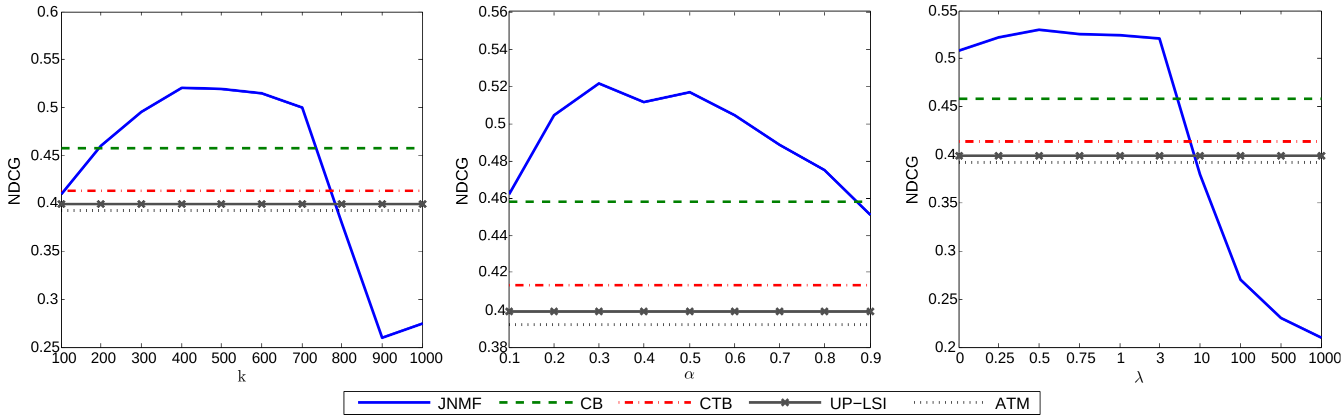

Figure 3: Behaviour of the JNMF hyper-parameters. Left: k, number of topics; Middle: α, weight of the content versus the collaborative information; and λ, the smoothness of the solution. The results are averaged on 10 runs (expect ATM, only one run) of the methods on Tana Jones' mailbox from the Enron data set.

∼41K articles, enclosing ∼3.5M comments posted by ∼650K users. The size of the vocabulary is ∼60K (*i.e.*, number of unique tokens in all articles) and ∼9M tokens. The content of the articles has been preprocessed such that all tokens are converted to lower case, and stop-words, digits, punctuation, short (< 3 characters) and infrequent (appearing < 3 times) tokens are removed.

Evaluation Metrics. Similar to the previous experiment, the output of each algorithm is a ranking. In this experiment, however, we do not have an explicit feedback of which news articles were undesired by the users. While, commenting an article is an evidence of the user's interest in it, the absence of a comment is not an indication that the recommended article was undesired, as not commenting may stem from multiple different reasons. Therefore, we adopt the average percentile ranking, a measure proposed in [19] and widely used to evaluate ranking based on implicit feedback (e.g., [23,26,28]). We define $rank_{u,i}$ as the percentile ranking of article i in the ranked list of articles for the user u; if $rank_{u,i} = 0\%$, then the article i is predicted to be the most interesting for u, while $rank_{u,i} = 100\%$ implies that the article is predicted to be the least interesting. Our quality measure is then the total average percentile ranking of an article:

$$\overline{rank} = \frac{\sum_{u,i} comment_{u,i} \cdot rank_{u,i}}{\sum_{u,i} comment_{u,i}},$$

where $comment_{u,i}$ is an indicator function that equals to: 1 if the user u commented on article i; and 0 otherwise. The lower $\overline{rank}$, the better the quality of the ranking. For random predictions, the expected value of $\overline{rank}$ is 50%. Thus, if $\overline{rank} < 50\%$, then the algorithm is better than random. To ease illustration, we convert the percentile ranking into *ranking accuracy* (RA). That is 1 (best/ideal predictions), if the percentile ranking is 0%; and it is 0 (random predictions), if the percentile ranking is 50%:

$$Ranking\ Accuracy = \frac{50\% - \overline{rank}}{50\%}.$$

We evaluate the Ranking Accuracy (RA) at different positions: 3, 5, 7 and 10.

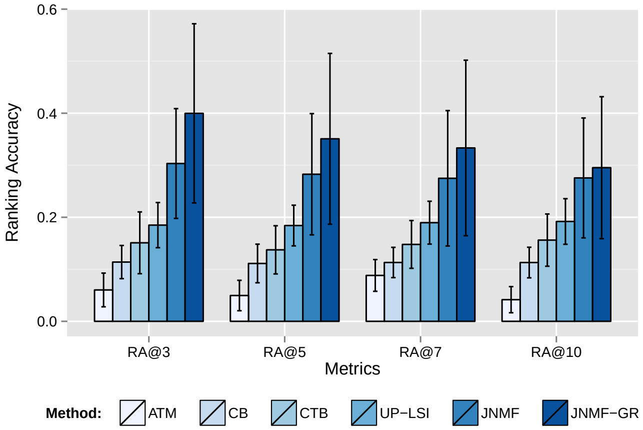

Figure 4: Comparison of the different methods on the Yahoo! News dataset.

Evaluation Protocol. We sort the data chronologically and we produce train/test subsets by shifting a time window, instead of sampling at random. We train using the past 30 days and we predict next day comments, shifting for one day at a time, resulting in 10 independent folds. We also restrict our test set to those users who have commented at least once in the training period. We tune the hyper-parameters of each method on an independent validation set, 10% of the training set, and we evaluate the statistical significance of the differences in performance by using Wilcoxon signed ranks test [13]. For the JNMF-GR method, we build a binary nearest neighbour graph, where we connect the each article with the three other most similar ones.

Results. We evaluate the different methods in each of the 10 testing days and we compute the average performance (Figure 4). All algorithms perform better then random, *i.e.*, RA $> 0\%$. JNMF and JNMF-GR outperform all other methods with statistically significant differences ($p < 0.05$). JNMF-GR achieves better ranking accuracy then JNMF in all positions, however the difference diminishes as we consider larger lists. The difference is statistically significant only for RA@3 and RA@5 ($p < 0.05$). This indicates that considering the local geometric structure of the data allows the algorithm to push the relevant items towards the top of the list. As users are usually presented a short list of recommendations, making accurate recommendations on the top of the list is crucial for improving the satisfaction of the users.

7.4 Parameter Analysis

The JNMF model has three essential parameters: k, number of latent variables, *i.e.*, topics; α, weight of the content versus the collaborative information; and λ, controlling the smoothness of the solution. Figure 3 shows a typical behaviour of the algorithm for different values of the parameters. The results are averaged over 10 runs of all algorithms on one mailbox of the Enron dataset.

The parameter k controls the complexity of the model. Small values of k, *i.e.*, simple models under-fit whereas large values of k over-fit the data and lead to poor performance (Figure 3, left). Thus, one has to find a balance between the two that fits best the problem at hand. Furthermore, balancing the importance of the content versus the collaborative information, *i.e.*, $\alpha \approx 0.5$ tends to achieve the best performance. Figure 3 (middle) also suggests that giving slightly more importance to the collaborative information (*e.g.*, $\alpha \in [0.2, 0.5]$) may be helpful. Finally, imposing the smoothness of the solutions helps. However, imposing it too strongly, *i.e.*, large values of λ, decreases the performance. Setting λ between 0 and 1 leads to stable and high performance (Figure 3, right) .

7.5 Running Time Analysis

In this section, we measure the cpu time required by the methods under different settings of the model complexity. All models are comparably fast at inference time, as they require only simple operations such as projections in the latent space or computing similarities. Hence, we only report the running times needed to train the models.

For this experiment, we consider one mailbox from the Enron dataset (4K messages, 500 recipients, 18K unique terms) on which we perform 10 runs of each method under different values of the hyper-parameter k. The averaged cpu times are reported in Figure 5. Due to the long computation time required by the author-topic model we test up to $k = 500$ and we perform only one run for each k. In the case of UP-LSI k is bounded by 500 due to the rank of the matrix.

The content-based recommender (CB) takes less time for training as it only requires building the user profiles. Little time is also required by the UP-LSI method relying on a fast sparse SVD implementation. The ATM, however, is the computationally most expensive method taking between 3 and 35 hours to train. The JNMF requires more time then CB and UP-LSI, but is faster than the CTB, JNMF-GR and the ATM. The JNMF-GR, on the other hand, is slightly slower than other methods, except for ATM. Both methods, the JNMF and JNMF-GR, are reasonably fast and require 10, *i.e.*, 25 minutes to train for the highest values of k. This makes frequent updates of the model possible.

7.6 Reproducibility of the Experiments

The Matlab implementations of the Joint NMF (JNMF) and the Joint NMF with Graph Regularization (JNMF-GR) are made publicly available at: https://github.com/jnmf/demo. As discussed in Section 7.4, the parameters may be set as: $\alpha = 0.5$, $\beta = 0.05$, and $\lambda = 0.5$, while the parameter k depends on the data and needs to be tuned. A Matlab implementation of the Author-topic Model is publicly available as part of the Matlab Topic Modeling Toolbox at: http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm. Finally, the Enron dataset is available at: https://www.cs.cmu.edu/~enron/. In the experiments, we consider the 10 largest mailboxes owned by: Steven Kean, Vince Kaminski, Jeff Dasovich, Sally Beck, Tana Jones, Mark Haedicke, Sara Shackleton, Mark Taylor, John Lavorato, and Louise Kitchen. The parameter and run time analysis experiments are performed on the mailbox of Tana Jones.

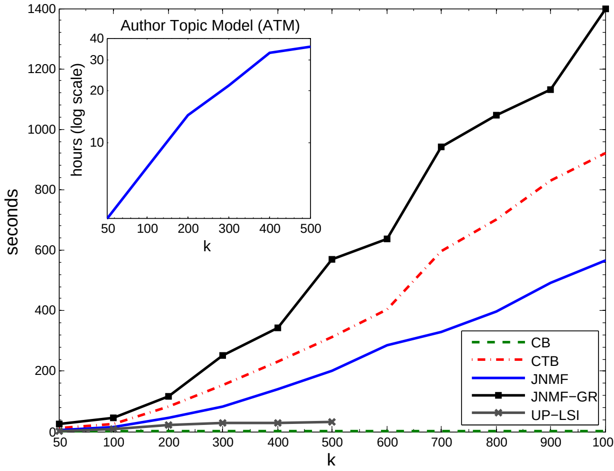

Figure 5: Running Time Analysis of ATM, CB, CTB, JNMF, JNMF-GR and UP-LSI. We report the cpu times in seconds averaged on 10 runs (expect ATM, only one run) of the methods on Tana Jones' mailbox from the Enron data set. The machine used has an Intel(R) Xeon(R) CPU E5620 @ 2.40GHz with 12288 KB cache size and 64GB of RAM. All the methods have been implemented and run with Matlab R2011b 64-bit.

8. CONCLUSIONS AND FUTURE WORK

To overcome the item cold-start, in this work we have proposed JNMF, a recommender system that combines content and collaborative information in a unified matrix factorization framework. We have also introduced a extension of it, JNMF-GR, that takes into account the local geometric structure of the data and enforces smoothness of the solutions. Finally, we have experimentally shown that the two proposed methods outperform the existing item-cold start recommenders. Interestingly, in case of rich content (*e.g.*, news articles) graph regularization improves the ranking accuracy in the top positions of the ranking which is crucial for improving the user satisfaction on news platforms.

Towards Online Adaptive Models. In real world, the models need to be updated as new data is arriving. In this context, time plays in important role when modeling user preferences [20]. In this line of research, recently, McAuley *et al.* modeled user tastes evolution and showed an improvement in recommendations [24]. As far as we know none of the existing "time-aware" approaches have been applied to the item cold-start recommendations. As a future work we consider extending our models to close this gap.

Power Law Behaviors. As one may expect, in the Yahoo! News as well as in the Enron data set, both words and users are power law distributed. This phenomenon may reduce the coverage as several topics and communities may dominate the recommender system, *i.e*, new items will always be suggested to the same group of already active users. However, to increase user engagement one has to target also the less active users who are part of the long tail. Recent studies [29] in matrix completion have shown that introducing a *weighted trace norm regularization* leads to significant improvement of the performance when entries of the matrix are sampled non-uniformly. As future work, we would like to investigate whether introducing this kind of regularization in our joint factorization models will improve accuracy.

9. REFERENCES

[1] Z. Akata, C. Thurau, and C. Bauckhage. Non-negative matrix factorization in multimodality data for segmentation and label prediction. In *Computer Vision Winter Workshop*, 2011.

[2] L. Badea. Extracting gene expression profiles common to colon and pancreatic adenocarcinoma using simultaneous nonnegative matrix factorization. In *Pacific Symposium on Biocomputing*, 2008.

[3] R. Baeza-Yates and Ribeiro-Neto. *Modern information retrieval*. ACM press New York, 1999.

[4] M. Belkin. *Problems of learning on manifolds*. PhD thesis, The University of Chicago, 2003.

[5] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems*, 2001.

[6] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 2006.

[7] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 2007.

[8] D. Cai, X. He, X. Wu, and J. Han. Non-negative matrix factorization on manifold. In *International Conference on Data Mining*, 2008.

[9] F. Chung. *Spectral Graph Teory*. AMS, 1997.

[10] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. Wiley, 2009.

[11] L. Daniel and S. Sebastian. Learning the parts of objects by non-negative matrix factorization. *Nature*, 1999.

[12] L. Daniel and S. Sebastian. Algorithms for non-negative matrix factorization. In *Neural Information Processing Systems*, 2000.

[13] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 2006.

[14] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *International Conference on Data Mining*, 2010.

[15] E. Gaussier and C. Goutte. Relation between plsa and nmf and implications. In *Special Interest Group on Information Retrieval*, 2005.

[16] A. T. Gediminas Adomavicius. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 2005.

[17] T. Hofmann. Probabilistic latent semantic indexing. In *Special Interest Group on Information Retrieval*, 1999.

[18] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *International Joint Conferences on Artificial Intelligence*, 1999.

[19] Y. Hu, Y. Koren, and C. Volinsky. Collaborative Filtering for Implicit Feedback Datasets. In *International Conference on Data Mining*, 2008.

[20] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 2010.

[21] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

[22] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *Conference on Web search and data mining*, 2012.

[23] N. Lathia, J. Froehlich, and L. Capra. Mining public transport usage for personalised intelligent transport systems. In *International Conference on Data Mining*, 2010.

[24] J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *World Wide Web Conference*, 2013.

[25] A. K. Menon and C. Elkan. Link prediction via matrix factorization. In *Machine Learning and Knowledge Discovery in Databases*, 2011.

[26] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo, and J. Crowcroft. Recommending social events from mobile phone location data. In *International Conference on Data Mining*, 2010.

[27] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Uncertainty in Artificial Intelligence*, 2004.

[28] D. Saez-Trumper, D. Quercia, and J. Crowcroft. Ads and the city: considering geographic distance goes a long way. In *ACM conference on Recommender systems*, 2012.

[29] R. Salakhutdinov and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Neural Information Processing Systems*, 2012.

[30] A. Schein, A. Popescul, L. Ungar, and D. Pennock. Methods and metrics for cold-start recommendations. In *Special Interest Group on Information Retrieval*, 2002.

[31] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *ACM Conference on Knowledge Discovery and Data Mining*, 2008.

[32] I. Soboroff. Combining content and collaboration in text filtering. In *IJCAI Workshop on Machine Learning for Information Filtering*, 1999.

[33] D. Yin, S. Guo, B. Chidlovskii, B. D. Davison, C. Archambeau, and G. Bouchard. Connecting comments and tags: improved modeling of social tagging systems. In *Conference on Web search and data mining*, 2013.

[34] X. Zhu and J. Lafferty. Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *International Conference on Machine learning*, 2005.

APPENDIX

A. PROOF OF THEOREM 1

The Joint NMF objective function J (Equation 1) is certainly bounded from below by zero. To prove Theorem 1, we need to show that J is non-increasing under the updating steps in Equations 8, 9, and 10. The multiplicative update rules for $\mathbf{H_s}$ and $\mathbf{H_u}$ are exactly the same as in the original NMF, thus we can use the convergence proof of NMF to show that J is non-increasing under the update steps in Equations 9 and 10 (see [12] for details). Thus, we only need to prove that J is non-increasing under the update step for $\mathbf{W}$ (Equation 8).

Since the objective function J can be decoupled to considering only one instance at time, *i.e.*, one row of $\mathbf{X_s}$, $\mathbf{X_u}$ and $\mathbf{W}$, we can write J as:

$$\begin{aligned}\min : J =& \frac{1}{2}(\alpha||x_s^{\mathrm{T}} - w^{\mathrm{T}}\mathbf{H_s}||_{\mathrm{F}}^2 + (1-\alpha)||x_u^{\mathrm{T}} - w^{\mathrm{T}}\mathbf{H_u}||_{\mathrm{F}}^2 + \\ &+ \lambda||w^{\mathrm{T}}||_2^2 + \lambda||\mathbf{H_s}||_{\mathrm{F}}^2 + \lambda||\mathbf{H_u}||_{\mathrm{F}}^2),\end{aligned}$$

minimizing it with respect to each of the rows of $\mathbf{W}$ separately.

We consider a current approximation $\hat{w}^{\mathrm{T}}$ of the solution and we formulate the following problem:

$$\min : \hat{J}(w^{\mathrm{T}}) = J(w^{\mathrm{T}}) + \frac{1}{2}(w^{\mathrm{T}} - \hat{w}^{\mathrm{T}})^{\mathrm{T}} S (w^{\mathrm{T}} - \hat{w}^{\mathrm{T}}),$$

where $S = Diag(x) - (\alpha\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}} + \lambda)$, with $x = \frac{\hat{w}^{\mathrm{T}}(\alpha\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}} + \lambda)}{\hat{w}^{\mathrm{T}}}$.

Since S is positive semi-definite matrix [12], we have that $\hat{J}(w^{\mathrm{T}}) \geq J(w^{\mathrm{T}})$ for all w^{T} and specifically $\hat{J}(\hat{w}^{\mathrm{T}}) = J(\hat{w}^{\mathrm{T}})$.

Furthermore, the function is also convex. We set the derivative of $\hat{J}(\hat{w}^{\mathrm{T}})$ to zero, *i.e.*,

$$\begin{aligned}\frac{\partial \hat{J}}{\partial w^{\mathrm{T}}} =& \alpha w^{\mathrm{T}}\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}} - \alpha x_s^{\mathrm{T}}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)w^{\mathrm{T}}\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}} - \\ &- (1-\alpha)x_u^{\mathrm{T}}\mathbf{H_u}^{\mathrm{T}} + \lambda\hat{w}^{\mathrm{T}} + (w^{\mathrm{T}} - \hat{w}^{\mathrm{T}})S = 0\end{aligned}$$

in order to obtain the minimizer $w^{\mathrm{T}*}$:

$$\begin{aligned}w^{\mathrm{T}*}(\alpha\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}} + \lambda + S) &= \\ = \alpha x_s^{\mathrm{T}}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)x_u^{\mathrm{T}}\mathbf{H_u}^{\mathrm{T}} + \hat{w}^{\mathrm{T}}S&,\end{aligned}$$

notice that $\hat{w}^{\mathrm{T}}S = 0$ (by the construction of S).

Expanding S and cancelling terms, we obtain:

$$\begin{aligned}w^{\mathrm{T}*}\cdot diag^{-1}(\hat{w}^{\mathrm{T}})\cdot diag(\hat{w}^{\mathrm{T}}(\alpha\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}} + \\ +\lambda)) = \alpha x_s^{\mathrm{T}}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)x_u^{\mathrm{T}}\mathbf{H_u}^{\mathrm{T}}.\end{aligned}$$

Notice that multiplying vector by a diagonal matrix formed by another vector, corresponds to performing an element-wise product between the two vectors. Thus, we obtain:

$$\begin{aligned}w^{\mathrm{T}*} = \hat{w}^{\mathrm{T}} &\odot (\alpha x_s^{\mathrm{T}}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)x_u^{\mathrm{T}}\mathbf{H_u}^{\mathrm{T}}) \\ &\oslash (\alpha\hat{w}^{\mathrm{T}}\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}} + (1-\alpha)\hat{w}^{\mathrm{T}}\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}} + \lambda\hat{w}^{\mathrm{T}}).\end{aligned}$$

Leading to the multiplicative update rule of Equation 8.

Since, $w^{\mathrm{T}*}$ is the global minimizer of $\hat{J}(w^{\mathrm{T}})$, we have $\hat{J}(w^{\mathrm{T}*}) \leq \hat{J}(\hat{w}^{\mathrm{T}})$. Moreover, $\hat{J}(w^{\mathrm{T}})$ is constructed to satisfy $\hat{J}(w^{\mathrm{T}}) \geq J(w^{\mathrm{T}})$ for all w^{T}. This implies that $J(w^{\mathrm{T}*}) \leq \hat{J}(w^{\mathrm{T}*}) \leq \hat{J}(\hat{w}^{\mathrm{T}}) = J(\hat{w}^{\mathrm{T}})$ *i.e.* we have a decrease of the objective function. □

B. PROOF OF THEOREM 2

Similar to Theorem 1, J (Equation 11) is bounded from below by zero and the update rules for $\mathbf{H_s}$ and $\mathbf{H_u}$ are the same as in the original NMF formulation. Thus, we only need to prove that J is non-increasing under the update step for $\mathbf{W}$ (Equation 13). We will follow a procedure based on auxiliary functions, similar to the one described in [8].

Definition. $G(w, w')$ is an auxiliary function for $J(w)$ if the conditions:

$$G(w, w') \geq J(w), \quad G(w, w) = J(w)$$

are satisfied. The auxiliary function is very useful because of the following lemma.

Lemma. If G is an auxiliary function of J, then J is non-increasing under the update:

$$w^{(t+1)} = \arg\min_{w} G(w, w^{(t)}) \tag{14}$$

Proof.

$$J(w^{(t+1)}) \leq G(w^{(t+1)}, w^{(t)}) \leq G(w^{(t)}, w^{(t)}) = J(w^{(t)}).$$

For brevity throughout the proof we use $\gamma = 1 - \alpha$. We start by rewriting the objective function of JNMF-GR in Equation 11 as follows:

$$\begin{aligned}\min : J = \frac{1}{2}(\alpha\sum_{i=1}^{N}\sum_{j=1}^{M}(x_{ij}^{s} - \sum_{k=1}^{K}w_{ik}h_{kj}^{s})^2 + \\ + \gamma\sum_{i=1}^{N}\sum_{j=1}^{F}(x_{ij}^{u} - \sum_{k=1}^{K}w_{ik}h_{kj}^{u})^2 + \beta\sum_{k=1}^{K}\sum_{j=1}^{N}\sum_{l=1}^{N}w_{jk}[\mathbf{L}]_{jl}w_{lk} + \\ + \lambda\sum_{i=1}^{N}\sum_{j=1}^{K}w_{ij}^2 + \lambda\sum_{i=1}^{K}\sum_{j=1}^{M}(h_{ij}^{s})^2 + \lambda\sum_{i=1}^{K}\sum_{j=1}^{F}(h_{ij}^{u})^2).\end{aligned}$$

Considering any element w_{ab} of $\mathbf{W}$, we use J_{ab} to denote the part of J which is only relevant to w_{ab}. It is easy to check that:

$$\begin{aligned}J'_{ab} = [\nabla_{\mathbf{W}}J]_{ab} =& [\alpha\mathbf{W}\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}} - \alpha\mathbf{X_s}\mathbf{H_s}^{\mathrm{T}} + \gamma\mathbf{W}\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}} - \\ &- \gamma\mathbf{X_u}\mathbf{H_u}^{\mathrm{T}} + \beta\mathbf{L}\mathbf{W} + \lambda\mathbf{W}]_{ab}, \\ J''_{ab} = \alpha[\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}}]_{bb}& + (1-\alpha)[\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}}]_{bb} + \beta[\mathbf{L}]_{aa} + \lambda.\end{aligned}$$

Since our update is essentially element-wise, it is sufficient to show that each J_{ab} is non-increasing under the update step of Equation 13.

We define:

$$\begin{aligned}G(w, w_{ab}^{(t)}) = J_{ab}(w_{ab}^{(t)}) + J'_{ab}(w_{ab}^{(t)})(w - w_{ab}^{(t)}) + \frac{\alpha[\mathbf{W}\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}}]_{ab}}{w_{ab}^{(t)}} \\ + \frac{\gamma[\mathbf{W}\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}}]_{ab} + \beta[\mathbf{D}\mathbf{W}]_{ab} + \lambda[\mathbf{W}]_{ab}}{w_{ab}^{(t)}}(w - w_{ab}^{(t)})^2\end{aligned}$$

is an auxiliary function for J_{ab}, the part of J which is only relevant to w_{ab}.

Since $G(w, w) = J_{ab}(w)$ is obvious, we need only show that $G(w, w_{ab}^{(t)}) \geq J_{ab}(w)$. To do this, we compare the Tylor series expansion of $J_{ab}(w)$:

$$\begin{aligned}J_{ab}(w) = J_{ab}(w_{ab}^{(t)}) + J'_{ab}(w_{ab}^{(t)})(w - w_{ab}^{(t)}) + (\alpha[\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}}]_{bb} + \\ + \gamma[\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}}]_{bb} + \beta[\mathbf{L}]_{aa} + \lambda)(w - w_{ab}^{(t)})^2,\end{aligned}$$

with $G(w, w_{ab}^{(t)})$ to find that $G(w, w_{ab}^{(t)}) \geq J_{ab}(w)$ is equivalent to:

$$\begin{aligned}\frac{\alpha[\mathbf{W}\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}}]_{ab} + \gamma[\mathbf{W}\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}}]_{ab} + \beta[\mathbf{D}\mathbf{W}]_{ab} + \lambda[\mathbf{W}]_{ab}}{w_{ab}^{(t)}} \geq \\ \geq \alpha[\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}}]_{bb} + \gamma[\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}}]_{bb} + \beta[\mathbf{L}]_{aa} + \lambda.\end{aligned}$$

We have:

$$\alpha[\mathbf{W}\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}}]_{ab} = \alpha\sum_{l=1}^{k}w_{al}^{(t)}[\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}}]_{lb} \geq \alpha w_{ab}^{(t)}[\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}}]_{bb},$$

$$\gamma[\mathbf{W}\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}}]_{ab} = \gamma\sum_{l=1}^{k}w_{al}^{(t)}[\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}}]_{lb} \geq \gamma w_{ab}^{(t)}[\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}}]_{bb},$$

$$\begin{aligned}\beta[\mathbf{D}\mathbf{W}]_{ab} = \beta\sum_{j=1}^{N}[\mathbf{D}]_{aj}w_{jb}^{(t)} &\geq \beta[\mathbf{D}]_{aa}w_{ab}^{(t)} \\ &\geq \beta[\mathbf{D} - \mathbf{A}]_{aa}w_{ab}^{(t)} = \beta[\mathbf{L}]_{aa}.\end{aligned}$$

Thus, $G(w, w_{ab}^{(t)}) \geq J_{ab}(w)$ holds.

Replacing $G(w, w_{ab}^{(t)})$ in 14 results in the update rule:

$$\begin{aligned}w_{ab}^{(t+1)} &= w_{ab}^{(t)} - w_{ab}^{(t)}\cdot \\ &\cdot \frac{J'_{ab}(w_{ab})}{\alpha[\mathbf{W}\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}}]_{ab} + \gamma[\mathbf{W}\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}}]_{ab} + \beta[\mathbf{D}\mathbf{W}]_{ab} + \lambda[\mathbf{W}]_{ab}},\end{aligned}$$

$$w_{ab}^{(t+1)} = w_{ab}^{(t)}\frac{[\alpha\mathbf{X_s}\mathbf{H_s}^{\mathrm{T}} + \gamma\mathbf{X_u}\mathbf{H_u}^{\mathrm{T}} + \beta\mathbf{A}\mathbf{W}]_{ab}}{[\alpha\mathbf{W}\mathbf{H_s}\mathbf{H_s}^{\mathrm{T}} + \gamma\mathbf{W}\mathbf{H_u}\mathbf{H_u}^{\mathrm{T}} + \beta\mathbf{D}\mathbf{W} + \lambda\mathbf{W}]_{ab}}.$$

Since G is an auxiliary function, J_{ab} is non-increasing under this update rule. □