Improving NMF clustering by leveraging contextual relationships among words

Mickael Febrissy, Aghiles Salah, Melissa Ailem, Mohamed Nadif

Improving NMF Clustering by Leveraging Contextual Relationships Among Words

Mickael Febrissy[a], Aghiles Salah[b], Melissa Ailem[c], Mohamed Nadif[a]

[a]*Centre Borelli UMR 9010, Université Paris Cité, 75 006 Paris, France*
[b]*Rakuten Institute Technology (France)*
[c]*Lingua Custodia (France)*

Abstract

Non-negative Matrix Factorization (NMF) and its variants have been successfully used for clustering text documents. However, NMF approaches like other models do not explicitly account for the contextual dependencies between words. To remedy this limitation, we draw inspiration from neural word embedding and posit that words that frequently co-occur within the same context (e.g., sentence or document) are likely related to each other in some semantic aspect. We then propose to jointly factorize the document-word and word-word co-occurrence matrices. The decomposition of the latter matrix encourages frequently co-occurring words to have similar latent representations and thereby reflecting the relationships among them. Empirical results, on several real-world datasets, provide strong support for the benefits of our approach. Our main finding is that we can drastically improve the clustering performance of NMF by leveraging the contextual relationships among words explicitly.

1. Introduction

Since the work of Lee and Seung [29, 30], Non-negative Matrix Factorization (NMF) has been receiving more and more attention as a fundamental tool for

analyzing positive data arising in various areas, such as computer vision and text analysis.

Compared to other matrix factorization techniques such as Singular Value Decomposition (SVD), NMF is attractive due to the non-negativity constraints, making it possible to produce latent representations which are highly interpretable and well-suited for some tasks such as clustering, for which we focus on.

Although clustering is not the primary purpose of Non-negative Matrix Factorization (NMF) [29, 30], the latter has received a lot of interest in the clustering community resulting in a new class of clustering algorithms—based on NMF. Furthermore, a series of works [16, 35, 18] established theoretical connections between NMF and k-means, spectral clustering, providing strong support and theoretical foundations for NMF-based clustering. More precisely, NMF is equivalent to a relaxed k-means clustering yielding a soft partitioning.

In this paper, we deal with NMF from a clustering perspective, and we particularly focus on the task of text document clustering, which constitutes one of the most popular and prominent applications of NMF. Clustering text documents is of great interest for several practical reasons such as: automatic summarization and organization of documents, efficient browsing and navigation of huge text corpora, visualization, speed up search engines, etc. In this context, NMF-based algorithms decompose the document-word matrix into a document and word factor matrices containing, respectively, the low dimensional representations—embeddings—of each document and word [55]. It turns out that, the document factor matrix encodes some latent patterns of the original matrix that are well suited to cluster text documents.

Despite all the notable efforts which highlighted the potential of NMF for

clustering text documents [34], this approach still exhibits some limitations, namely it does not explicitly account for the semantic relationships between words. Hence, words having a common meaning—synonyms—or more generally words that are about the same topic are not guaranteed to be mapped in the same direction in the latent space. This is simply due to the fact that words with similar meanings are not necessarily used exactly in the same documents. Consequently, similar embeddings are not guaranteed even for closely related documents using words with similar meanings. Hence, our intuition is that, if we are successful in capturing the semantic relationships among words in an NMF model we can expect document factors which are even better for clustering.

The research question is how to capture and leverage the relationships among words in an NMF model? In this paper, we draw inspiration from neural word embedding and rely on the distributional hypothesis [21], which states that words in similar contexts have similar meanings. The context is a modeling choice that could be data- or problem-specific. For instance, a document or a sentence is a context in which words co-occur. Note that other definitions of "contexts" are possible [32]. An early application of that hypothesis in Matrix Factorization is the *Hyperspace Analogue to Language* (HAL) [36] framework. It employs a word-word co-occurrence matrix whose entries encode the number of times each pair of words has occurred in the same context. Thus, following the distributional hypothesis, we assume that words which frequently co-occur in the same context are likely related to each other in some semantic aspect. We then, propose a new NMF model which jointly decomposes the document-word and word-word co-occurrence matrices into two separate products that share one factor. The intuition behind the decomposition of the latter matrix is to make the representations of

frequent co-occurring words closer to each other in the latent space so as to reflect the relationships among them. We further consider a non-linear transformation of the word co-occurrences, based on the Pointwise Mutual Information (PMI), for effectiveness and efficiency purposes.

In order to infer the factor matrices, we propose a scalable alternating optimization procedure based on a set of multiplicative update rules, similar to original NMF, which guarantees to decrease monotonically our objective function at each iteration, until convergence. We conduct extensive experiments to illustrate the benefits of our model and better characterize the circumstances in which it offers the most significant improvements. Our main finding is that, we can drastically improve the clustering performance of NMF by leveraging explicitly the contextual relationships among words[1].

The outline of the paper is as follows. Section 2 is devoted to present related work. In Section 3, we briefly review the original NMF model and some recent results concerning the skip-gram model with negative sampling—`word2vec`—that we exploit in our work. In section 4, we present our model and derive a scalable iterative algorithm for inference. Section 5 is devoted to numerical experiments. Subsequently, we discuss some possible extensions of our model that we have already investigated (section 6). Finally, we conclude and suggest paths for future research in section 7.

[1] In this paper we use " contextual relationships" and "semantic relationships" interchangeably. The former relationships underlie the latter ones, and our objective is to rely on the words' context to capture the semantic relationships among them

2. Related Works

Our contribution is mainly related to the works on non-negative matrix factorization for clustering. The literature on NMF is very large; for instance the reader can refer to [54]. Below we try to provide a brief review of works that are most closely related to our contribution.

There has been a lot of research on developing new variants of NMF in the direction of clustering. For instance, Ding et al. [17], Yoo and Choi [58] proposed orthogonal NMF, which constrains the document factor matrix to be orthogonal, and highlighted the importance of such constraint for clustering. Ding et al. [18] introduced Semi-NMF and Convex-NMF, enforcing the non-negativity constraint on the document factor matrix, but allowing the original data to have mixed signs [2]. Although both variants have been found to perform somewhat less well than NMF for clustering positive data, they expand the applicable range of NMF models, to data having mixed sings, as well as strengthen their relation to clustering. In [60, 57] the authors proposed Projective NMF (PNMF) with a single latent factor matrix only. PNMF is equivalent to *soft* k-means clustering and performs better than NMF for text document clustering.

Cai et al. [9] developed Graph Regularized NMF (GNMF), which aims to preserve the intrinsic geometry of the data distribution. In GNMF, the data (document) manifold structure is modeled by a nearest neighbor graph, and it is incorporated into original NMF as an additional regularization term in order to force the embeddings of documents, which are connected, to be close to each other. Shang et al. [47] proposed graph dual regularized NMF, which extends GNMF to model both the data manifold and feature manifold simultaneously. In order to reduce the sensitivity of GNMF to the nearest neighbor graph's parameters, the authors

in [53] developed multiple graph regularized NMF where the the data manifold is approximated by a linear combination of several nearest neighbor graphs having different parameters. In the same vein, more robust extensions of NMF, which can handle data points lying in complex manifolds, have been recently proposed [22, 19]. Modeling document manifolds is an orthogonal direction to the present work; here we focus on preserving the contextual relationships between words.

Extensions of NMF to co-clustering have been developed to cluster objects (e.g. documents) and features (e.g. words), simultaneously. These approaches [17, 59, 26, 12, 3], commonly referred to as Non Negative Matrix Tri-Factorization (NMTF), seek a decomposition of the original data matrix into three non-negative factor matrices. Some other works [15, 11, 10, 44] have focused on developing algorithmic extensions of NMF to accommodate different cost functions and different machine learning tasks, or to develop more effective and efficient algorithms to solve the NMF problem [25, 20]. For more details on NMF and its extensions for clustering, please refer to [34].

In order to leverage the relationships among words in NMF, we draw inspiration from neural word embedding. These approaches, seek continuous representations of words that reflect various linguistic regularities between them [4, 39, 37]. To achieve their objective, most neural word embedding methods rely on the distributional hypothesis of Harris [21]. For instance, the recently proposed skip-gram model with negative sampling aims to maximize the dot-product between the vectors of frequently occurring word-context pairs, and minimize it for random word-context pairs. For more details please refer to [37]. What makes these models particularly appealing is their ability to learn word vectors that are good at capturing meaningful semantic and syntactic regularities between words [38]. Similar

to word embedding techniques, the model we propose relies on the distributional hypothesis to capture the semantic relationships between words in NMF. Our preliminary investigation of infusing NMF with contextual relationships among words has appeared recently as a short paper [1]. In the present manuscript, we delve in-depth into this idea and present several new theoretical and empirical results.

Notation. Matrices are denoted with boldface uppercase letters and vectors with boldface lowercase letters. The Frobenius norm is denoted by $\|.\|_F$ and the Hadamard product by $\odot$. The document-word matrix is represented by a matrix $\mathbf{X} = (x_{ij}) \in \mathbb{R}_+^{n \times d}$, its i^{th} row represents the weighted term frequency vector of document $i \in \mathcal{I}$, i.e., $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})^\top$ where $\top$ denotes the transpose. The word co-occurrence matrix is represented by $\mathbf{C} = (c_{jj'}) \in \mathbb{R}_+^{d \times d'}$, following the nomenclature in neural word embedding, row $j \in \mathcal{J}$ corresponds to word w_j, column $j' \in \mathcal{J}'$ denotes context word $w_{j'}$, and each entry $c_{jj'}$ denotes the number of times the word-context pair (w_j,$w_{j'}$) occurred in the same context (e.g., a sentence or a document). The word and context word vocabularies, $\mathcal{J}$ and $\mathcal{J}'$ might be different.

3. Preliminaries

In the context of text data, NMF seeks a decomposition of a document-word matrix $\mathbf{X}$ into two low dimensional factor matrices $\mathbf{Z} = (z_{ik}) \in \mathbb{R}_+^{n \times g}$ and $\mathbf{W} = (w_{jk}) \in \mathbb{R}_+^{d \times g}$, such that $\mathbf{X} \approx \mathbf{Z}\mathbf{W}^\top$. To infer the latent factor matrices, NMF attempts to solve the following optimization problem

$$\arg\min_{\mathbf{Z},\mathbf{W}} \mathcal{D}(\mathbf{X}, \mathbf{Z}\mathbf{W}^\top), \quad s.t.\ \mathbf{Z},\ \mathbf{W} \geq 0. \tag{1}$$

where $\mathcal{D}$ is a cost function that allows us to quantify the quality of the approximation of $\mathbf{X}$ by $\mathbf{Z}\mathbf{W}^\top$; $\mathcal{D}$ can be, for instance, the Frobenius norm or the I-divergence. As

NMF has an inherent clustering property, the document factor matrix $\mathbf{Z}$ is usually considered as a soft cluster membership matrix, where z_{ik} denotes the degree to which document i belongs to cluster k. A partition of the set of documents can then be obtained by assigning each document to the most likely cluster. Notice that, in order to make the solution of (1) unique, $\mathbf{Z}$ is usually normalized to have unit-length column vectors.

The PMI is an information theoretic measure widely used to quantify the association between pairs of outcomes coming from discrete random variables. Formally, the PMI between word w_j and its context word $w_{j'}$ is given by

$$\text{PMI}(w_j, w_{j'}) = \log \frac{p(w_j, w_{j'})}{p(w_j)p(w_{j'})}. \tag{2}$$

Given the word co-occurrence matrix $\mathbf{C}$ defined above, the PMI between w_j and $w_{j'}$ can be empirically estimated as follows

$$\text{PMI}(w_j, w_{j'}) = \log \frac{c_{jj'} \times c_{..}}{c_{j.} \times c_{.j'}}, \tag{3}$$

where $c_{..} = \sum_{j=1}^{d} \sum_{j'=1}^{d'} c_{jj'}$, $c_{j.} = \sum_{j'=1}^{d'} c_{jj'}$ and $c_{.j'} = \sum_{j=1}^{d} c_{jj'}$.

The expected value of the PMI across all the possible events is the Mutual Information (MI) that is positive. A null PMI indicates that the events are independent, negative values of PMI indicate that those events occur less frequently than expected. Therefore a useful variation called Positive PMI (PPMI) is to set all negative PMI values to zero. This transformation has been shown to produce good semantic representations [7].

4. Method

4.1. Formulation

In this section, we describe our model, Semantic-NMF, which jointly performs NMF on the document-word matrix and word-word PPMI matrix, with shared word

factors, to better capture and leverage the semantic relationships among words. Formally the objective function of Semantic-NMF, to be minimized, is given by

$$\mathcal{F}(\mathbf{Z}, \mathbf{W}, \mathbf{Q}) = \underbrace{\mathcal{D}_1(\mathbf{X}, \mathbf{Z}\mathbf{W}^\top)}_{\text{NMF}} + \lambda \underbrace{\mathcal{D}_2(\mathbf{M}, \mathbf{W}\mathbf{Q}^\top)}_{\text{word embedding}}, \tag{4}$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$, are cost functions for measuring the divergence between non-negative matrices, λ is a regularization parameter, and following the nomenclature in neural word embedding, we refer to $\mathbf{Q} \in \mathbb{R}_+^{d' \times g}$ as the context factor matrix. The above objective function can be viewed as regularizing the word factors in NMF beyond usual regularization schemes (e.g., L_2 norm). Note that, both terms in (4) infer low dimensional representations of words. In the NMF term, word factors encode how words are used in documents, while in the word embedding term, word representations encode word co-occurrence patterns. Semantic-NMF seeks to leverage both of the above information, simultaneously. Additionally, whilst $d' = d$ due to $\mathbf{M}$ defined as a word-word PPMI matrix, Semantic-NMF can easily accommodate the definition of $\mathbf{M}$ as a word embedding matrix where $d' \neq d$ (favorably $d' \leq d$). Figure 1 provides a graphical illustration of Semantic-NMF.

4.2. Inference

In this section, we shall investigate the case where both $\mathcal{D}_1$ and $\mathcal{D}_2$ are the square of the Frobenius norm, and derive an iterative optimization procedure to

infer the latent factor matrices. In this case, (4) takes the following form:

$$\begin{aligned}\mathcal{F}(\mathbf{Z},\mathbf{W},\mathbf{Q}) =& \frac{1}{2}||\mathbf{X}-\mathbf{Z}\mathbf{W}^\top||_F^2 + \frac{\lambda}{2}||\mathbf{M}-\mathbf{W}\mathbf{Q}^\top||_F^2 \\ =& \frac{1}{2}\operatorname{Tr}\left((\mathbf{X}-\mathbf{Z}\mathbf{W}^\top)(\mathbf{X}-\mathbf{Z}\mathbf{W}^\top)^\top\right) \\ &+ \frac{\lambda}{2}\operatorname{Tr}\left((\mathbf{M}-\mathbf{W}\mathbf{Q}^\top)(\mathbf{M}-\mathbf{W}\mathbf{Q}^\top)^\top\right) \\ =& \frac{1}{2}\operatorname{Tr}\left(\mathbf{X}\mathbf{X}^\top - 2\mathbf{X}\mathbf{W}\mathbf{Z}^\top + \mathbf{Z}\mathbf{W}^\top\mathbf{W}\mathbf{Z}^\top\right) \\ &+ \frac{\lambda}{2}\operatorname{Tr}\left(\mathbf{M}\mathbf{M}^\top - 2\mathbf{M}\mathbf{Q}\mathbf{W}^\top + \mathbf{W}\mathbf{Q}^\top\mathbf{Q}\mathbf{W}^\top\right). \end{aligned} \tag{5}$$

In the following, we derive a set of multiplicative update rules in order to minimize $\mathcal{F}$ under the constraints of positivity of $\mathbf{Z}$, $\mathbf{W}$ and $\mathbf{Q}$. Let $\boldsymbol{\alpha} \in \mathbb{R}^{n\times g}$, $\boldsymbol{\beta} \in \mathbb{R}^{d\times g}$, $\boldsymbol{\gamma} \in \mathbb{R}^{d'\times g}$ be the Lagrange multipliers for the constraints, the Lagrange function $\mathcal{L}(\mathbf{Z},\mathbf{W},\mathbf{Q},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) = \mathcal{L}$ is given by

$$\mathcal{L} = \mathcal{F}(\mathbf{Z},\mathbf{W},\mathbf{Q}) + \operatorname{Tr}(\boldsymbol{\alpha}\mathbf{Z}^\top) + \operatorname{Tr}(\boldsymbol{\beta}\mathbf{W}^\top) + \operatorname{Tr}(\boldsymbol{\gamma}\mathbf{Q}^\top).$$

The derivatives of $\mathcal{L}$ with respect to $\mathbf{Z}$, $\mathbf{W}$ and $\mathbf{Q}$ are

$$\nabla_{\mathbf{Z}}\mathcal{L} = -\mathbf{X}\mathbf{W} + \mathbf{Z}\mathbf{W}^\top\mathbf{W} + \boldsymbol{\alpha}, \tag{6a}$$

$$\nabla_{\mathbf{W}}\mathcal{L} = -(\mathbf{X}^\top\mathbf{Z} + \lambda\mathbf{M}\mathbf{Q}) + \mathbf{W}(\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{Q}^\top\mathbf{Q}) + \boldsymbol{\beta}, \tag{6b}$$

$$\nabla_{\mathbf{Q}}\mathcal{L} = -\lambda\mathbf{M}^\top\mathbf{W} + \lambda\mathbf{Q}\mathbf{W}^\top\mathbf{W} + \boldsymbol{\gamma}. \tag{6c}$$

Setting these gradients to zero and making use of the Kuhn-Tucker conditions

$$\begin{cases} \boldsymbol{\alpha}\odot\mathbf{Z} &= 0 \\ \boldsymbol{\beta}\odot\mathbf{W} &= 0 \\ \boldsymbol{\gamma}\odot\mathbf{Q} &= 0 \end{cases}$$

we obtain the following stationary equations:

$$-(\mathbf{X}\mathbf{W})\odot\mathbf{Z} + (\mathbf{Z}\mathbf{W}^\top\mathbf{W})\odot\mathbf{Z} = 0,$$

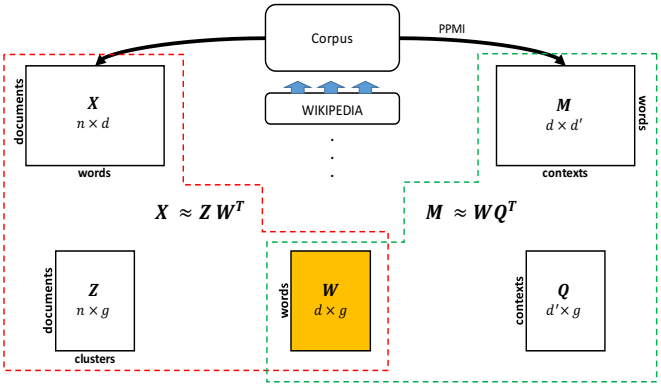


Figure 1: Illustrative scheme of the proposed Semantic-NMF model. $\mathbf{X} \approx \mathbf{Z}\mathbf{W}^\top$ and $\mathbf{M} \approx \mathbf{W}\mathbf{Q}^\top$

$$-(\mathbf{X}^\top\mathbf{Z} + \lambda\mathbf{M}\mathbf{Q}) \odot \mathbf{W} + \mathbf{W}(\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{Q}^\top\mathbf{Q}) \odot \mathbf{W} = 0,$$

$$-(\mathbf{M}^\top\mathbf{W}) \odot \mathbf{Q} + (\mathbf{Q}\mathbf{W}^\top\mathbf{W}) \odot \mathbf{Q} = 0.$$

Based on the above equations we derive the following multiplicative update rules

$$\mathbf{Z} \leftarrow \mathbf{Z} \odot \frac{\mathbf{X}\mathbf{W}}{\mathbf{Z}\mathbf{W}^\top\mathbf{W}}, \tag{7a}$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{(\mathbf{X}^\top\mathbf{Z} + \lambda\mathbf{M}\mathbf{Q})}{\mathbf{W}(\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{Q}^\top\mathbf{Q})}, \tag{7b}$$

$$\mathbf{Q} \leftarrow \mathbf{Q} \odot \frac{\mathbf{M}^\top\mathbf{W}}{\mathbf{Q}\mathbf{W}^\top\mathbf{W}}. \tag{7c}$$

These update rules are analogous to those of NMF [30]. The difference is in how we update the word factors in Semantic-NMF. In the latter, the update of $\mathbf{W}$ depends on two sources of data (i) the document-word matrix and (ii) the PPMI co-occurence matrix $\mathbf{M}$.

Theorem 1. *The objective function of Semantic-NMF is non-increasing under the update formulas (7a), (7b) and (7c).*

Proof. Equations (7a) and (7c) are similar to those of NMF [30], therefore based on the proof of [30] the objective function of Semantic-NMF is non-increasing under these two equations. Hence, we only need to demonstrate that $\mathcal{F}$ is non-increasing under the update rule (7b), given $\mathbf{Z}$ and $\mathbf{Q}$. To this end, we follow a similar approach to the one described in [30], which is inspired by the Expectation-Maximization (EM) algorithm [13] and consists in using an auxiliary function.

Definition. $\mathcal{G}(w, w')$ *is an auxiliary function for* $\mathcal{F}(w)$ *if the following conditions are satisfied* $\mathcal{G}(w, w') \geq \mathcal{F}(w)$ *and* $\mathcal{G}(w, w) = \mathcal{F}(w)$.

A key point to the auxiliary function is described by the following lemma.

Lemma 1. If $\mathcal{G}$ *is an auxiliary function for* $\mathcal{F}$*, then* $\mathcal{F}$ *is non-increasing under the update*

$$w^{(t+1)} = \arg\min_{w} \mathcal{G}(w, w^{(t)}). \tag{8}$$

Proof.

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

Now we will make use of an appropriate auxiliary function to demonstrate that our objective function $\mathcal{F}$ is non-increasing under the update rule (7b). Let w_{jk} denote any element in $\mathbf{W}$, and let $\widetilde{\mathcal{F}}(w_{jk})$ denote the part of $\mathcal{F}$ containing w_{jk}. As the update (7b) is element-wise, it is sufficient to show that $\widetilde{\mathcal{F}}$ is non-increasing under the update of w_{jk} based on equation (7b). The first and second partial derivatives of $\widetilde{\mathcal{F}}$ noted $\widetilde{\mathcal{F}}'$, $\widetilde{\mathcal{F}}''$ are respectively given by

$$\widetilde{\mathcal{F}}'(w_{jk}) = \left(-\mathbf{X}^\top \mathbf{Z} - \lambda \mathbf{M}\mathbf{Q} + \mathbf{W}(\mathbf{Z}^\top \mathbf{Z} + \lambda \mathbf{Q}^\top \mathbf{Q})\right)_{jk},$$

$$\widetilde{\mathcal{F}}''(w_{jk}) = \left(\mathbf{Z}^\top \mathbf{Z} + \lambda \mathbf{Q}^\top \mathbf{Q}\right)_{kk}.$$

The following lemma yields an auxiliary function for $\widetilde{\mathcal{F}}$.

Lemma 2. The function $\mathcal{G}$ defined as follows

$$\mathcal{G}(w_{jk}, w_{jk}^{(t)}) = \widetilde{\mathcal{F}}(w_{jk}^{(t)}) + \widetilde{\mathcal{F}}'(w_{jk}^{(t)})(w_{jk} - w_{jk}^{(t)}) + \frac{\left(\mathbf{W}\left(\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{Q}^\top\mathbf{Q}\right)\right)_{jk}}{2w_{jk}^{(t)}}(w_{jk} - w_{jk}^{(t)})^2 \tag{9}$$

is an auxiliary function for $\widetilde{\mathcal{F}}$.

Proof. Based on Lemma 2 it straightforward to verify that $\mathcal{G}(w_{jk}, w_{jk}) = \widetilde{\mathcal{F}}(w_{jk})$. We will now show that $\mathcal{G}(w_{jk}, w_{jk}^{(t)}) \geq \widetilde{\mathcal{F}}(w_{jk})$, by making use of the second order Taylor expansion of $\widetilde{\mathcal{F}}$ about $w_{jk}^{(t)}$ given by

$$\widetilde{\mathcal{F}}(w_{jk}) = \widetilde{\mathcal{F}}(w_{jk}^{(t)}) + \widetilde{\mathcal{F}}'(w_{jk}^{(t)})(w_{jk} - w_{jk}^{(t)}) + \frac{\left(\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{Q}^\top\mathbf{Q}\right)_{kk}}{2}(w_{jk} - w_{jk}^{(t)})^2. \tag{10}$$

Since

$$\left(\mathbf{W}\mathbf{Z}^\top\mathbf{Z}\right)_{jk} = \sum_{k'=1}^{g} w_{jk'}^{(t)}\left(\mathbf{Z}^\top\mathbf{Z}\right)_{k'k} \geq w_{jk}^{(t)}\left(\mathbf{Z}^\top\mathbf{Z}\right)_{kk}$$

and similarly

$$\left(\mathbf{W}\mathbf{Q}^\top\mathbf{Q}\right)_{jk} \geq w_{jk}^{(t)}\left(\mathbf{Q}^\top\mathbf{Q}\right)_{kk},$$

we have $\frac{\left(\mathbf{W}(\mathbf{Z}^\top\mathbf{Z}+\lambda\mathbf{Q}^\top\mathbf{Q})\right)_{jk}}{w_{jk}^{(t)}} \geq \left(\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{Q}^\top\mathbf{Q}\right)_{kk}$. Thereby, from (9) and (10), $\mathcal{G}(w_{jk}, w_{jk}^{(t)}) \geq \widetilde{\mathcal{F}}(w_{jk})$ holds. □

Thus, to prove Theorem 1 it is sufficient to show that equation (7b) for w_{jk} satisfies *Lemma 1* where the auxiliary function $\mathcal{G}$ is given by *Lemma 2*. Substituting equation (9) to $\mathcal{G}(w_{jk}, w_{jk}^{(t)})$ in *Lemma 1* leads to solve $\frac{\partial\mathcal{G}(w_{jk}, w_{jk}^{(t)})}{\partial w_{jk}} = 0$ or,

$$\widetilde{\mathcal{F}}'(w_{jk}^{(t)}) + \frac{\left(\mathbf{W}(\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{Q}^\top\mathbf{Q})\right)_{jk}}{2w_{jk}^{(t)}}(2w_{jk} - 2w_{jk}^{(t)}) = 0.$$

Then $w_{jk}^{(t+1)} = \arg\min_w \mathcal{G}(w_{jk}, w_{jk}^{(t)})$ leads to

$$
\begin{aligned}
w_{jk}^{(t+1)} &= -w_{jk}^{(t)} \frac{\widetilde{\mathcal{F}}'(w_{jk}^{(t)})}{\left(\mathbf{W}\left(\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{Q}^\top\mathbf{Q}\right)\right)_{jk}} + w_{jk}^{(t)} \\
&= w_{jk}^{(t)} \frac{(\mathbf{X}^\top\mathbf{Z} + \lambda\mathbf{M}\mathbf{Q})_{jk}}{\left(\mathbf{W}(\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{Q}^\top\mathbf{Q})\right)_{jk}}.
\end{aligned}
$$

It follows from the latter result and *Lemma 1* that $\widetilde{\mathcal{F}}$ is non-increasing under the update of w_{jk} in equation (7b), $\forall j, k$. Given that (7b) is element-wise, the objective function of Semantic-NMF is non-increasing under the update rule (7b). ■

Thereby, based on Theorem 1, the fact that (7a), (7b) and (7c) satisfy the KKT conditions at convergence and $\mathcal{F}$ is bounded from below by 0, iteratively alternating the application of (7a), (7b) and (7c) will monotonically decrease criterion (5) and converge to a locally optimal solution. Our optimization procedure is depicted in Algorithm 1.

Algorithm 1 Semantic-NMF (`SNMF`).

Input: $\mathbf{X}$, $\mathbf{M}$, λ and g the dimension of the latent factors.
Output: $\mathbf{Z}$, $\mathbf{W}$ and $\mathbf{Q}$.
1. Initialization: $\mathbf{Z} \leftarrow \mathbf{Z}^{(0)}$; $\mathbf{W} \leftarrow \mathbf{W}^{(0)}$ and $\mathbf{Q} \leftarrow \mathbf{Q}^{(0)}$;
repeat
  2. $\mathbf{Z} \leftarrow \mathbf{Z} \odot \frac{\mathbf{X}\mathbf{W}}{\mathbf{Z}\mathbf{W}^\top\mathbf{W}}$;
  3. $\mathbf{W} \leftarrow \mathbf{W} \odot \frac{(\mathbf{X}^\top\mathbf{Z} + \lambda\mathbf{M}\mathbf{Q})}{\mathbf{W}(\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{Q}^\top\mathbf{Q})}$;
  4. $\mathbf{Q} \leftarrow \mathbf{Q} \odot \frac{\mathbf{M}^\top\mathbf{W}}{\mathbf{Q}\mathbf{W}^\top\mathbf{W}}$;
until convergence
5. Normalize $\mathbf{Z}$ so as it has unit-length column vectors.

4.3. Computational Complexity Analysis

The following Proposition shows that the computational complexity of the `SNMF` algorithm scales linearly with the number of non-zero entries in the document-

word and PPMI matrices. In practice $\mathbf{X}$ and $\mathbf{M}$ are very sparse, i.e., $nz_X \ll n \times d$ and $nz_M \ll d \times d$. Furthermore, multiplicative update rules (7a), (7a) and (7c) are parallelizable across documents and words, thereby Semantic-NMF can easily scale to large datasets.

Proposition 1. *Let nz_X and nz_M denote respectively the number of non-zero entries in $\mathbf{X}$ and $\mathbf{M}$, and let it be the number of iterations. The computational complexity of Semantic-NMF is given in $O(it \cdot g \cdot (nz_X + nz_M) + it \cdot g^2 \cdot (n+d))$.*

Proof. The computational bottleneck of `SNMF` is with the multiplicative update formulas (7a), (7b) and (7c). Equations (7a) and (7c) are similar to those of NMF, and their respective complexities are $O(nz_X \cdot g + (n+d) \cdot g^2)$ and $O(nz_M \cdot g + d \cdot g^2)$. The number of operation in (7b), including multiplications, additions and divisions, is $g(2nz_X + 3nz_M + 3d + g(4d + 2n + 1))$, where we used $d' = d$. The complexity of (7b) is thereby given in $O(g \cdot (nz_X + nz_M) + (n+d) \cdot g^2)$. Therefore, the total computational complexity of Semantic-NMF is

$$O(it \cdot g \cdot (nz_X + nz_M) + it \cdot g^2 \cdot (n+d)).\blacksquare$$

5. Experimental study

Our objective is to investigate the effect of the contextual relationships between words on NMF models. To this end, we conduct extensive experiments in which we benchmark our model, Semantic-NMF (`SNMF`), against several state-of-the-art algorithms (including NMF models and clustering algorithms) on several real-world datasets. Furthermore, we also challenge the choice of the PPMI for $\mathbf{M}$ by considering another transformation arising from the word-word co-occurrence

matrix, namely the Global Vectors for Word Representation (GloVe) [43]. Note that, the Hellinger PCA (HPCA) [27] was also tested but did not demonstrated good enough performances to be considered in our proposal.

5.1. Datasets

We use six popular benchmark datasets, described in Table 1, namely **CSTR** [33], **CLASSIC4**[2], **RCV1** containing the four largest classes of the Reuters corpus[3], the **SPORTS** dataset (from the CLUTO toolkit [24]) containing documents relating to seven different sports, the 20-newsgroups dataset **NG20**[3], and the **NG5** dataset consisting of five classes[4] of NG20. These datasets are carefully selected so as to represent various particular challenging situations: different numbers of clusters, different sizes, different degrees of cluster overlap and different degrees of cluster balance (the *Balance* coefficient being the ratio of the minimum cluster size to the maximum cluster size). For each dataset, we apply the TF-IDF weighting scheme and normalize each document to unit L_2 norm so as to remove the biases induced by the length of documents.

Table 1: Description of Datasets, # denotes the cardinality.

Datasets	Characteristics				
	#Documents	#Words	#Clusters	nz_X (%)	Balance
CSTR	475	1000	4	3.40	0.399
CLASSIC4	7095	5896	4	0.59	0.323
RCV1	6387	16921	4	0.25	0.080
NG5	4905	10167	5	0.92	0.943
SPORTS	8580	14870	7	0.86	0.0358
NG20	18846	14390	20	0.59	0.628

[2] http://www.dataminingresearch.com/

[3] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

[4] rec.sport.baseball, soc.religion.christian, talk.politics.mideast, sci.electronics and sci.med

5.2. Competing methods

Without the word embedding term in (5), when $\lambda = 0$, the proposed SNMF degenerates to the original NMF (NMF) [55]. Hence, we can achieve our objective of studying the effects of the word relationships on NMF, most effectively by comparing SNMF to NMF. Moreover, in order to show that leveraging the contextual relationships among words in NMF is effective for text document clustering, we also consider three strong NMF variants, namely orthogonal NMF (ONMF) [58], Projective NMF (PNMF) [60] and graph regularized NMF (GNMF) [9]. All the above models have been found to perform very well and better than several other approaches in terms of text document clustering. A Deep-Learning algorithm, namely Deep Clustering Network (DCN) [56] is also considered in our comparison; it outperforms several clustering (k-means, Spectral Clustering), NMF based method such as (LCCF) [8] and Deep Learning algorithms (e.g. SAE [52]). The Spherical k-means algorithm Skmeans [14], which to this day, remains popular for the task document clustering is also included rather than k-means that is not suitable for sparse data.

5.3. Evaluation metrics

We retain two widely used measures to assess the quality of clustering, namely the Normalized Mutual Information (NMI) [50] and the Adjusted Rand Index (ARI) [23]. Intuitively, NMI quantifies how much the estimated clustering is informative about the true clustering, while the ARI measures the degree of agreement between an estimated clustering and a reference clustering; both NMI and ARI are equal to 1 if the resulting clustering is identical to the true one.

5.4. Settings

For each dataset, g is the true number of clusters. To produce a fair comparison, the same initialization (namely Skmeans) was used across the NMF-like algorithms. Similar settings to the ones used to in [43] are employed for producing the GloVe embeddings; note that any other type of *word-embedding* can be used for the matrix $\mathbf{M}$. Therefore, the GloVe embeddings dimension (in our case d') was set to 100, x_{max} to 100, α to 3/4. A stochastic gradient descent algorithm with a learning rate of 0.15 was used to train the model. Subsequently, all negative entries in the GloVe embeddings are set to zero. In the following, this transformation is referred to as PGLOVE. The setting of the regularization parameter λ is achieved empirically and established w.r.t. the PPMI and PGLOVE matrices.

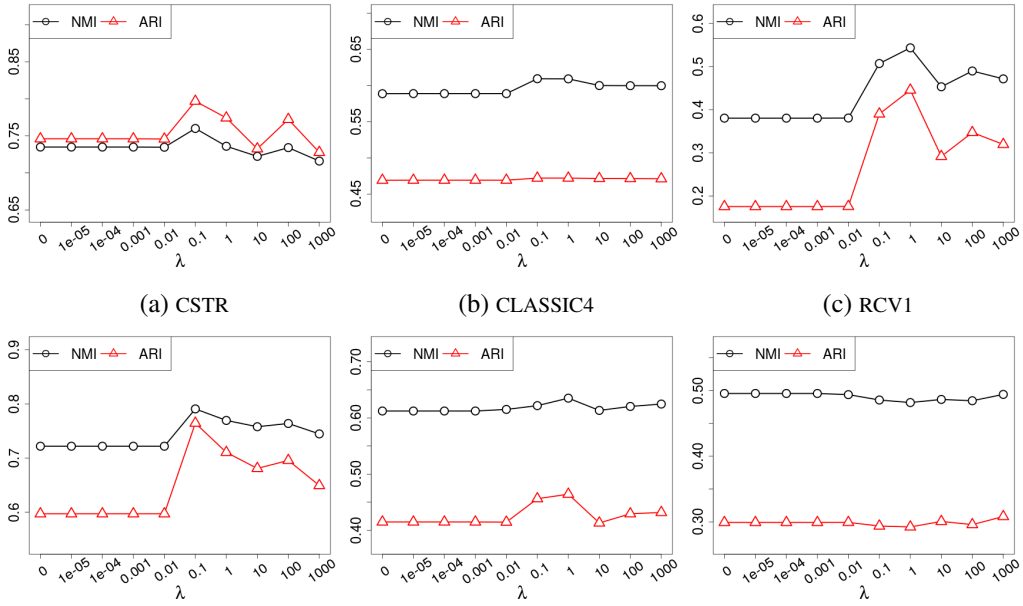
5.5. Empirical results

Below we comment on the results of our experiments and answer several questions related to our proposal.

What is the impact of the regularization parameter on the performances of SNMF?

Figure 2 and 3 display the behaviors of `SNMF` w.r.t. the PGLOVE and PPMI matrices respectively. The results are shown in terms of NMI and ARI scores for several values of λ going from 0 to 10^3. In the case PGLOVE (see Figure 2), the variations of the NMI and ARI scores are unfortunately inconsistent across the range of λ values (see CSTR, RCV1, NG5, SPORTS) making the setting of λ quite difficult and unreliable. However, a good trade off would be $\lambda = 0.1$. On the other hand, using the PPMI (see Figure 2), the variations of the NMI and ARI scores are consistent and linear once a jump is observed. In this case, setting λ is much trivial and reliable and we recommend to set λ to 0.1 since we observe good performance

scores even for higher values of λ on all the datasets. For these reasons, using the PPMI appears as safer alternative.

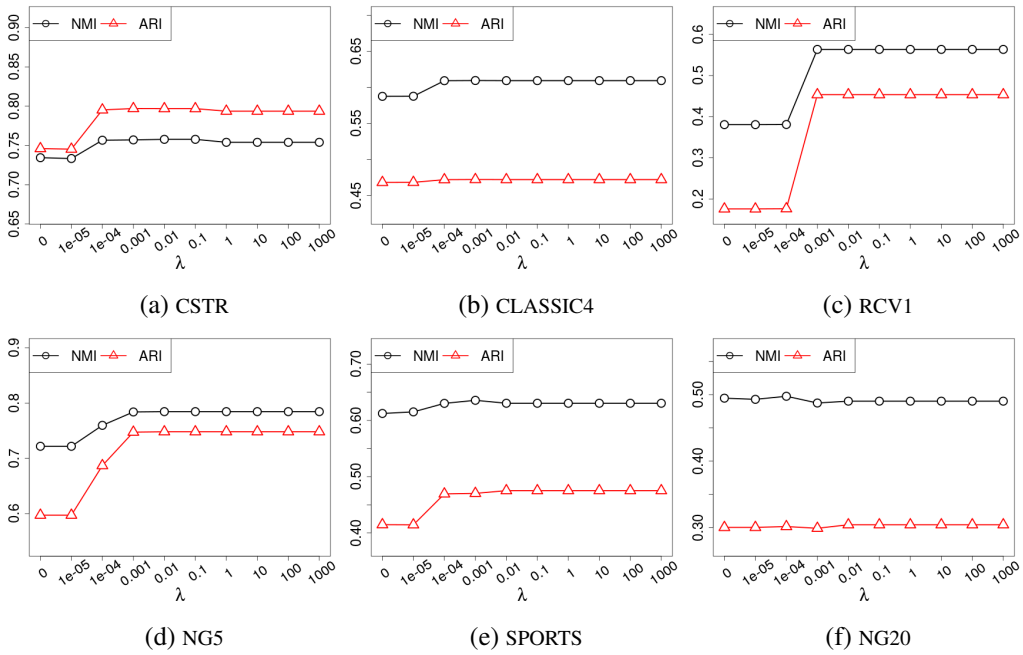


(a) CSTR (b) CLASSIC4 (c) RCV1

(d) NG5 (e) SPORTS (f) NG20

Figure 2: Impact of the regularization parameter λ (PGLOVE).

Table 2, summarizes the results of the different methods in terms of NMI and ARI, over all datasets. All the scores are averages considering the 10 best solutions (in terms of criterion) among a set of fifty different trials. As this table clearly shows, both versions of our model SNMF outperform the other competing methods by an important margin, in most cases. Recalling that SNMF corresponds to NMF with an extra term encoding word co-occurrences. We can therefore attribute the improvement of SNMF upon the performance of NMF to the additional factorization of the PGLOVE or PPMI matrix. In addition, between our two versions (PGLOVE, PPMI), using the PPMI appears to offer better performance overall and will be the version considered in the rest of the paper.



(a) CSTR (b) CLASSIC4 (c) RCV1

(d) NG5 (e) SPORTS (f) NG20

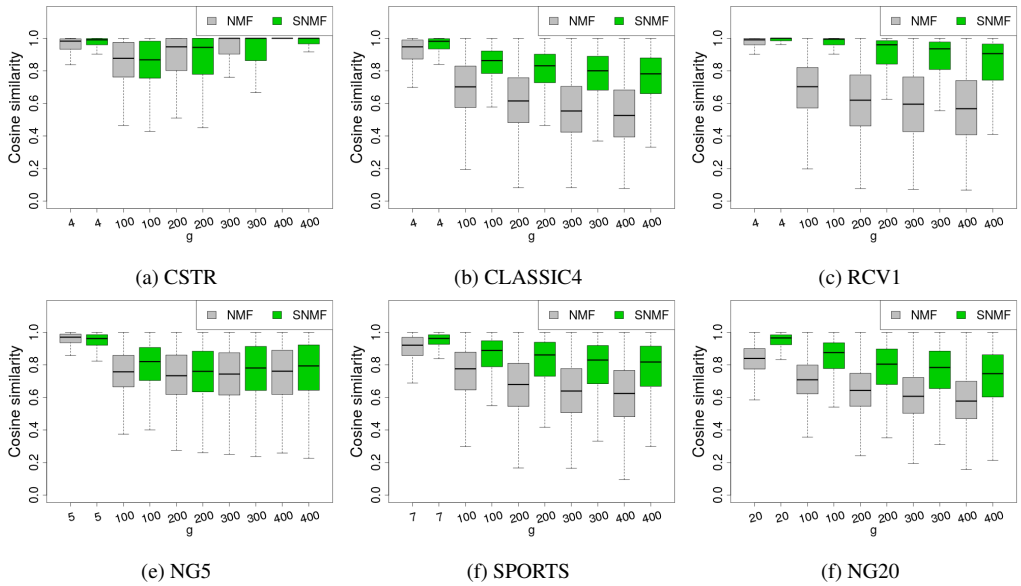
Figure 3: Impact of the regularization parameter λ (PPMI).

Table 2: Mean $\pm$ SD of NMI and ARI over different datasets.

Datasets	Metrics	Skmeans	NMF	ONMF	PNMF	GNMF	**DCN**	**SNMF (PGLOVE)**	SNMF (PPMI)
CSTR	NMI	0.76±0.00	0.73±0.04	0.65±0.00	0.72±0.04	0.69±0.00	0.63±0.024	0.76±0.00	0.76±0.01
	ARI	0.80±0.00	0.75±0.10	0.60±0.03	0.73±0.09	0.75±0.02	0.53±0.03	0.80±0.00	0.80±0.01
CLASSIC4	NMI	0.60±0.00	0.59±0.00	0.49±0.02	0.51±0.00	0.62±0.00	0.57±0.01	0.61±0.02	0.61±0.03
	ARI	0.47±0.00	0.47±0.00	0.41±0.01	0.42±0.00	0.45±0.00	0.42±0.01	0.47±0.00	0.47±0.00
RCV1	NMI	0.38±0.00	0.38±0.00	0.35±0.00	0.36± 0.00	0.34±0.00	0.34±0.00	0.51±0.08	0.56±0.00
	ARI	0.18±0.00	0.18±0.00	0.14±0.00	0.16±0.00	0.12±0.00	0.12±0.00	0.39±0.15	0.45±0.00
NG5	NMI	0.72±0.02	0.72±0.02	0.52±0.01	0.69± 0.00	0.58±0.04	0.62±0.02	0.79±0.00	0.78±0.00
	ARI	0.60±0.01	0.60±0.01	0.29±0.00	0.54±0.00	0.50±0.07	0.47±0.02	0.76±0.00	0.75±0.01
SPORTS	NMI	0.62±0.02	0.61±0.03	0.55±0.02	0.56±0.00	0.55±0.00	0.59±0.01	0.62±0.05	0.63±0.04
	ARI	0.40±0.04	0.41±0.04	0.28±0.01	0.28±0.00	0.28±0.00	0.37±0.03	0.46±0.07	0.48±0.05
NG20	NMI	0.49±0.02	0.49±0.02	0.38±0.01	0.43±0.03	0.00±0.00	0.43±0.01	0.49±0.02	0.49±0.02
	ARI	0.30±0.02	0.30±0.02	0.20±0.00	0.22±0.02	0.00±0.00	0.17±0.01	0.29±0.01	0.33±0.03

To gain further insights into the performances of SNMF and characterize the circumstances in which it provides the most significant improvements, we investigate several research questions below.

What happens with document embeddings? Figure 4 shows the distribution of



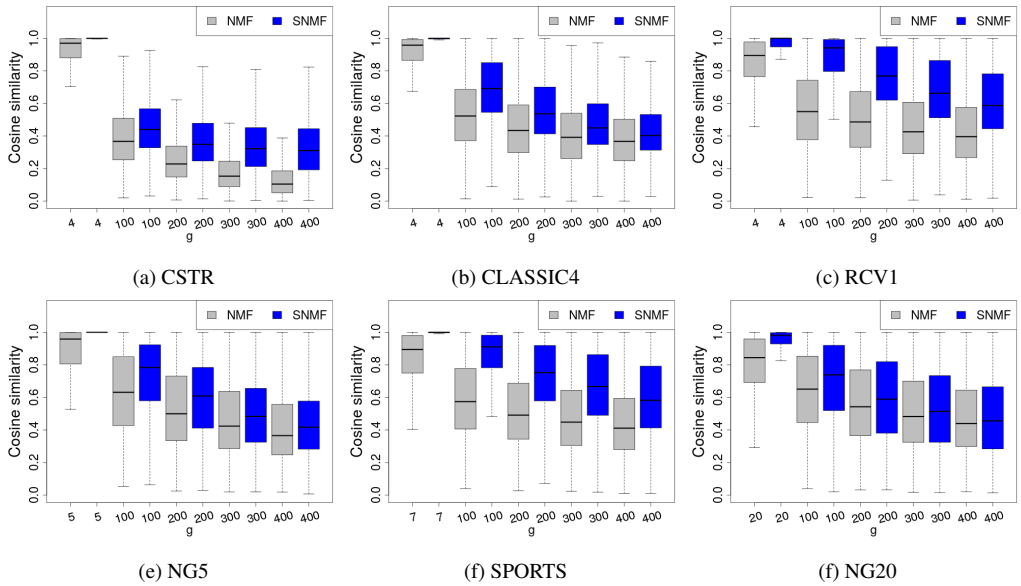
(a) CSTR (b) CLASSIC4 (c) RCV1

(e) NG5 (f) SPORTS (f) NG20

Figure 4: Distribution of cosine similarities between pairs of documents belonging to the same class, computed using the documents' embeddings obtained by NMF and SNMF. The documents of the same class tend to have more similar embeddings under SNMF than NMF.

the cosine similarities between pairs of documents belonging to the same "true" class, computed using the document embeddings produced by NMF (grey boxplots) and SNMF (green boxplots). We observe that documents from the same class (topic) tend to have more similar embeddings under SNMF than NMF. This provides empirical evidence that accounting for the semantic relationships among words yields document factors that encode the clustering structure even better.

Is SNMF actually capturing the semantic relationships between words? Based on the document-word matrix, we select the top thirty words of each true class. In Figure 5, we report the distribution of the cosine similarities between pairs of top words of the same class, computed using the word vectors inferred by NMF (grey boxplots) and SNMF (blue boxplots). Because the cosine similarity is likely



(a) CSTR (b) CLASSIC4 (c) RCV1

(e) NG5 (f) SPORTS (f) NG20

Figure 5: Distribution of cosine similarities between the top 30 words characterizing each document class, computed using the words' embeddings obtained by NMF and SNMF. The top words of the same class tend to have more similar embeddings under SNMF than NMF.

to be high between low dimensional vectors (e.g. $g = 4$), we vary g from the real number of clusters to 400 for each dataset. As this figure shows clearly, the top words of each class have more similar embeddings under SNMF than NMF. This confirms that SNMF does a better job than NMF in capturing semantics, by making the representations of words which are about the same topic (class) closer to each other in the latent space.

We also investigate the effect of the contextual relationships between words by comparing SNMF with NMF in terms of cluster interpretability. To human subjects, interpretability is closely related to coherence [42], i.e., how much the top words of each cluster are "associated" with each other. For each cluster k, we select its top 30 words based on the kth column of $\mathbf{W}$. We use the PMI, which is highly correlated with human judgments [41, 45], to measure the degree of association

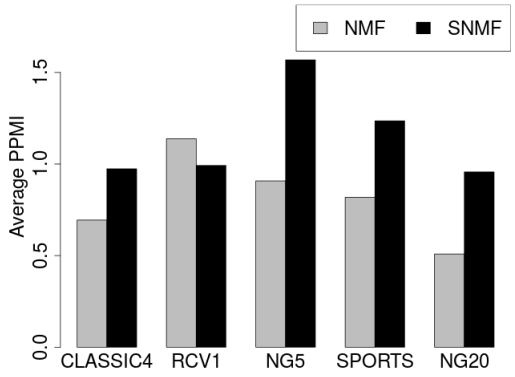


Figure 6: Cluster interpretability: Average PMI score. Semantic-NMF leads more interpretable document clusters than NMF.

between top word pairs. For each cluster we average the PMI's among its top words, and for a model we average PMI across clusters. Because SNMF already exploits the PMI estimated from word co-occurrences in each dataset, we propose to use an external corpus to estimate the PMI in this experiment. Following Newman et al. [41], we use the whole English WIKIPEDIA corpus, that consists of approximately 4 millions of documents and 2 billions of words. Hence, $p(w_j)$ is the probability that word w_j occurs in WIKIPEDIA, and $p(w_j, w_{j'})$ is the probability that words w_j and $w_{j'}$ co-occur in a 5-word window in any WIKIPEDIA document.

Figure 6 shows the average PMI obtained by SNMF and NMF, over the different datasets; it is clear that SNMF successes in capturing more semantics and inferring more interpretable clusters than NMF.

5.6. Cluster Ensembles

Throughout our experiments, Skmeans has proved to be a good initialization for gaining better NMF solutions with text data. However, we noticed that random starting values could sometimes lead to better solutions. Table 3 reports results

of SNMF initialized with Skmeans and randomly. We can see that with RCV1, SNMF (Random) provides better partitions than SNMF (Skmeans). While this improvement only appears with one dataset (other encountering losses, see CLASSIC4 and NG5), we tried to benefit from that infrequent/inconsistent behavior by using the SNMF (Random) solutions along side those obtained with a Skmeans initialization. Furthermore, in unsupervised learning, selecting an unique partition among the set of trials has also been a reluctant problem which to this day remains unclearly addressed. In the case of NMF, the objective function is not defined as a clustering problem, therefore, it often happens that the selection of the best run (criterion-wise) among several does not account for getting the best clustering. However the best clustering could be among a set of lead solutions (for instance the 10 first ones). In other words, a consensus approach will also help us to overcome this issue.

In machine learning, the idea of utilizing multiple sources of data partitions firstly occurred with multi-learner systems where the output of several classifier algorithms where used together in order to improve the accuracy and robustness of a classification or regression, for which strong performances were acknowledged [50, 48, 49]. At this stage, very few approaches have worked toward applying a similar concept to unsupervised learning algorithms. In this sense, we denote the work of [6] who tried to combine several clustering partitions according to the combination of the cluster centers. In the early 2000, [50] were the first to consider an idea of combining several data partitions however, without accessing any original sources of information (features) or led computed centers. This approach is referred to as $cluster\ ensembles$. At the time, their idea was motivated by the possibility of taking advantage of existing information such as a prior clustering partitions or

an expert categorization (all regrouped under the terms Knowledge Reuse), which may still be relevant or substantial for a user to consider in a new analysis on the same objects, whether or not the data associated with these objects may also be different than the ones used to define the prior partitions. Another motivation was *Distributed computing*, referring to analyzing different sources of data (which might be complicated to merge together for instance for privacy reasons) stored in different locations. In our concept, we will use *cluster ensembles* to improve the quality of the final partition (as opposed to selecting a unique one) and therefore extract all the possibilities offered by the miscellaneous best solutions created by NMF.

In [50], the authors introduced three consensus methods that can produce a partition. All of them consider the consensus problem on a hypergraph representation $\mathbf{H}$ of the set of partitions $\mathbf{H}^r$. More specifically, each partition $\mathbf{H}^r$ equals a binary classification matrix (with objects in rows and clusters in columns) where the concatenation of all the set defines the hypergraph $\mathbf{H}$.

- The first one is called Cluster-based Similarity Partitioning Algorithm (**CSPA**) and consists in performing a clustering on the hypergraph according to a similarity measure.
- The second is referred to as HyperGraph Partitioning Algorithm (**HGPA**) and aims at optimizing a minimum cut objective.
- The third one is called Meta-CLustering Algorithm (**MCLA**) and looks forward to identifying and constructing groups of clusters.

Furthermore, in [50] the authors proposed an objective function to characterize the *cluster ensembles* problem and therefore allowing a selection of the best consensus

algorithm among the three to deliver its ensemble partition. Let $\Lambda = \{\lambda^{(q)} | q \in \{1, \ldots, r\}\}$ be a given set of r partitions $\lambda^{(q)}$ represented as labels vectors. The ensemble criterion denoted as $\lambda^{(k-opt)}$ is called the optimal combine clustering and aims at maximizing the Average Normalized Mutual Information (ANMI). It is defined as follows:

$$\lambda^{(k-opt)} = \arg\max_{\widetilde{\lambda}} \sum_{q=1}^{r} \text{NMI}(\widetilde{\lambda}, \lambda^{(q)}). \tag{11}$$

The ANMI is simply the average of the normalized mutual information of a labels vector $\widetilde{\lambda}$ with all labels vectors $\lambda^{(q)}$ in Λ:

$$\text{ANMI}(\Lambda, \widetilde{\lambda}) = \frac{1}{r} \sum_{q=1}^{r} \text{NMI}(\widetilde{\lambda}, \lambda^{(q)}). \tag{12}$$

To cast with cases where the vector labels $\lambda^{(q)}$ have missing values, the authors have proposed a generalized expression of (11) not substantially different that viewers can refer to in the original paper [50].

Following the cluster-based consensus approach which implies a similarity-based clustering algorithm, we decided to make use of a model-based clustering to go and try to obtain a better final partition than the one delivered by $cluster\ ensembles$. In [51], the authors have used the Multinomial mixture approach to propose a consensus function. In model-based clustering, it is assumed that the data are generated by a mixture of underlying probability distributions, where each component k of the mixture represents a cluster.

Let $\Lambda \in \mathbb{N}_0^{n \times r}$ be the data matrix of labels vectors from the top r solutions. Our data being categorical, we used a Multinomial Mixture Model (MMM) in order to partition the elements λ_i. Categorical data being a generalization of binary data; assuming a perfect scenario where there is no partition with an empty cluster, a

disjunctive matrix $\mathbf{M} \in \{0,1\}^{n \times rg}$ is usually used instead of Λ with value $m_{iq}^{(h)}$ where $h \in 1, \ldots, g$ is a cluster label. Therefore, the data value $m_{iq}^{(h)}$ are assumed to be generated from a Multinomial distribution of parameter $\mathcal{M}(m_{iq}^{(h)}; \alpha_{kq}^{(h)})$ where $\alpha_{kq}^{(h)}$ is the probability that an element m_i in the group k takes the category h for the partition/variable λ_q. The density probability function of the model can be stated as:

$$f(\mathbf{M}; \boldsymbol{\Theta}) = \prod_{i=1}^{n} \sum_{k=1}^{g} \pi_k \prod_{q=1}^{r} \prod_{h=1}^{g} (\alpha_{kq}^{(h)})^{m_{iq}^{(h)}}, \tag{13}$$

where $\boldsymbol{\Theta} = (\pi, \boldsymbol{\alpha})$ are the parameters of the model with $\pi = (\pi_1, \ldots, \pi_k)$ being the proportions and $\boldsymbol{\alpha}$ the vector of the components parameters.

The Rmixmod package [5] [5, 28] is used to achieve our analysis. We employ the default settings to compute the clustering, allowing the selection between 10 parsimonious models according to the Bayesian information Criterion (BIC) [46].

5.6.1. Consensus results

Following the previous statements, we believe that using `SNMF (Random)` solutions could potentially improve the quality of the final partition. While they look unattractive compared to those of `SNMF (Skmeans)` due to their lower performance (see Table 3 where overall, SNMF (Random) appears to be a bad initialization strategy except for RCV1), these solutions still lead to minima which in an unsupervised situation, could benefit to other groups of individuals. More specifically, clusters could be different to the ones captured by `SNMF (Skmeans)` and therefore might bring another source of information to get closer to the actual partition. Our proposition referred to as SNMF (Skmeans & Random) consists in retrieving the 5 top SNMF solutions given by each initialization strategy (Skmeans

[5] https://cran.r-project.org/web/packages/Rmixmod/Rmixmod.pdf

and Random) and performing a consensus using the ensemble methods defined earlier. For comparison, we also provide a consensus for `SNMF (Skmeans)` and `SNMF (Random)` individually. Table 3 also reports the average performances of the mix of solutions of SNMF (Skmeans & Random). Consensus results obtained with CE and MMM for each strategy are also available.

Figure 7 displays the pairwise NMI and ARI between the top partitions of each strategy: SNMF (Skmeans) denoted "SNMF Sk", SNMF (Random) denoted "SNMF Ra" and SNMF (Skmeans& Random) denoted "SNMF Sk & Ra". This allow us to assess how similar/related the respective partitions of each strategy are among each other. For instance SNMF Sk & Ra will translate how different SNMF (Random) solutions are from SNMF (Skmeans), while SNMF Sk relates how different SNMF (Skmeans solutions) are between each other. The closer we are to 1, the less diversity there is in the set of partitions.

Through our experiments, one can wonder what strategy should we use to improve clustering performance ? As we are in an unsupervised context, this question is difficult but through our obtained results we can nevertheless make some useful recommendations for the user.

1. First, it is clear that the MMM approach is undoubtedly superior to the CE approach [50] (see Table 3).

2. Between the two approaches Skmeans and Random, the former seems more often better than the latter. This can be due to the diversity it offers; see for example SPORTS and NG20.

3. In the absence of diversity, the MMM approach does not bring improvement whatever the strategy used (Skmeans or Random). In this case combining

them (Skmeans & Random) can even degrade the result as is the case with RCV1. Otherwise, with a great diversity of the two strategies one can expect an improvement; this is the case of NG20.

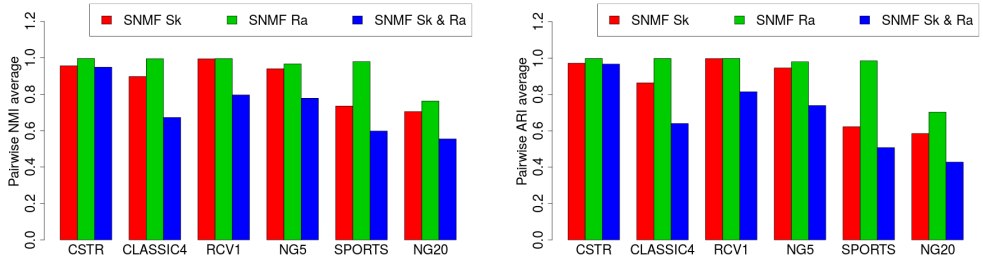


Figure 7: Pairwise NMI & ARI averages between the top 10 solutions.

Table 3: Mean ± SD of NMI and ARI & consensus over different datasets using CE and the Multinomial Mixture Model (MMM).

Datasets	Metrics	SNMF (Skmeans)			SNMF (Random)			SNMF (Skmeans & Random)		
		Mean±SD	CE	MMM	Mean±SD	CE	MMM	Mean±SD	CE	MMM
CSTR	NMI	0.76±0.01	(0.76)	(0.76)	0.75±0.00	(0.75)	(0.75)	0.75±0.00	(0.75)	(**0.77**)
	ARI	0.80±0.01	(0.80)	(0.80)	0.80±0.00	(0.80)	(0.80)	0.80±0.00	(0.80)	(**0.81**)
CLASSIC4	NMI	0.61±0.03	(0.60)	(0.60)	0.54±0.00	(0.49)	(0.54)	0.58±0.05	(0.57)	(**0.65**)
	ARI	0.47±0.00	(0.47)	(0.47)	0.38±0.00	(0.31)	(0.38)	0.48±0.05	(0.40)	(**0.47**)
RCV1	NMI	0.56±0.00	(0.56)	(0.56)	0.61±0.00	(0.51)	(**0.61**)	0.59±0.03	(0.51)	(0.52)
	ARI	0.45±0.00	(0.45)	(0.45)	0.63±0.00	(0.38)	(**0.63**)	0.54±0.04	(0.45)	(0.45)
NG5	NMI	0.78±0.00	(**0.78**)	(**0.78**)	0.67±0.00	(0.67)	(0.67)	0.73±0.06	(0.67)	(0.77)
	ARI	0.75±0.01	(0.75)	(0.74)	0.64±0.00	(0.64)	(0.64)	0.69±0.06	(0.60)	(**0.79**)
SPORTS	NMI	0.63±0.04	(0.63)	(**0.66**)	0.43±0.00	(0.43)	(0.43)	0.54±0.12	(0.53)	(0.57)
	ARI	0.48±0.05	(0.48)	(**0.54**)	0.32±0.00	(0.32)	(0.32)	0.41±0.10	(0.40)	(0.46)
NG20	NMI	0.49±0.02	(0.50)	(0.50)	0.47±0.01	(0.47)	(0.47)	0.48±0.02	(0.50)	(**0.52**)
	ARI	0.33±0.03	(0.33)	(0.30)	0.32±0.02	(0.33)	(0.33)	0.32±0.02	(0.34)	(**0.37**)

6. Discussion

In this section, we discuss some directions that we have already investigated since we developed Semantic-NMF. We also discuss some weaknesses and possible improvements of Semantic-NMF.

6.1. The orthogonality constraint

The orthogonality constraint on $\mathbf{Z}$ is almost always adopted for the clustering task [17, 58]. With this constraint NMF is equivalent to k-means clustering, and

several work empirically demonstrated that such constrain improves the clustering performance of NMF, in most situations. In our case, we found that the orthogonality constraint on $\mathbf{Z}$ has only a slight impact on the performances of Semantic-NMF. Since this constraint adds a little computational overhead, we have chosen not to consider it for efficiency purposes. Note that, introducing the orthogonality constraint into Semantic-NMF is trivial as we only need to replace the update rule of $\mathbf{Z}$ (7a) by the one of Orthogonal NMF [17, 58].

6.2. Regularizing document factors using document-document co-occurrences

A natural extension of Semantic-NMF is to regularize the document factors using the document-document co-occurrence information. While such an extension is expected to yield further improvements, our first results show that in some cases adding this regularization declines the clustering performance of Semantic-NMF. We believe that this is might be due to the fact that even the most closely related documents do not necessarily use exactly the same words. We are currently performing further investigations and try to figure out what is causing this issue.

6.3. Weaknesses and possible improvements

Although we have shown that Semantic-NMF improves the performances of NMF models by a noticeable amount, Semantic-NMF has two potential weaknesses: (i) as in most NMF models, the dimensionality, g, of the latent space is the same for both documents and words. For the clustering task, g also denotes the number of clusters. When the latter is small (< 10), this may not be enough to learn high quality word representations that capture finer linguistic regularities and patterns between words. A better alternative, is to make the dimensionality of the word embeddings independent from the number of clusters. This is possible using Non-Negative Matrix Tri-factorization [17]. (ii) In some situations, when

the PPMI matrix, $\mathbf{M}$, is defined deterministically from the local corpus of each dataset—as this is the case in this paper—, Semantic-NMF does not have a clear generative interpretation, which could limit the scope of its use. We can overcome this weakness by using a huge external corpora such as `WIKIPEDIA` and `GOOGLE` to build the PPMI matrix. In this case, not only Semantic-NMF has a clear generative interpretation and can be embedded in a well defined probabilistic model [40], but also the PPMI matrix encodes richer and more accurate semantic regularities between words. Leveraging a huge external corpora, such as the aforementioned ones, so as to preserve semantics in NMF, constitutes our main focus for a future extension of Semantic-NMF.

7. Conclusion

We describe Semantic-NMF, a novel non-negative factorization model which explicitly accounts for the semantic relationships among words. Similar to neural word embedding techniques, our model follows the distributional hypothesis so as to leverage the relationships between words. Formally, Semantic-NMF jointly decomposes the document-word and PPMI word-context matrices, with shared word factors. The intuition behind our approach, is to map words having common meaning roughly to the same direction in the latent space. More interestingly, by capturing more semantics, our model implicitly brings the embeddings of documents which are about the same topic closer to each other, as illustrated in our experiments. This results in document factors that are even better for clustering. Moreover, we identify in which situations Semantic-NMF does provide the most significant improvements, which allows us to gain further insights into the benefits of leveraging the word relationships.

Overall, our findings suggest that, leveraging the contextual relationships between words, explicitly, makes it possible to preserve more semantics and drastically improve the clustering performance of NMF models. Interestingly, our approach does not require an additional source of data.

Semantic-NMF is a flexible model that can be extended in several directions, which open up good opportunities for future research. For instance, it could benefit from the wide range of regularization schemes already applied to the original NMF. As a concrete example, we believe that it would be very interesting to combine Graph regularized NMF (GNMF), which relies on document manifold regularization, with Semantic-NMF to lead to Semantic-GNMF. We expect such combination to yield further improvements as manifold regularization has proven to be useful for clustering [9, 47, 53, 31]. On the other hand, the idea of Semantic-NMF could be extended to other variants of NMF. Furthermore, because theoretical connections have been already established between NMF and k-means, spectral clustering, our work could be the building block for *Semantic Clustering Models*, i.e., clustering models which account for the semantic relationships between words.

References

[1] Melissa Ailem, Aghiles Salah, and Mohamed Nadif. Non-negative matrix factorization meets word embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1081–1084, 2017.

[2] Kais Allab, Lazhar Labiod, and Mohamed Nadif. A semi-nmf-pca unified framework for data clustering. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):2–16, 2016.

[3] Kais Allab, Lazhar Labiod, and Mohamed Nadif. Multi-manifold matrix decomposition for data co-clustering. *Pattern Recognition*, 64:386–398, 2017.

[4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155, 2003.

[5] Christophe Biernacki, Gilles Celeux, Gérard Govaert, and Florent Langrognet. Model-based cluster and discriminant analysis with the mixmod software. *Computational Statistics & Data Analysis*, 51(2):587–600, 2006.

[6] Paul S Bradley and Usama M Fayyad. Refining initial points for k-means clustering. In *ICML*, volume 98, pages 91–99. Citeseer, 1998.

[7] John A Bullinaria and Joseph P Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39(3):510–526, 2007.

[8] Deng Cai, Xiaofei He, and Jiawei Han. Locally consistent concept factorization for document clustering. *IEEE Transactions on Knowledge and Data Engineering*, 23(6):902–913, 2010.

[9] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.

[10] Andrzej Cichocki, Hyekyoung Lee, Yong-Deok Kim, and Seungjin Choi. Non-negative matrix factorization with α-divergence. *Pattern Recognition Letters*, 29(9):1433–1440, 2008.

[11] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. Csiszar's divergences for non-negative matrix factorization: Family of new algorithms. In *International Conference on Independent Component Analysis and Signal Separation*, pages 32–39. Springer, 2006.

[12] Nicoletta Del Buono and Gianvito Pio. Non-negative matrix tri-factorization for co-clustering: an analysis of the block matrix. *Information Sciences*, 301:13–26, 2015.

[13] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[14] Inderjit S Dhillon and Dharmendra S Modha. Concept decompositions for large sparse text data using clustering. *Machine learning*, 42(1):143–175, 2001.

[15] Inderjit S Dhillon and Suvrit Sra. Generalized nonnegative matrix approximations with bregman divergences. In *NIPS*, volume 18. Citeseer, 2005.

[16] Chris Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 606–610. SIAM, 2005.

[17] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135, 2006.

[18] Chris HQ Ding, Tao Li, and Michael I Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):45–55, 2008.

[19] Hongchang Gao, Feiping Nie, and Heng Huang. Local centroids structured non-negative matrix factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[20] Pinghua Gong and Changshui Zhang. Efficient nonnegative matrix factorization via projected newton method. *Pattern Recognition*, 45(9):3557–3565, 2012.

[21] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[22] Jin Huang, Feiping Nie, Heng Huang, and Chris Ding. Robust manifold nonnegative matrix factorization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(3):1–21, 2014.

[23] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[24] George Karypis. Cluto-a clustering toolkit. Technical report, MINNESOTA UNIV MINNEAPOLIS DEPT OF COMPUTER SCIENCE, 2002.

[25] Bradley Klingenberg, James Curry, and Anne Dougherty. Non-negative matrix factorization: Ill-posedness and a geometric algorithm. *Pattern Recognition*, 42(5):918–928, 2009.

[26] Lazhar Labiod and Mohamed Nadif. Co-clustering under nonnegative ma-

trix tri-factorization. In *International Conference on Neural Information Processing*, pages 709–717. Springer, 2011.

[27] Rémi Lebret and Ronan Collobert. Word emdeddings through hellinger pca. *arXiv preprint arXiv:1312.5542*, 2013.

[28] Rémi Lebret, Serge Iovleff, Florent Langrognet, Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Rmixmod: the r package of the model-based unsupervised, supervised and semi-supervised classification mixmod library. 2015.

[29] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[30] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[31] Chengcai Leng, Guorong Cai, Dongdong Yu, and Zongyue Wang. Adaptive total-variation for non-negative matrix factorization on manifold. *Pattern Recognition Letters*, 98:68–74, 2017.

[32] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27:2177–2185, 2014.

[33] Tao Li. A general model for clustering binary data. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 188–197, 2005.

[34] Tao Li and Cha-charis Ding. Nonnegative matrix factorizations for clustering: A survey. In *Data Clustering*, pages 149–176. Chapman and Hall/CRC, 2018.

[35] Tao Li and Chris Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 362–371. IEEE, 2006.

[36] Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments, & computers*, 28(2):203–208, 1996.

[37] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013.

[38] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

[39] Andriy Mnih and Geoffrey E Hinton. A scalable hierarchical distributed language model. *Advances in neural information processing systems*, 21:1081–1088, 2008.

[40] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20:1257–1264, 2007.

[41] David Newman, Sarvnaz Karimi, and Lawrence Cavedon. External evaluation of topic models. In *in Australasian Doc. Comp. Symp., 2009*. Citeseer, 2009.

[42] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 100–108, 2010.

[43] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[44] Wei Qian, Bin Hong, Deng Cai, Xiaofei He, Xuelong Li, et al. Non-negative matrix factorization with sinkhorn distance. In *IJCAI*, pages 1960–1966, 2016.

[45] François Role and Mohamed Nadif. Handling the impact of low frequency events on co-occurrence based measures of word similarity-a case study of pointwise mutual information. In *KDIR*, pages 226–231, 2011.

[46] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[47] Fanhua Shang, LC Jiao, and Fei Wang. Graph dual regularization non-negative matrix factorization for co-clustering. *Pattern Recognition*, 45(6):2237–2250, 2012.

[48] Amanda J.C Sharkey. On combining artificial neural nets. *Connection Science*, 8(3-4):299–314, 1996.

[49] Amanda JC Sharkey. Multi-net systems. In *Combining artificial neural nets*, pages 1–30. Springer, 1999.

[50] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

[51] Alexander Topchy, Anil K Jain, and William Punch. A mixture model for clustering ensembles. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 379–390. SIAM, 2004.

[52] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

[53] Jim Jing-Yan Wang, Halima Bensmail, and Xin Gao. Multiple graph regularized nonnegative matrix factorization. *Pattern Recognition*, 46(10):2840–2847, 2013.

[54] Yu-Xiong Wang and Yu-Jin Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1336–1353, 2012.

[55] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273, 2003.

[56] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870, 2017.

[57] Zhirong Yang and Erkki Oja. Linear and nonlinear projective nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, 21(5):734–749, 2010.

[58] Jiho Yoo and Seungjin Choi. Orthogonal nonnegative matrix factorization: Multiplicative updates on stiefel manifolds. In *International conference on intelligent data engineering and automated learning*, pages 140–147. Springer, 2008.

[59] Jiho Yoo and Seungjin Choi. Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on stiefel manifolds. *Information processing & management*, 46(5):559–570, 2010.

[60] Zhijian Yuan, Zhirong Yang, and Erkki Oja. Projective nonnegative matrix factorization: Sparseness, orthogonality, and clustering. *Neural Process. Lett*, pages 11–13, 2009.