A Consensus Approach to Improve NMF Document Clustering

Mickael Febrissy(✉) and Mohamed Nadif

LIPADE, Université de Paris, 75006 Paris, France
mickael.febrissy@u-paris.fr

Abstract. Nonnegative Matrix Factorization (NMF) which was originally designed for dimensionality reduction has received throughout the years a tremendous amount of attention for clustering purposes in several fields such as image processing or text mining. However, despite its mathematical elegance and simplicity, NMF has exposed a main issue which is its strong sensitivity to starting points, resulting in NMF struggling to converge toward an optimal solution. On another hand, we came to explore and discovered that even after providing a meaningful initialization, selecting the solution with the best local minimum was not always leading to the one having the best clustering quality, but somehow a better clustering could be obtained with a solution slightly off in terms of criterion. Therefore in this paper, we undertake to study the clustering characteristics and quality of a set of NMF best solutions and provide a method delivering a better partition using a consensus made of the best NMF solutions.

Keywords: NMF · Clustering · Clustering ensemble · Consensus

1 Introduction

When dealing with text data, document clustering techniques allow to divide a set of documents into groups so that documents assigned to the same group are more similar to each other than to documents assigned to other groups [12,18,21,22]. In information retrieval, the use of clustering relies on the assumption that if a document is relevant to a query, then other documents in the same cluster can also be relevant. This hypothesis can be used at different stages in the information retrieval process, the two most notable being: cluster-based retrieval to speed up search, and search result clustering to help users navigate and understand what is in the search results. The document clustering which still remains a hot topic can be tackled under different approaches. In our contribution we rely on the non-negative matrix factorization for its simplicity and popularity. We will not propose a new variant of NMF but rather a consensus approach that will boost its performance.

Unlike supervised learning, the evaluation of clustering algorithms - unsupervised learning - remains a difficult problem. When relying on generative models,

M. R. Berthold et al. (Eds.): IDA 2020, LNCS 12080, pp. 171–183, 2020.
https://doi.org/10.1007/978-3-030-44584-3_14

it is easier to evaluate the performance of a given clustering algorithm based on the simulated partition. On real data already labeled, many papers evaluate the performance of clustering algorithms by relying on indices such as Accuracy (ACC), Normalized Mutual Information (NMI) [25] and Adjusted Rand Index (ARI) [14]. However, the algorithms commonly used which are of type k-means, EM [8], Classification EM [6], NMF [15] etc. are iterative and require several initializations; the resulting partition is the one optimizing the objective function. Sometimes in these works, we observe comparative studies between methods on the basis of maximum ACC/NMI/ARI measures obtained after several initializations and not optimizing the criterion used in the algorithm. Such a comparison is thereby not accurate, because in fact these measures cannot be calculated in practice and cannot be used in this way to evaluate the quality of a clustering algorithm.

A fair comparison can only be made on the basis of objective functions considered in a clustering purpose; for example, within-cluster inertia, likelihood, classification likelihood for mixture models, factorization, etc. Nonetheless, in our experiences, we realized that while the clustering results become better in terms of ACC/NMI/ARI when the objective function value increases, the best value is not necessarily associated with the best results. However, by ranking the objective values, the best partition tends to be among those leading to the first best scores. We illustrate this behavior in Fig. 4. This remark leads us to consider an *ensemble method* that is widely used in supervised learning [11,24] but a little less in unsupervised learning [25]. If this approach, referred to as *consensus clustering*, is often used in the context of comparing partitions obtained with different algorithms, it is less studied considering the same algorithm.

The paper is organized as follows. In Sect. 2, we review the nonnegative matrix factorization with the Frobenius norm and the Kullback–Leibler divergence. Section 3 is devoted to describe the ensemble method and the popular used algorithms. In Sect. 4, we perform comparisons on document-term matrices and propose a strategy to improve document clustering with NMF.

2 Nonnegative Matrix Factorization

Nonnegative Matrix Factorization (NMF) [15], aiming to deliver a lower rank decomposition of a nonnegative data matrix $\boldsymbol{X}$ has highlighted clustering properties for which strong connections with K-means or Spectral clustering can be drawn [16]. However, while several variants arise in order to accommodate its clustering property [10,29–31], its premier model formulation does not involve a clustering objective and was originally presented as a dimension reduction algorithm with exclusive nonnegative factors. More specifically in text mining where NMF produces a meaningful interpretation for document-term matrices in comparison with methods like Singular Value Decomposition (SVD) components or Latent Semantic Analysis (LSA) [7] arising factors with possible negative values. NMF seeks to approximate a matrix $\boldsymbol{X} \in \mathbb{R}_+^{n \times d}$ by the product of two lower rank matrices $\boldsymbol{Z} \in \mathbb{R}_+^{n \times g}$ and $\boldsymbol{W} \in \mathbb{R}_+^{d \times g}$ with $g(n+d) < ng$. This problem can be formulated as a constrained optimization problem

$$\mathrm{F}(\boldsymbol{Z},\boldsymbol{W}) = \underset{\boldsymbol{Z}\geq 0,\boldsymbol{W}\geq 0}{min} D(\boldsymbol{X},\boldsymbol{Z}\boldsymbol{W}^\top) \tag{1}$$

where D is a fitting error allowing to measure the quality of the approximation of $\boldsymbol{X}$ by $\boldsymbol{Z}\boldsymbol{W}^\top$, the most popular ones being the Frobenius norm and Kullback-Leibler (KL) divergence. For a clustering setup, $\boldsymbol{Z}$ will be referred to as the soft classification matrix while $\boldsymbol{W}$ will be the centers matrix. Despite its multiple applications benefits, NMF has a recurrent downside which takes place at its initialization. NMF provides a different solution for every different initialisation making it substantially sensitive to starting points as its convergence directly relies on the characteristics of the given entries. Several publications have shown interest in finding the best way to start a NMF algorithm by providing a structured initialization, in some cases obtained from results of clustering algorithms such as k-means or Spherical K-means [27,28] (especially for applying NMF on document-term matrices), Nonnegative Singular Value decomposition (NNDSVD) [4] or SVD based strategies [17]. The optimization procedures for D respectively equal to the Frobenius norm and the KL divergence, based on multiplicative update rules are given in Algorithms 1 and 2.

Algorithm 1. (`NMF-F`).

```
Input: $\boldsymbol{X}$, $g$, $\boldsymbol{Z}^{(0)}$; $\boldsymbol{W}^{(0)}$.
Output: $\boldsymbol{Z}$ and $\boldsymbol{W}$.
repeat
    1. $\boldsymbol{Z} \leftarrow \boldsymbol{Z} \odot \frac{\boldsymbol{X}\boldsymbol{W}}{\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W}}$;
    2. $\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{\boldsymbol{X}^\top\boldsymbol{Z}}{\boldsymbol{W}\boldsymbol{Z}^\top\boldsymbol{Z}}$;
until convergence
5. Normalize $\boldsymbol{Z}$ so as it has unit-length column vectors.
```

Algorithm 2. (`NMF-KL`).

```
Input: $\boldsymbol{X}$, $g$, $\boldsymbol{Z}^{(0)}$; $\boldsymbol{W}^{(0)}$.
Output: $\boldsymbol{Z}$ and $\boldsymbol{W}$.
repeat
    1. $\boldsymbol{Z} \leftarrow \boldsymbol{Z} \odot \big(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top}\boldsymbol{W}\big)/\sum_j \boldsymbol{W}_{jk}$;
    2. $\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \big(\frac{\boldsymbol{X}^\top}{\boldsymbol{W}\boldsymbol{Z}^\top}\boldsymbol{Z}\big)/\sum_i \boldsymbol{Z}_{ik}$;
until convergence
5. Normalize $\boldsymbol{Z}$ so as it has unit-length column vectors.
```

3 Cluster Ensembles (CE)

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 [24,25]. 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 [5] who tried to combine several clustering partitions according to the combination of the cluster centers. In the early 2000, [25] 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 possibilities 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 [25], the authors introduced three consensus methods that can produce a partition. All of them consider the consensus problem on a hypergraph representation $\boldsymbol{H}$ of the set of partitions $\boldsymbol{H}^r$. More specifically, each partition $\boldsymbol{H}^r$ equals a binary classification matrix (with objects in rows and clusters in columns) where the concatenation of all the set defines the hypergraph $\boldsymbol{H}$.

- 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 [25] 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:

$$\lambda^{(k-opt)} = \underset{\widetilde{\lambda}}{argmax} \sum_{q=1}^{r} \text{NMI}(\widetilde{\lambda}, \lambda^{(q)}) \tag{2}$$

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:

$$\text{ANMI}(\Lambda, \widetilde{\lambda}) = \frac{1}{r} \sum_{q=1}^{r} \text{NMI}(\widetilde{\lambda}, \lambda^{(q)}) \tag{3}$$

To cast with cases where the vector labels $\lambda^{(q)}$ have missing values, the authors have proposed a generalized expression of (2) not substantially different that viewers can refer to in the original paper [25].

4 Experiments

We conduct several experiences leading to emphasise the behavior of NMF regarding a clustering task compared to a dedicated clustering algorithm such as Spherical K-means referred to as `S-Kmeans` [9] which was introduced for clustering large sets of sparse text data (or directional data) and remains appealing for its low computational cost beside its good performances. It was also retained along side the random starting points (generated according to an uniform distribution $\mathcal{U}(0,1) \times mean(\boldsymbol{X})$) as initialization for NMF. We use two error measures frequently employed for NMF: the Frobenius norm (which will be referred to as `NMF-F`) and the Kullback-Leibler divergence (`NMF-KL`). Eventually, we compute the consensus partition by using the Cluster Ensemble Python package[1] which utilizes the consensus methods defined earlier [25].

4.1 Datasets

We apply NMF on 5 bench-marking document-term matrices for which the detailed characteristics are available in Table 1 where nz indicates the percentage of values other than 0 and the *balance* coefficient is defined as the ratio of the number of documents in the smallest class to the number of documents in the largest class. These datasets highlight several varieties of challenging situations such as the amount of clusters, the dimensions, the clusters balance, the degree of mixture of the different groups and the sparsity. We normalized each data matrix with TF-IDF and their respective documents-vectors to unit L_2-norm to remove the bias introduced by their length.

Table 1. Datasets description: # denotes the cardinality

Datasets	Characteristics				
	#Documents	#Words	#Clusters	nz(%)	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
NG20	18846	14390	20	0.59	0.628

4.2 NMF Raw Performances and Initialization

The results obtained by `NMF-F` and `NMF-KL` according to `S-Kmeans` and the random starting points are available in Table 2. The clustering quality of the

[1] https://pypi.org/project/Cluster_Ensembles/.

S-Kmeans partitions given as entry to both algorithms are also displayed. We make use of two relevant measures to quantify and assess the clustering quality of each algorithm. The first one is the NMI [25] which quantifies how much information the clustering partition shares with the true partition, the second is the ARI [14], sensitive to the clusters proportions and measures the degree of agreement between the clustering and the true partition. To replicate a relevant user experience achieving an unsupervised task, we refer to the criterion of each algorithm in order to select the 10 first best solutions (out of 30 runs) and report their average NMI and ARI with the true partition.

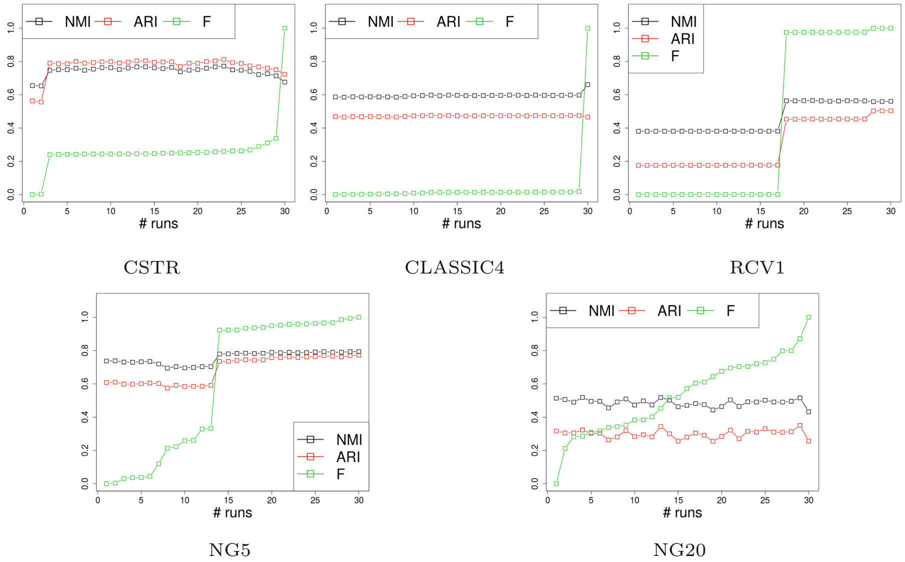
One can clearly see that NMF-F and NMF-KL do not react similarly to the different initializations. While NMF-F substantially benefits from the S-kmeans initialization on every datasets compared to the random initialization, NMF-KL does not seem to accommodate S-kmeans entries. In fact, S-Kmeans as starting values seems to worsen NMF-KL solutions, especially on CLASSIC4 and NG5. For this reason, we will avoid this initialization strategy for NMF-KL in the future although it improves on RCV1. Also, NMF-KL with a random initialization provides much better results than the other algorithms on almost all datasets.

Table 2. Mean and standard deviation of NMI and ARI computed over the 10 best solutions.

Datasets	Metrics	Skmeans	NMF-F (Random)	NMF-F (Skmeans)	NMF-KL (Random)	NMF-KL (Skmeans)
CSTR	NMI	0.76 ± 0.007	0.65 ± 0.002	0.73 ± 0.04	0.73 ± 0.03	0.76 ± 0.006
	ARI	0.80 ± 0.007	0.55 ± 0.002	0.75 ± 0.10	0.77 ± 0.04	0.80 ± 0.006
CLASSIC4	NMI	0.60 ± 0.001	0.53 ± 0.003	0.59 ± 0.002	0.71 ± 0.02	0.61 ± 0.03
	ARI	0.47 ± 0.0009	0.45 ± 0.003	0.47 ± 0.002	0.65 ± 0.06	0.47 ± 0.004
RCV1	NMI	0.38 ± 0.0003	0.35 ± 0.0005	0.38 ± 0.0002	0.47 ± 0.02	0.53 ± 0.002
	ARI	0.18 ± 0.0004	0.13 ± 0.0008	0.18 ± 0.0003	0.42 ± 0.02	0.46 ± 0.02
NG5	NMI	0.72 ± 0.02	0.56 ± 1.0e−05	0.72 ± 0.02	0.80 ± 0.03	0.79 ± 0.003
	ARI	0.60 ± 0.01	0.33 ± 2.5e−05	0.60 ± 0.01	0.82 ± 0.04	0.76 ± 0.005
NG20	NMI	0.49 ± 0.02	0.41 ± 0.01	0.49 ± 0.02	0.48 ± 0.02	0.51 ± 0.01
	ARI	0.30 ± 0.02	0.23 ± 0.01	0.30 ± 0.02	0.34 ± 0.02	0.32 ± 0.02

We reported in Figs. 1, 2, 3 and 4 the clustering quality of the algorithm's solutions ranked from the best one in terms of criterion to the poorest one. The respective criterion of each algorithm is normalized to belong to $[0, 1]$.

When one does have the real partition, a common practice to evaluate the clustering result, one relies on the best solution obtained by optimizing the objective function. Figures 1 and 3 highlight a critical behavior of NMF-F which tends to produce solutions with the lowest minima that do not fulfil the best clustering partitions, sometimes with a substantial gap (see CSTR, RCV1, NG5 in Fig. 1). Moreover, a surprising lesser but still similar behavior is delivered by S-Kmeans which compared to NMF, optimizes a clustering objective by definition. The results are displayed in Fig. 2. In reality, this behavior can be observed with several types of what we refer to clustering algorithms hosting an optimization procedure. Initializing NMF-F randomly as shown in Fig. 3 seems to lighten this



CSTR

CLASSIC4

RCV1

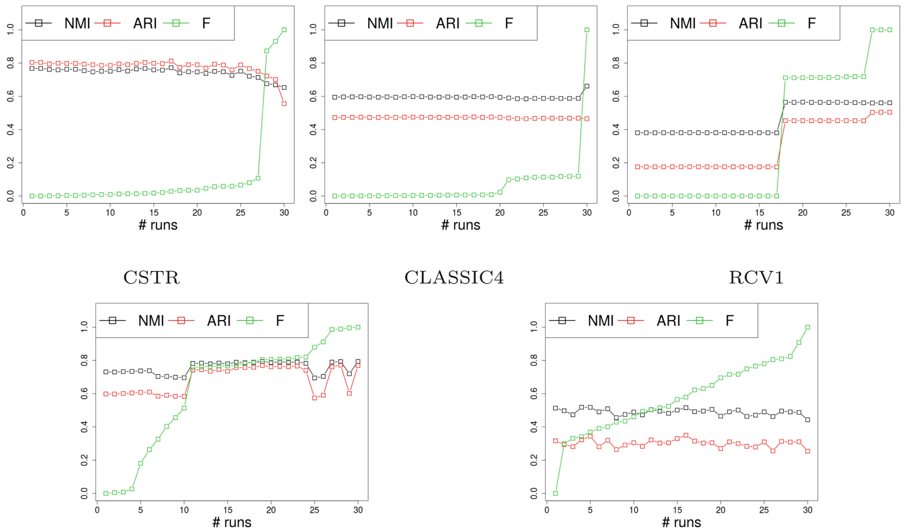
NG5

NG20

Fig. 1. `NMF-F`: NMI/ARI behaviour according to the objective function F (initializations by `S-Kmeans`)

effect (on CSTR, Classic4 and RCV1). On another hand, `NMF-KL` which to this day remains recognized as a relevant method for document clustering [13] seems to consistently deliver solutions with the lowest criteria aligned with the goodness of their clustering, sustaining the use of NMF for clustering purposes. Furthermore, compared to all, `NMF-KL` is the only method emphasizing a wide variety of solutions and therefore seems to explore way more possibilities than `NMF-F` or `S-Kmeans`. Its better behavior might almost comfort the idea of selecting the best partition in terms of criterion as the one to keep. However, it still fails on RCV1 which is the toughest dataset to partition mainly because of its scant density. Eventually, it remains concerning to select the best partition just based on the fact that, even with `NMF-KL`, the solution among the best ones providing the best clustering, is not necessarily the first one (see on CSTR, CLASSIC4 and NG5).

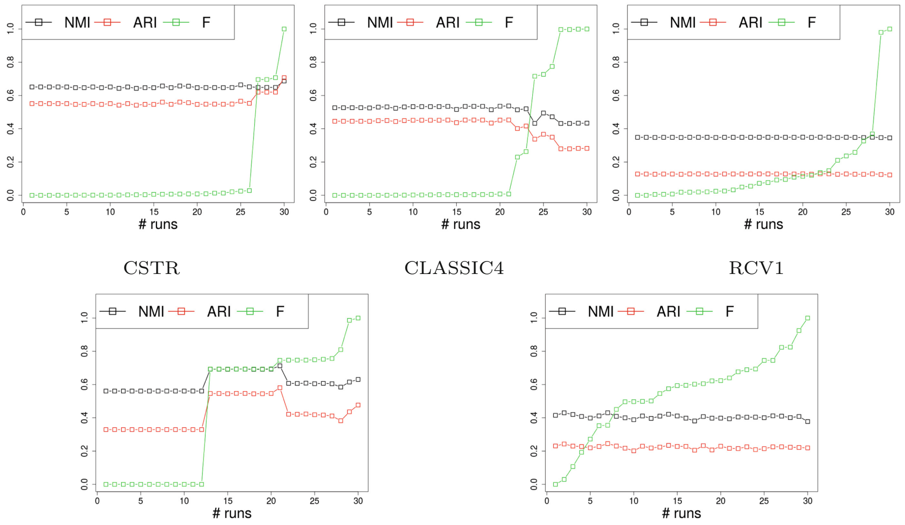
In addition, while the best solutions possibly share a similar amount of information with the true partition, they could be fairly distinct from each other, making their use appealing to deduce an even more exhaustive solution. Figure 5 shows results of pairwise NMI and ARI between the top 10 partitions (criterion-wise) of each algorithm. `NMF-KL`'s best solutions appear to be fairly different among each other.



CSTR CLASSIC4 RCV1

NG5 NG20

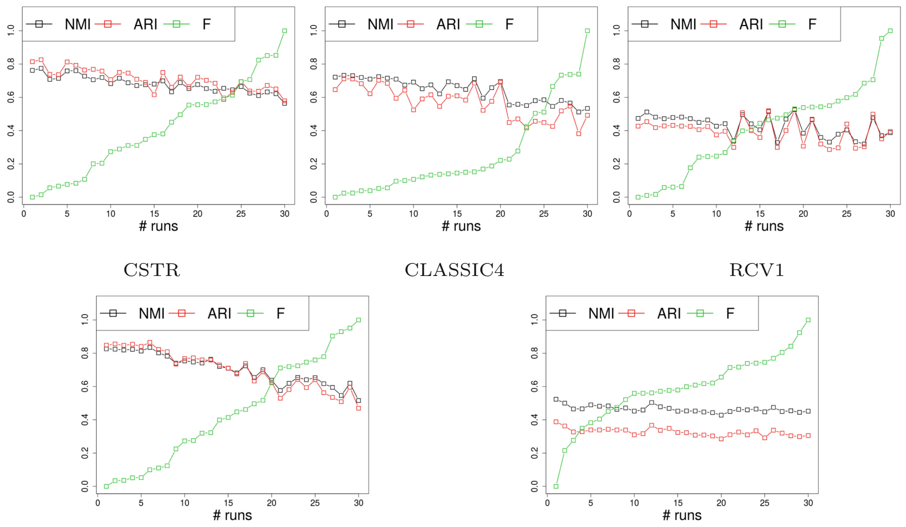
Fig. 2. `S-Kmeans`: NMI/ARI behaviour according to the objective function F (Random initializations)



CSTR CLASSIC4 RCV1

NG5 NG20

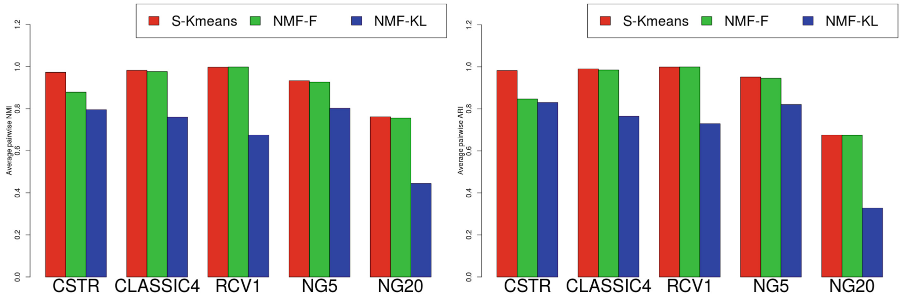
Fig. 3. `NMF-F`: NMI/ARI behaviour according to the objective function F (Random initializations)



CSTR CLASSIC4 RCV1

NG5 NG20

Fig. 4. `NMF-KL`: NMI/ARI behaviour according to the objective function F (Random initializations)



Average pairwise NMI Average pairwise ARI

Fig. 5. Average pairwise NMI & ARI between top 10 solutions

4.3 Consensus Clustering

Following the previous statement, we went ahead and computed a cluster ensemble (CE) for `NMF-F` and `NMF-KL` according to their best initialization strategy as well as for `S-Kmeans` due to its pertinence for initializing `NMF-F` and the method being widely known as relevant for document clustering. The results are reported in Table 3. It appears that the consensus obtained with the top 10 results of each method generally outperforms the best solution. This result is even stronger for `NMF-KL` where the ensemble clustering increases the NMI and ARI by respectively 11 and 13 points on NG20. Note that NG20 is the dataset where the average pairwise NMI and ARI between the 10 top partitions are the lowest, meaning the most different (see Fig. 5). Furthermore, it is interesting to note that these performances are obtained from solutions giving an average NMI and ARI smaller than the best solution itself.

Table 3. Mean and standard deviation, first best result and CE consensus computed over the 10 best solutions.

Datasets	Metrics	`NMF-F (Skmeans)`			`Skmeans`			`NMF-KL (Random)`		
		Mean ± SD	(best)	CE	Mean ± SD	(best)	CE	Mean ± SD	(best)	CE
CSTR	NMI	0.73 ± 0.04	(0.65)	(0.76)	0.76 ± 0.007	(0.77)	(0.77)	0.73 ± 0.03	(0.76)	**(0.80)**
	ARI	0.75 ± 0.10	(0.56)	(0.80)	0.80 ± 0.007	(0.80)	(0.80)	0.77 ± 0.04	(0.81)	**(0.83)**
CLASSIC4	NMI	0.59 ± 0.002	(0.59)	(0.59)	0.60 ± 0.001	(0.59)	(0.60)	0.71 ± 0.02	(0.72)	**(0.74)**
	ARI	0.47 ± 0.002	(0.47)	(0.47)	0.47 ± 0.0009	(0.47)	(0.47)	0.65 ± 0.06	(0.65)	**(0.72)**
RCV1	NMI	0.38 ± 0.0002	(0.38)	(0.35)	0.38 ± 0.0003	(0.38)	(0.35)	0.47 ± 0.02	(0.47)	**(0.52)**
	ARI	0.18 ± 0.0003	(0.18)	(0.26)	0.18 ± 0.0004	(0.18)	(0.26)	0.42 ± 0.02	(0.43)	**(0.46)**
NG5	NMI	0.72 ± 0.02	(0.74)	(0.76)	0.72 ± 0.02	(0.73)	(0.75)	0.80 ± 0.03	(0.83)	**(0.86)**
	ARI	0.60 ± 0.01	(0.61)	(0.60)	0.60 ± 0.01	(0.60)	(0.64)	0.82 ± 0.04	(0.85)	**(0.88)**
NG20	NMI	0.49 ± 0.02	(0.51)	(0.50)	0.49 ± 0.02	(0.51)	(0.50)	0.48 ± 0.02	(0.50)	**(0.61)**
	ARI	0.30 ± 0.02	(0.32)	(0.34)	0.30 ± 0.02	(0.32)	(0.34)	0.34 ± 0.02	(0.36)	**(0.49)**

4.4 Consensus Multinomial

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 [26], 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 $\boldsymbol{M} \in \{0,1\}^{n\times rg}$ is usually used instead of Λ with value $m_{iq}^{(h)}$ where $h \in \{1,\ldots,g\}$ is a cluster label. Therefore, the data values $m_{iq}^{(h)}$ 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:

$$f(\boldsymbol{M};\boldsymbol{\theta}) = \prod_{i=1}^{n}\sum_{k=1}^{g}\pi_k \prod_{q,h}^{r,g}(\alpha_{kq}^{(h)})^{m_{iq}^{(h)}} \tag{4}$$

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

Table 4. MMM consensus results over the 10 best solutions

Datasets	Metrics	`NMF-KL (Random)`			
		Mean±SD	(best)	CE	MMM
CSTR	NMI	0.73 ± 0.03	(0.76)	**(0.80)**	(0.77)
	ARI	0.77 ± 0.04	(0.81)	**(0.83)**	(0.82)
CLASSIC4	NMI	0.71 ± 0.02	(0.72)	(0.74)	**(0.77)**
	ARI	0.65 ± 0.06	(0.65)	(0.72)	**(0.75)**
RCV1	NMI	0.47 ± 0.02	(0.47)	**(0.52)**	**(0.52)**
	ARI	0.42 ± 0.02	(0.43)	**(0.46)**	**(0.46)**
NG5	NMI	0.80 ± 0.03	(0.83)	**(0.86)**	**(0.86)**
	ARI	0.82 ± 0.04	(0.85)	(0.88)	**(0.89)**
NG20	NMI	0.48 ± 0.02	(0.50)	(0.61)	**(0.63)**
	ARI	0.34 ± 0.02	(0.36)	(0.49)	**(0.50)**

The Rmixmod package[2] 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) [23]. With CSTR, the model mainly selected is the one keeping the proportions π_k free with the model also independent from the variables (labels vectors), meaning $\mathcal{M}(m_{iq}^{(h)}; \alpha_k)$. CSTR is the dataset with the highest pairwise NMI and ARI therefore with the most similar best solutions. On CLASSIC4 and RCV1 where the pairwise NMI & ARI are a little bit lower, it is the model with free proportions and parameters $\boldsymbol{\alpha}$ depending on distinct components and labels vectors ($\mathcal{M}(m_{iq}^{(h)}; \alpha_{kq}^{(h)})$) which is mainly chosen. On NG5 where the best solutions are fairly similar (high pairwise NMI & ARI), it is the model depending on the components and the labels vectors which has been retained. However, the proportions here were kept equal. For NG20 where the best solutions were fairly distinct, the model selected is the one depending on the components and the variables. As previously, the proportions π_k are kept equal. Following the characteristics in Table 1, it is notable to see that the datasets where the proportions are kept equal are actually those with the more balanced real clusters proportions. The results of the obtained consensus are displayed in Table 4 which only retains prior results of `NMF-KL` top 10 solutions and CE consensus, as they were the best overall. Apart from CSTR, we can see that MMM does a better job at computing a better partition from the top 10 solutions than CE.

5 Conclusion

In this paper, by using *cluster ensembles*, we have proposed a simple method to obtain a better clustering for the scope of NMF algorithms on text data. From its

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

gathering nature, this process should also alleviate the uncertainty based around the overall quality of the final partition compared to other selection practices such as keeping an unique solution according to the best criterion. Furthermore, we have shown that it was possible to improve the consensus quality through the use of finite mixture models, allowing more powerful underlying settings than cluster-based consensus involving plain similarities or distances. A future work will be to investigate the use of *cluster ensembles* for other recent clustering algorithms [1–3, 19, 20].

References

1. Ailem, M., Salah, A., Nadif, M.: Non-negative matrix factorization meets word embedding. In: SIGIR, pp. 1081–1084 (2017)
2. Allab, K., Labiod, L., Nadif, M.: A semi-NMF-PCA unified framework for data clustering. IEEE Trans. Knowl. Data Eng. **29**(1), 2–16 (2016)
3. Allab, K., Labiod, L., Nadif, M.: Simultaneous spectral data embedding and clustering. IEEE Trans. Neural Netw. Learn. Syst. **29**(12), 6396–6401 (2018)
4. Boutsidis, C., Gallopoulos, E.: SVD based initialization: a head start for nonnegative matrix factorization. Pattern Recogn. **41**(4), 1350–1362 (2008)
5. Bradley, P.S., Fayyad, U.M.: Refining initial points for k-means clustering. In: ICML, vol. 98, pp. 91–99. Citeseer (1998)
6. Celeux, G., Govaert, G.: A classification EM algorithm for clustering and two stochastic versions. Comput. Stat. Data Anal. **14**(3), 315–332 (1992)
7. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391–407 (1990)
8. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the *EM* algorithm. J. Roy. Stat. Soc.: Ser. B (Methodol.) **39**(1), 1–22 (1977)
9. Dhillon, I.S., Modha, D.S.: Concept decompositions for large sparse text data using clustering. Mach. Learn. **42**(1–2), 143–175 (2001)
10. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix t-factorizations for clustering. In: SIGKDD, pp. 126–135. ACM (2006)
11. Ghosh, J.: Multiclassifier systems: back to the future. In: Roli, F., Kittler, J. (eds.) MCS 2002. LNCS, vol. 2364, pp. 1–15. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45428-4_1
12. Govaert, G., Nadif, M.: Mutual information, phi-squared and model-based co-clustering for contingency tables. Adv. Data Anal. Classif. **12**(3), 455–488 (2016). https://doi.org/10.1007/s11634-016-0274-6
13. Hosseini-Asl, E., Zurada, J.M.: Nonnegative matrix factorization for document clustering: a survey. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2014. LNCS (LNAI), vol. 8468, pp. 726–737. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07176-3_63
14. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**(1), 193–218 (1985)
15. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems, pp. 556–562 (2001)
16. Li, T., Ding, C.: The relationships among various nonnegative matrix factorization methods for clustering. In: ICDM, pp. 362–371 (2006)
17. Qiao, H.: New SVD based initialization strategy for non-negative matrix factorization. Pattern Recogn. Lett. **63**, 71–77 (2015)

18. Role, F., Morbieu, S., Nadif, M.: Coclust: a Python package for co-clustering. J. Stat. Softw. **88**, 1–29 (2019)
19. Salah, A., Ailem, M., Nadif, M.: A way to boost SEMI-NMF for document clustering. In: CIKM, pp. 2275–2278 (2017)
20. Salah, A., Ailem, M., Nadif, M.: Word co-occurrence regularized non-negative matrix tri-factorization for text data co-clustering. In: AAAI, pp. 3992–3999 (2018)
21. Salah, A., Nadif, M.: Model-based von Mises-Fisher co-clustering with a conscience. In: Proceedings of the 2017 SIAM International Conference on Data Mining, pp. 246–254. SIAM (2017)
22. Salah, A., Nadif, M.: Directional co-clustering. Adv. Data Anal. Classif. **13**(3), 591–620 (2018). https://doi.org/10.1007/s11634-018-0323-4
23. Schwarz, G., et al.: Estimating the dimension of a model. Ann. Stat. **6**(2), 461–464 (1978)
24. Sharkey, A.J.: Multi-net systems. In: Sharkey, A.J.C. (ed.) Combining Artificial Neural Nets, pp. 1–30. Springer, London (1999). https://doi.org/10.1007/978-1-4471-0793-4_1
25. Strehl, A., Ghosh, J.: Cluster ensembles-a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**(Dec), 583–617 (2002)
26. Topchy, A., Jain, A.K., Punch, W.: A mixture model for clustering ensembles. In: SDM, pp. 379–390. SIAM (2004)
27. Wild, S., Curry, J., Dougherty, A.: Improving non-negative matrix factorizations through structured initialization. Pattern Recogn. **37**(11), 2217–2232 (2004)
28. Wild, S., Wild, W.S., Curry, J., Dougherty, A., Betterton, M.: Seeding non-negative matrix factorizations with the spherical k-means clustering. Ph.D. thesis, University of Colorado (2003)
29. Yang, Z., Oja, E.: Linear and nonlinear projective nonnegative matrix factorization. IEEE Trans. Neural Netw. **21**(5), 734–749 (2010)
30. Yoo, J., Choi, S.: Orthogonal nonnegative matrix factorization: multiplicative updates on stiefel manifolds. In: Fyfe, C., Kim, D., Lee, S.-Y., Yin, H. (eds.) IDEAL 2008. LNCS, vol. 5326, pp. 140–147. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88906-9_18
31. Yuan, Z., Oja, E.: Projective nonnegative matrix factorization for image compression and feature extraction. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 333–342. Springer, Heidelberg (2005). https://doi.org/10.1007/11499145_35