# Model-based Co-clustering for High Dimensional Sparse Data

**Aghiles Salah**
aghiles.salah@parisdescartes.fr

**Nicoleta Rogovschi**
nicoleta.rogovschi@parisdescartes.fr

**Mohamed Nadif**
mohamed.nadif@parisdescartes.fr

LIPADE, University of Paris Descartes
45 Rue des Saints-Peres,
75006, Paris, France

## Abstract

We propose a novel model based on the von Mises-Fisher (vMF) distribution for co-clustering high dimensional sparse matrices. While existing vMF-based models are only suitable for clustering along one dimension, our model acts simultaneously on both dimensions of a data matrix. Thereby it has the advantage of exploiting the inherent duality between rows and columns. Setting our model under the maximum likelihood (ML) approach and the classification ML (CML) approach, we derive two novel, hard and soft, co-clustering algorithms. Empirical results on numerous synthetic and real-world text datasets, demonstrate the effectiveness of our approach, for modelling high dimensional sparse data and co-clustering. Furthermore, thanks to our formulation, that performs an implicitly adaptive dimensionality reduction at each stage, our model alleviates the problem of high concentration parameters kappa's, a well known difficulty in the classical vMF-based models.

## 1 Introduction

In the case of sparse high dimensional data, such as document-term and user-item matrices arising respectively in text mining and collaborative filtering, most of existing models such as Multinomial and Multivariate Gaussian mixture models, suffer from low performances. The von Mises-Fisher (vMF) distribution based models, dealing with directional data distributed on the surface of a unit hypersphere, may turn out to be a wise choice. In fact, the mixture of vMF distributions is one of the most appropriate model for modelling and clustering high dimensional sparse data, such as document-term matrices. In this context, it has been empirically demonstrated that vMF-based models performs better than several existing approaches, including multivariate Bernoulli, Multinomial and Gaussian mixture models, see for instance [Zhong and Ghosh, 2005, Gopal and Yang, 2014].

The vMF distribution is a probability distribution on a unit hypersphere and it belongs to the field of directional statistics [Mardia and Jupp, 2000]. In particular, it focuses on the directions of objects and measures the distance between them using cosine similarity. Most of earlier works using the vMF distribution focused on low dimensional data, i.e, using 2- or 3-dimensional vMF distributions [McLachlan and Peel, 2004], due to difficulties related to the estimation of the concentration parameter $\kappa$, that involves inversion of ratios of Bessel functions. In the context of clustering and for high dimensionality, Banerjee et al. [2005] proposed algorithms derived from a mixture of vMF distributions. They used an EM-based solution to estimate the parameters of their model and proposed an accurate approximation for estimating concentration parameter $\kappa$ for a high dimensional vMF distribution. Since this contribution, different vMF-based models for clustering high dimensional sparse data were proposed. For instance Reisinger et al. [2010] proposed a spherical topic model based on a mixture of vMF distributions, which is inspired from Latent Dirichlet Allocation (LDA). More recently, for clustering text data, Gopal and Yang [2014] proposed a Bayesian formulation and two vMF-based mixture models namely hierarchical and temporal variants.

In this paper, we propose a novel model based on the vMF distribution, for the analysis of high dimen-

Appearing in Proceedings of the $19^{th}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.

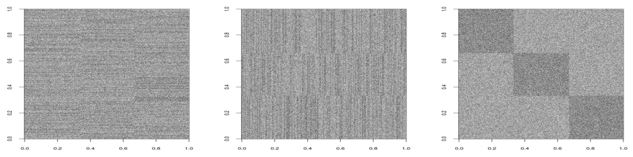


Figure 1: Left original data, Middle data reorganized according to row partition, Right data reorganized according to row and column partitions.

sional sparse data. Unlike existing vMF-based models, which focus only on clustering along one dimension, i.e, either row or column clustering, our model acts simultaneously on both dimensions of a data matrix. Intuitively, the model we propose can be viewed as an extension of the mixture of vMF distributions proposed by Banerjee et al. [2005] to the context of co-clustering or simultaneous clustering of rows and columns of a data matrix [Hartigan, 1972, Bock, 2003, Madeira and Oliveira, 2004, Van Mechelen et al., 2004, Banerjee et al., 2007, Rocci and Vichi, 2008, Wyse and Friel, 2012, Govaert and Nadif, 2013]. Specifically, our model seeks a diagonal co-clustering, meaning that rows and columns have the same number of clusters and that after a proper reorganisation of rows and columns we obtain a block diagonal structure, see Figure 1.

Setting our model under the maximum likelihood (ML) approach and the classification ML approach (CML), we derive two novel co-clustering algorithms a soft and hard variants, respectively. The proposed model exhibits several advantages over existing vMF-based models (i) it exploits the inherent duality between the rows and columns of a data matrix which improves clustering performances, (ii) by intertwining row clustering and column clustering at each stage, the derived algorithms perform an implicitly adaptive dimensionality reduction, which is imperative to handle high dimensional sparse matrices. Furthermore, in the case of large positives matrices, thanks to the dimensionality reduction, our formulation alleviates the problem of high concentration parameters $\kappa$ involved in Bessel functions that induces over and under flows, a well known difficulty in the classical vMF models [Banerjee et al., 2005]. (iii) Far from adding complexity, our model is more informative and produces meaningful and directly interpretable clusters.

**Notation.**

- Matrices are denoted with boldface uppercase letters, vectors with boldface lowercase letters and sets by script style uppercase letters. The $L_2$ norm is denoted by $\|.\|$. The $(d-1)$ dimensional unit sphere embedded in $\mathbb{R}^d$ is denoted by $\mathbb{S}^{d-1}$.
- Data is represented by a matrix $\mathbf{X} = (x_{ij})$ of size $n \times d$, $x_{ij} \in \mathbb{R}$, the $i^{th}$ row of this matrix is represented by a vector $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})^T$, where $T$ denotes the transpose.
- The partition of the set of rows $\mathcal{I}$ into $g$ clusters can be represented by a classification matrix $\mathbf{Z}$ of elements $z_{ih}$ in $\{0,1\}^g$ satisfying $\sum_{h=1}^{g} z_{ih} = 1$. The notation $\mathbf{z} = (z_1, \ldots, z_n)^T$, where $z_i \in \{1, \ldots, g\}$ represents the cluster label of $i$, is also used.
- Similarly the notations $\mathbf{W} = (w_{jh})$, $w_{jh} \in \{0,1\}^g$ satisfying $\sum_{h=1}^{g} w_{ih} = 1$, and $\mathbf{w} = (w_1, \ldots, w_d)$, where $w_j \in \{1, \ldots, g\}$ represents the cluster label of $j$, is used to represent the partition of the set of columns $\mathcal{J}$.
- In the same way, the fuzzy classification matrix of $\mathcal{I}$ is denoted by $\tilde{\mathbf{Z}} = (\tilde{z}_{ih})$. Where $\tilde{z}_{ih} \in [0,1]$, satisfying $\sum_{h=1}^{g} \tilde{z}_{ih} = 1$, for all $i$ in $\mathcal{I}$.

## 2 Mixture of von Mises-Fisher Distributions (movMFs)

Let $\mathbf{x}_i \in \mathbb{S}^{d-1}$, the von Mises-Fisher probability density function is given by

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

where $\boldsymbol{\mu}$ is the mean direction or centroid parameter and $\kappa$ denotes the concentration parameter, such that $\|\boldsymbol{\mu}\| = 1$ and $\kappa \geq 0$. The normalization term $c_d(\kappa)$ is equal to $c_d(\kappa) = \frac{\kappa^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\kappa)}$ where $I_r(\kappa)$ represents the modified Bessel function of the first kind and order $r$. In the vMF distribution the parameter $\kappa$ controls the concentration of data points $\mathbf{x}_i$ following (1), around the mean direction $\boldsymbol{\mu}$. For more details on vMF distribution, please refer to [Mardia and Jupp, 2000].

In the mixture model context, $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are supposed to be generated from a mixture of $g$ vMF distributions with a set of unknown parameters $\Theta$ [Banerjee et al., 2005]. The density function of this mixture takes the following form:

$$f(\mathbf{x}_i|\Theta) = \sum_h \alpha_h f_h(\mathbf{x}_i|\boldsymbol{\mu}_h, \kappa_h), \quad (2)$$

where $\Theta = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_g, \alpha_1, \ldots, \alpha_g, \kappa_1, \ldots, \kappa_g\}$, $\boldsymbol{\mu}_h$ and $\kappa_h$ represent the centroid and the concentration parameters of the $h^{th}$ component, respectively. Each parameter $\alpha_h$ denotes the proportion of points $\mathbf{x}_i$ generated from the $h^{th}$ component, such that $\sum_h \alpha_h = 1$ and $\alpha_h > 0$, $\forall h \in \{1, \ldots, g\}$. The complete data likelihood of the observed data is given by:

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

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

$$\begin{aligned} L_c(\Theta|\mathbf{X},\mathbf{Z}) &= \sum_h z_{.h} \log \alpha_h + \sum_h z_{.h} \log c_d(\kappa_h) \\ &+ \sum_{i,h} z_{ih} \kappa_h \boldsymbol{\mu}_h^T \mathbf{x}_i \end{aligned} \tag{4}$$

where $z_{.h}$ denotes the cardinality of the $h^{th}$ cluster. As the latent variable $\mathbf{z}$ is unknown in practice, Banerjee et al. [2005] proposed to use the EM algorithm [Dempster et al., 1977] to obtain the maximum likelihood estimates for the parameters $\Theta$. Thus, the E-step finds the conditional expectation of the missing variable $\mathbf{z}$ given the current estimated parameters $\Theta^{(t)}$ and the observed data, which is given by [Neal and Hinton, 1998]: $\tilde{z}_{ih} = \mathbb{E}(z_{ih} = 1|\mathbf{x}_i, \Theta^{(t)}) = \frac{\alpha_h^{(t)} f_h(\mathbf{x}_i|\boldsymbol{\mu}_h^{(t)},\kappa_h^{(t)})}{\sum_l \alpha_l^{(t)} f_l(\mathbf{x}_i|\boldsymbol{\mu}_l^{(t)},\kappa_l^{(t)})}$. The M-step finds the new parameters $\Theta^{(t+1)}$ by maximizing the expectation of the complete data log-likelihood (4) subject to the constraints $\sum_h \alpha_h = 1$, $\|\boldsymbol{\mu}_h\| = 1$ and $\kappa_h > 0$ which leads, for all $h$, to the update formulas described below:

$$\hat{\alpha}_h = \frac{\sum_i \tilde{z}_{ih}}{n}, \tag{5a}$$

$$\hat{\boldsymbol{\mu}}_h = \frac{\mathbf{r}_h}{\|\mathbf{r}_h\|} \quad \text{where} \quad \mathbf{r}_h = \sum_i \tilde{z}_{ih} \mathbf{x}_i \tag{5b}$$

$$\hat{\kappa}_h \approx \frac{\bar{r}_h d - \bar{r}_h^3}{1 - \bar{r}_h^2} \quad \text{where} \quad \bar{r}_h = \frac{I_{d/2}(\hat{\kappa}_h)}{I_{d/2-1}(\hat{\kappa}_h)} = \frac{\|\mathbf{r}_h\|}{\sum_i \tilde{z}_{ih}} \tag{5c}$$

Note that computing $\hat{\kappa}_h$ from the latter equation implies to inverse a ratio of Bessel functions, a problem for which there is no closed-form solution. For handling this difficulty, Banerjee et al. [2005] proposed the efficient approximation (5c), which deals with high dimensional datasets.

Alternating the above E and M steps leads to the soft-movMF algorithm proposed in [Banerjee et al., 2005]. Moreover, by setting the movMFs under the CML approach, Banerjee et al. [2005] derived the hard-movMF algorithm, which consists in using the classification variant of EM (CEM) [Celeux and Govaert, 1992] to estimate the model parameters.

# 3 Diagonal Block Mixture of von Mises-Fisher Distributions

In this section, we propose to extend the movMFs to the context of co-clustering [Hartigan, 1972]. Following the results of Dhillon and Modha [2001], which state that the unit centroids produced by the spherical k-means algorithm (a restricted version of both soft- and hard-movMF) are localized in the features space and tends towards orthonormality, we propose to capture and exploit this structure during the clustering process. More precisely, we assume some natural assumptions on the structure of centroids, i.e, orthonormality and homogeneity, at the beginning. Formally, we introduce a new parameter $\mathbf{w}$ (see Figure 2) that simultaneously guarantees the above assumptions and plays the role of a column partition. From a co-clustering point of view, this is equivalent to assume that rows and columns have the same number of clusters and that each column cluster is associated or describes a single row cluster. Which induces a block diagonal structure illustrated in Figure 1.

## 3.1 Definition

Instead of the classical movMFs, which partitions only the set of rows $\mathcal{I}$, our model called dbmovMFs partitions simultaneously the set of rows $\mathcal{I}$ and columns $\mathcal{J}$. Thereby it has the advantage of exploiting the duality between rows and columns of a data matrix. The density function of dbmovMFs is given by:

$$f(\mathbf{x}_i|\Theta) = \sum_h \alpha_h f_h(\mathbf{x}_i|\boldsymbol{\mu}_h^{\mathbf{w}}, \kappa_h^{\mathbf{w}}, \mathbf{w}), \tag{6}$$

where $\Theta$ is now formed by $\boldsymbol{\mu}_1^{\mathbf{w}}$,..., $\boldsymbol{\mu}_g^{\mathbf{w}}$, $\alpha_1, \ldots, \alpha_g$, $\kappa_1^{\mathbf{w}}, \ldots, \kappa_g^{\mathbf{w}}$ and the column partition $\mathbf{w}$, i.e, $w_j = h$ if the $j^{th}$ column belongs to $h^{th}$ column cluster, that is associated with $h^{th}$ row cluster. The superscript $\mathbf{w}$ is used to denote the fact that the centroid and the concentration parameters $\boldsymbol{\mu}_h^{\mathbf{w}}$, $\kappa_h^{\mathbf{w}}$, respectively, depend on the column partition $\mathbf{w}$. It follows from the orthonormality assumption that $\boldsymbol{\mu}_h^{\mathbf{w}}$ takes a "diagonal" form, i.e, $\mu_{hj} = 0$ if $w_{jh} = 0$, and from the homogeneity assumption follows that all non-zero entries of $\boldsymbol{\mu}_h^{\mathbf{w}}$ are equal, i.e, $\mu_{hj} = \mu_{hh}$, for all $j$ such as $w_{jh} = 1$. For instance, if we have a mixture of 3 vMF distributions, i.e, $g = 3$ and $h, k = 1, 2, 3$, each centroid $\boldsymbol{\mu}_h^{\mathbf{w}} \in \mathbb{S}^{d-1}$ takes this form: $\boldsymbol{\mu}_h^{\mathbf{w}} = (\mu_{h1}, \ldots, \mu_{h1}, \mu_{h2}, \ldots, \mu_{h2}, \mu_{h3}, \ldots, \mu_{h3})^T$ where $\mu_{hk}$ is repeated $w_{.h}$ times; $w_{.h}$ denotes the cardinality of the $h^{th}$ column cluster. In addition, these centroids are constrained by taking $\mu_{hk} = 0$ $\forall k \neq h$. This constraint leads to the orthonormality of centroid vectors.

Using the row and column classification matrices $\mathbf{Z}$ and $\mathbf{W}$, respectively, the complete data likelihood $\mathcal{L}(\mathbf{W}, \boldsymbol{\mu}, \boldsymbol{\alpha}, \kappa|\mathbf{X}, \mathbf{Z})$ takes the following form:

$$\prod_i \prod_h \left( \alpha_h c_d(\kappa_h) \times \prod_j (\exp^{\kappa_h \mu_{hh} x_{ij}})^{w_{jh}} \right)^{z_{ih}}$$

The corresponding complete data log-likelihood is

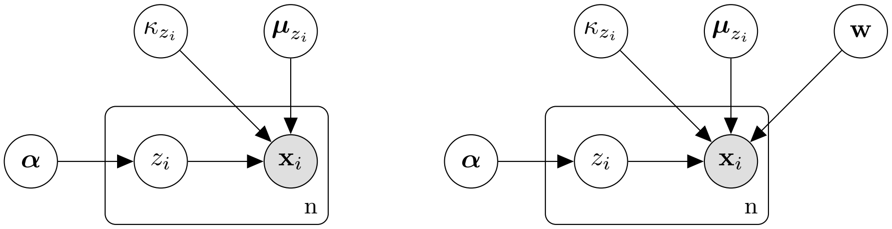


Figure 2: Graphical models of von-mises fisher mixture models, left movMFs, right dbmovMFs

given by:

$$\begin{aligned} L_c(\Theta|\mathbf{X},\mathbf{Z}) &= \sum_h z_{.h}\log\alpha_h + \sum_h z_{.h}\log(c_d(\kappa_h)) \\ &+ \sum_{i,h} z_{ih}\kappa_h\mu_{hh}\sum_j w_{jh}x_{ij} \\ &= \sum_h z_{.h}\log\alpha_h + \sum_h z_{.h}\log(c_d(\kappa_h)) \\ &+ \sum_{i,h} z_{ih}\kappa_h\mu_{hh}u_{ih} \qquad (7) \end{aligned}$$

where $u_{ih} = \sum_j w_{jh}x_{ij}$. This leads to

$$\begin{aligned} L_c(\Theta|\mathbf{X},\mathbf{Z}) &= \sum_h z_{.h}\log\alpha_h + \sum_h z_{.h}\log(c_d(\kappa_h)) \\ &+ \sum_{i,h} z_{ih}y_{ih} \qquad (8) \end{aligned}$$

where $y_{ih} = \kappa_h\mu_{hh}u_{ih}$, and in the same manner, we can give another expression of $L_c(\Theta|\mathbf{X},\mathbf{Z})$ in terms of column assignments as follows

$$\sum_h z_{.h}\log\alpha_h + \sum_h z_{.h}\log(c_d(\kappa_h)) + \sum_{j,h} w_{jh}t_{jh} \qquad (9)$$

where $t_{jh} = \kappa_h\mu_{hh}v_{hj}$, with $v_{hj} = \sum_i z_{ih}x_{ij}$.

### 3.2 Connection with other models

Assuming that the column partition $\mathbf{w}$ is fixed, the dbmovMFs model can be viewed as a classical movMFs [Banerjee et al., 2005] in which the mean directions vectors $\boldsymbol{\mu}_h$ are constrained to be of "diagonal" form introduced above.

Furthermore, connections with the Gaussian mixture model and the block Gaussian mixture model [Govaert and Nadif, 2013, Nadif and Govaert, 2010] can be established. More precisely, using the equivalence between the vMF and the Gaussian distributions [Mardia and Jupp, 2000] and assuming that $\mathbf{w}$ is fixed, it can be shown that dbmovMFs is equivalent to a mixture of Gaussian distributions of spherical form, i.e, the variance of the $h^{th}$ cluster is given by $\sigma_h^2 = \|\mathbf{m}_h^{\mathbf{w}}\|/\kappa_h$, $\mathbf{m}_h^{\mathbf{w}}$ is the centroid of the corresponding Gaussian component and $\boldsymbol{\mu}_h^{\mathbf{w}} = \mathbf{m}_h^{\mathbf{w}}/\|\mathbf{m}_h^{\mathbf{w}}\|$. The latter model, is also equivalent to a diagonal version of the block Gaussian mixture model [Govaert and Nadif, 2013], where each Gaussian component is parameterized by the variance $\sigma_h^2$ and mean vector $\mathbf{m}_h^{\mathbf{w}}$.

### 3.3 Maximum Likelihood estimates

To obtain the maximum likelihood estimates for the parameters $\Theta$, we use the generalized EM algorithm [Dempster et al., 1977, McLachlan and Krishnan, 2007]. The E-step is to compute the posterior probabilities $\tilde{z}_{ih} \propto \alpha_h f_h(\mathbf{x}_i|\Theta^{(t)})$. The M-step is obtained by maximizing or increasing the expectation of the complete data log-likelihood (8), subject to the constraints $\sum_h \alpha_h = 1$, $\|\boldsymbol{\mu}_h^{\mathbf{w}}\|^2 = \sum_j w_{jh}\mu_{hh}^2 = 1$, and $\kappa_h > 0$. We obtain the following update formulas:

$$\hat{w}_{jh} \leftarrow \begin{cases} 1, & \text{if } h = \arg\max_{h'} \tilde{t}_{jh'} \\ 0, & \text{otherwise.} \end{cases} \qquad (10a)$$

$$\hat{\alpha}_h = \frac{\sum_i \tilde{z}_{ih}}{n}, \qquad (10b)$$

$$\hat{\mu}_{hh} = \frac{r_h^{\mathbf{w}}}{\|\mathbf{r}_h^{\mathbf{w}}\|} = \pm\frac{1}{\sqrt{\sum_j \hat{w}_{jh}}} \quad \text{where}$$

$$r_h^{\mathbf{w}} = \sum_{i,j} \tilde{z}_{ih}\hat{w}_{jh}x_{ij} \qquad (10c)$$

$$\hat{\kappa}_h \approx \frac{\bar{r}_h^{\mathbf{w}} d - (\bar{r}_h^{\mathbf{w}})^3}{1 - (\bar{r}_h^{\mathbf{w}})^2} \quad \text{where}$$

$$\bar{r}_h^{\mathbf{w}} = \frac{I_{d/2}(\hat{\kappa}_h)}{I_{d/2-1}(\hat{\kappa}_h)} = \frac{\|\mathbf{r}_h^{\mathbf{w}}\|}{\sum_i \tilde{z}_{ih}\sum_j \hat{w}_{jh}} \qquad (10d)$$

where $\tilde{t}_{jh} = \kappa_h\mu_{hh}\tilde{v}_{hj}$, with $\tilde{v}_{hj} = \sum_i \tilde{z}_{ih}x_{ij}$, $\mathbf{r}_h^{\mathbf{w}}$ is a $d$-dimensional vector, such that $r_{hj}^{\mathbf{w}} = r_h^{\mathbf{w}}$ if $w_{jh} = 1$ and $r_{hj}^{\mathbf{w}} = 0$, otherwise. Alternating the above E and M steps leads to our soft-dbmovMF algorithm described in Algorithm 1.

Note that, unlike in the classical movMFs where it is easy to verify that $\bar{r}_h \leq 1$ (see equation 5c) given the definition of $\mathbf{r}$, it is not straightforward to verify that $\bar{r}_h^{\mathbf{w}} \leq 1$, without careful analysis. Such a result is imperative, to guarantee that the concentration parameters are positive, i.e, $\kappa_h > 0$ , $\forall h$, specially when using the approximation of equation (10). Hence, Proposition 1 provides theoretical guarantee about the fact

**Algorithm 1** soft-dbmovMF ($\mathrm{EM}_{\mathbf{b}}$).

```
**Input:** $\mathbf{X}$ ($\mathbf{x}_i \in \mathbb{S}^{d-1}$), $g$ the number of co-clusters.
**Output:** $\tilde{\mathbf{Z}}$ and $\mathbf{W}$,
**Steps:**
Initialization: $\Theta \leftarrow \Theta^{(0)}$;
**repeat**
  1. Expectation step of EM:
  **for** $i = 1$ **to** $n$ **do**
    **for** $h = 1$ **to** $g$ **do**
      $\tilde{z}_{ih} \leftarrow \frac{\alpha_h f_h(\mathbf{x}_i|\boldsymbol{\mu}_h,\kappa_h)}{\sum_l \alpha_l f_l(\mathbf{x}_i|\boldsymbol{\mu}_l,\kappa_l)}$
    **end for**
  **end for**
  2. Maximization step of EM:
  **for** $j = 1$ **to** $d$ **do**
    **for** $h = 1$ **to** $g$ **do**
      $\tilde{v}_{hj} \leftarrow \sum_i \tilde{z}_{ih} x_{ij}$; $\tilde{t}_{jh} \leftarrow \kappa_h \mu_{hh} \tilde{v}_{hj}$
    **end for**
    **for** $h = 1$ **to** $g$ **do**
      $\hat{w}_{jh} \leftarrow \begin{cases} 1, & \text{if } h = \arg\max_{h'} \tilde{t}_{jh'} \\ 0, & \text{otherwise.} \end{cases}$
    **end for**
  **end for**
  **for** $h = 1$ **to** $g$ **do**
    $\hat{\alpha}_h \leftarrow \frac{\sum_i \tilde{z}_{ih}}{n}$
    $\hat{\mu}_{hh} \leftarrow \pm \frac{1}{\sqrt{\sum_j \hat{w}_{jh}}}$; $r_h^{\mathbf{w}} \leftarrow \sum_{i,j} \tilde{z}_{ih} \hat{w}_{jh} x_{ij}$
    $\bar{r}_h^{\mathbf{w}} \leftarrow \frac{\|\mathbf{r}_h^{\mathbf{w}}\|}{\sum_i \tilde{z}_{ih} \sum_j \hat{w}_{jh}}$; $\hat{\kappa}_h \leftarrow \frac{\bar{r}_h^{\mathbf{w}} d - (\bar{r}_h^{\mathbf{w}})^3}{1 - (\bar{r}_h^{\mathbf{w}})^2}$
  **end for**
**until** convergence
```

that $0 \leq \bar{r}_h^{\mathbf{w}} \leq 1$, thereby it ensures that $\kappa$ estimated from equation (10) is always positive.

**Proposition 1** *Let $\mathbf{r}$ be a non-zero vector in $\mathbb{R}^d$ (i.e., $\mathbf{r} = (r_1, \ldots, r_d)^T$, such as $d \geq 2$) which results from a weighted sum of $n$ d-dimensional unit vector, i.e, $\mathbf{r} = \sum_i p_i \mathbf{x}_i$, $\mathbf{x}_i \in \mathbb{R}^d$ and $\|\mathbf{x}_i\| = 1$ , $\forall i \in \{1, \ldots, n\}$ , $n \geq 2$, the weights $p_i \geq 0$, $\forall i$. Let $\mathbf{r}^d$ be a vector in $\mathbb{R}^d$, such as all its components are equal to the sum of elements of $\mathbf{r}$ (i.e, $\mathbf{r}_1^d = \cdots = \mathbf{r}_d^d = \sum_{j=1}^d r_j$). Then $0 < \|\mathbf{r}^d\| \leq d \times \sum_i p_i$ with equality only if all unit vectors $\mathbf{x}_i$ are equal/collinear.*

The proof is available in supplementary material. By replacing $\mathbf{r}^d$, $d$ and $p_i$ in Proposition 1 with $\mathbf{r}_h^{\mathbf{w}}$, $\sum_j \hat{w}_{jh}$ and $\tilde{z}_{ih}$ respectively, it is easy to verify $0 \leq \bar{r}_h^{\mathbf{w}} \leq 1$.

### 3.4 Classification Maximum Likelihood estimates

Setting our model dbmovMFs under the CML approach, that consists in maximizing the classification likelihood instead of its expectation [Scott and Symons, 1971, Symons, 1981, Celeux and Govaert, 1992], we derive a hard version of soft-dbmovMF called $\mathrm{CEM}_{\mathbf{b}}$. It can be obtained from Algorithm 1, by incorporating a C-step between the E and M steps as follows $z_{ih} = 1$ if $h = \arg\max_{h'} \tilde{z}_{ih'}$ and $z_{ih} = 0$ otherwise, and replacing $\tilde{z}_{ih}$ by $z_{ih}$. The C-step of $\mathrm{CEM}_{\mathbf{b}}$, generates a completed sample ($\mathbf{x}_i$,$z_i$) by allocating each object $\mathbf{x}_i$ to the cluster $z_i$ that maximizes $\tilde{z}_{ih}$. The corresponding M-step can be deduced from the M-Step of $\mathrm{EM}_{\mathbf{b}}$ by replacing $\tilde{z}_{ih}$ by $z_{ih}$ and thereby $\tilde{t}_{jh}$ by $t_{jh}$.

Regarding the clustering context, the main difference between the ML and CML approaches is that, under the ML approach, the partition $\mathbf{z}$ of the set of objects into $g$ clusters is deduced at convergence of $\mathrm{EM}_{\mathbf{b}}$, by assigning each object $\mathbf{x}_i$ to the cluster that maximizes the *a posteriori* probability $\tilde{z}_{ih}$, while under the CML approach the clustering process is taken into account during parameters estimation. In this way, $\mathrm{CEM}_{\mathbf{b}}$ simultaneously estimates the parameters and the partition $\mathbf{z}$. It is well known that the ML approach yields more consistent estimate of the parameters thus often provides better clustering than the classification approach, especially when the clusters are not well separated. However, the CML approach exhibits some nice properties generated by $\mathrm{CEM}_{\mathbf{b}}$.

- $\mathrm{CEM}_{\mathbf{b}}$ is considerably more faster and scalable than $\mathrm{EM}_{\mathbf{b}}$, for instance, consider the update of the parameter $\hat{w}_{jh}$, under the ML approach (see, equation 10a) we need to go through all objects $\mathbf{x}_i$ to compute $\tilde{t}_{jh}$, while with $\mathrm{CEM}_{\mathbf{b}}$ we only go through objects whithin the $h^{th}$ cluster to compute $t_{jh}$.
- $\mathrm{CEM}_{\mathbf{b}}$ allows to avoid numerical difficulties, i.e, over and under flows, related to the computation of the conditional probabilities $\tilde{z}_{ih}$, especially in the case of the vMF distribution where the normalization terms $c_d(\kappa_h)$ involve Bessel functions and the concentration parameter $\kappa$ acts as multiplier in the exponent. More precisely, with $\mathrm{CEM}_{\mathbf{b}}$ there is no need for computing the conditional probabilities $\tilde{z}_{ih}$, the C-step can be done equivalently by assigning each object $\mathbf{x}_i$ to the cluster maximising $\log \alpha_h + \log f_h(\mathbf{x}_i|\Theta^{(t)})$. Furthermore, this contributes to the efficiency, scalability and memory-space saving of $\mathrm{CEM}_{\mathbf{b}}$.

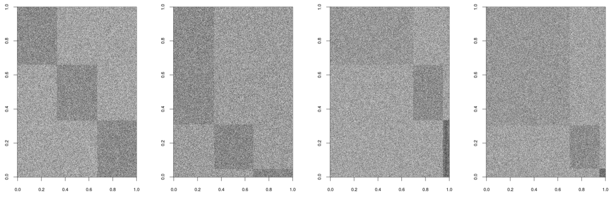
## 4 Experimental results

In this section, we shall provide extensive empirical results to validate and illustrate the benefits of our model dbmovMFs and the corresponding co-clustering algorithms $\mathrm{EM}_{\mathbf{b}}$ and $\mathrm{CEM}_{\mathbf{b}}$. To this end, we first propose to validate the correctness of our model on simulated datasets. In order to show the advantage of dbmovMFs over one sided movMFs and the ability of our algorithms to deal with high dimensionality and sparsity, we conduct extensive experiments on numerous real-world text datasets, in which we compare our algorithms against different vMF-based clustering methods denoted in our experiments as follows

- EM: the soft-movMF [Banerjee et al., 2005].
- CEM: the hard-movMF [Banerjee et al., 2005].
- Skmeans: the spherical k-means algorithm [Dhillon and Modha, 2001], it is a simplified version of CEM (hard-movMF), in which all clusters are assumed to have equal proportions and equal concentration parameters, i.e, $\kappa_h = \kappa$, $\alpha_h = \alpha$, $\forall h$.

### 4.1 Simulated data sets

To validate the correctness of our model and implementations, we evaluate our model on several simulated datasets corresponding to various particular situations, namely balanced co-clusters, unbalanced co-clusters, clusters with equal concentration and with different concentrations. Figure 3 illustrates these situations. We consider four different simulated datasets, each of them consists of a sample of 5000 unit vectors from a *Diagonal Block Mixture* of three 1000-dimensional vMF distributions. The different simulated datasets, i.e, sdata1,..., sdata4 are described in more details in Table 1. In this table, $\alpha$, $\kappa$ and $\boldsymbol{\mu}$ denote the true parameters while $\hat{\alpha}$, $\hat{\kappa}$ and $\hat{\boldsymbol{\mu}}$ denote their estimated. As it is evident from this table, both $\text{EM}_{\mathbf{b}}$ and $\text{CEM}_{\mathbf{b}}$ provide excellent performances even in situations where data exhibit unbalanced cluster sizes and concentration parameters. We also note that the estimations provided by $\text{EM}_{\mathbf{b}}$ are slightly better than those of $\text{CEM}_{\mathbf{b}}$ in almost all situations.



(a) (b) (c) (d)

Figure 3: Simulated datasets reorganized according to row and column partitions: (a) sdata1, (b) sdata2, (c) sdata3, (d) sdata4

### 4.2 Evaluation on real-world datasets

In the sequel, we aim at evaluating the performances of our algorithms on several real-world datasets. As a practical example we select the text mining domain and we concentrate on the challenging task of document clustering using high dimensional sparse document-term matrices. Hence, we selected six popular benchmark text datasets, CSTR used in [Li, 2005], CLASSIC4[1], WEBACE, the 20-newsgroups data NG20, SPORTS used in [Zhong and Ghosh, 2005] and TDT2[2]. All these datasets are carefully selected to represent various particular challenging situations in clustering: balanced clusters, unbalanced clusters, different number of clusters, i.e, raging from 4 to 30, different sizes, i.e, small and large datasets, different degrees of cluster overlap, i.e, well separated clusters and poorly separated clusters. The characteristics of the retained datasets are summarized in Table 2.

Table 2: Description of Datasets

| Datasets | Characteristics | | | | |
|---|---|---|---|---|---|
| | n | d | g | Sparsity (%) | Balance[3] |
| CSTR | 475 | 1000 | 4 | 96.60 | 0.399 |
| WEBACE | 2340 | 1000 | 20 | 91.83 | 0.169 |
| CLASSIC4 | 7094 | 5896 | 4 | 99.41 | 0.323 |
| NG20 | 19949 | 43586 | 20 | 99.99 | 0.991 |
| SPORTS | 8580 | 14870 | 7 | 99.14 | 0.036 |
| TDT2 | 9394 | 36771 | 30 | 99.64 | 0.028 |

Note that, in the context of text document clustering, it has been empirically shown on numerous real-world datasets, that the aforementioned baselines perform better than several existing clustering and co-clustering algorithms including: k-means with the euclidean distance, generative model using Gaussian, Bernoulli and Multinomial distributions, spectral co-clustering, and Latent Dirichlet Allocation (LDA). Therefore, we do not include these approaches in our comparisons. For more details see [Zhong and Ghosh, 2005, Gopal and Yang, 2014].

#### 4.2.1 Evaluation measures

Evaluating clustering results is not a trivial task, however, when the true category labels are known, a commonly used approach to validate clustering results consists in comparing the estimated partition with the true one. To this end, several measures have been proposed to asses the "similarity" between the estimated clustering and the true clustering, in our experiments we retain two widely used measures to asses the quality of clustering, namely the Normalized Mutual Information NMI [Strehl and Ghosh, 2003] and the Adjusted Rand Index ARI [Hubert and Arabie, 1985, Milligan and Cooper, 1986]. Further details can be found in the supplementary material.

#### 4.2.2 Performance comparison

In all our experiments we use the TF-IDF—proposed in Scikit-learn [Pedregosa et al., 2011]—normalized data representation. For each dataset we set $g$ to the real number of clusters, and in order for comparisons to be consistent, all algorithms are initialized using

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

[2] http://www.cad.zju.edu.cn/home/dengcai/

[3] The balance coefficient is the ratio of the number of documents in the smallest class to the number of documents in the largest class.

Table 1: Comparison of true and estimated parameters using $\mathrm{CEM}_{\mathbf{b}}$ and $\mathrm{EM}_{\mathbf{b}}$ on different simulated dataset, ($\alpha$,$\kappa$,$\boldsymbol{\mu}$) denote the true parameters while ($\hat{\alpha}$,$\hat{\kappa}$,$\hat{\boldsymbol{\mu}}$) denote estimated parameters

| Data | Components | True Parameters | | | Algorithms | Parameter Estimation | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $\alpha$ | $\kappa$ | $\mu_{hh}$ | | $\hat{\alpha}$ | $\lvert\alpha - \hat{\alpha}\rvert$ | $\hat{\kappa}$ | $\lvert\kappa - \hat{\kappa}\rvert$ | $\boldsymbol{\mu}^T\hat{\boldsymbol{\mu}}$ |
| sdata1 | cluster1 | 0.34 | 500.00 | $1/\sqrt{340}$ | $\mathrm{CEM}_{\mathbf{b}}$ | 0.339 | 0.001 | 501.38 | 1.38 | 1.00 |
| | | | | | $\mathrm{EM}_{\mathbf{b}}$ | 0.339 | 0.001 | 499.87 | 0.13 | 1.00 |
| | cluster2 | 0.33 | 500.00 | $1/\sqrt{330}$ | $\mathrm{CEM}_{\mathbf{b}}$ | 0.328 | 0.002 | 500.90 | 0.90 | 1.00 |
| | | | | | $\mathrm{EM}_{\mathbf{b}}$ | 0.329 | 0.001 | 499.33 | 0.67 | 1.00 |
| | cluster3 | 0.33 | 500.00 | $1/\sqrt{330}$ | $\mathrm{CEM}_{\mathbf{b}}$ | 0.333 | 0.003 | 500.56 | 0.56 | 1.00 |
| | | | | | $\mathrm{EM}_{\mathbf{b}}$ | 0.332 | 0.002 | 500.42 | 0.42 | 1.00 |
| sdata2 | cluster1 | 0.70 | 320.00 | $1/\sqrt{340}$ | $\mathrm{CEM}_{\mathbf{b}}$ | 0.691 | 0.009 | 320.72 | 0.72 | 1.00 |
| | | | | | $\mathrm{EM}_{\mathbf{b}}$ | 0.700 | 0.00 | 319.62 | 0.38 | 1.00 |
| | cluster2 | 0.25 | 400.00 | $1/\sqrt{330}$ | $\mathrm{CEM}_{\mathbf{b}}$ | 0.261 | 0.011 | 398.23 | 1.77 | 1.00 |
| | | | | | $\mathrm{EM}_{\mathbf{b}}$ | 0.250 | 0.00 | 401.51 | 1.51 | 1.00 |
| | cluster3 | 0.05 | 500.00 | $1/\sqrt{330}$ | $\mathrm{CEM}_{\mathbf{b}}$ | 0.048 | 0.002 | 497.75 | 2.25 | 1.00 |
| | | | | | $\mathrm{EM}_{\mathbf{b}}$ | 0.050 | 0.00 | 499.28 | 0.72 | 1.00 |
| sdata3 | cluster1 | 0.34 | 320.00 | $1/\sqrt{700}$ | $\mathrm{CEM}_{\mathbf{b}}$ | 0.345 | 0.005 | 319.82 | 0.18 | 0.998 |
| | | | | | $\mathrm{EM}_{\mathbf{b}}$ | 0.345 | 0.005 | 319.83 | 0.17 | 1.00 |
| | cluster2 | 0.33 | 400.00 | $1/\sqrt{250}$ | $\mathrm{CEM}_{\mathbf{b}}$ | 0.330 | 0.00 | 399.11 | 0.89 | 1.00 |
| | | | | | $\mathrm{EM}_{\mathbf{b}}$ | 0.330 | 0.00 | 399.30 | 0.70 | 1.00 |
| | cluster3 | 0.33 | 500.00 | $1/\sqrt{50}$ | $\mathrm{CEM}_{\mathbf{b}}$ | 0.325 | 0.005 | 487.82 | 12.18 | 0.980 |
| | | | | | $\mathrm{EM}_{\mathbf{b}}$ | 0.325 | 0.005 | 500.40 | 0.40 | 1.00 |
| sdata4 | cluster1 | 0.70 | 320.00 | $1/\sqrt{700}$ | $\mathrm{CEM}_{\mathbf{b}}$ | 0.697 | 0.003 | 320.20 | 0.20 | 1.00 |
| | | | | | $\mathrm{EM}_{\mathbf{b}}$ | 0.702 | 0.002 | 320.10 | 0.10 | 1.00 |
| | cluster2 | 0.250 | 400.00 | $1/\sqrt{250}$ | $\mathrm{CEM}_{\mathbf{b}}$ | 0.254 | 0.004 | 400.37 | 0.37 | 1.00 |
| | | | | | $\mathrm{EM}_{\mathbf{b}}$ | 0.248 | 0.002 | 399.77 | 0.23 | 1.00 |
| | cluster3 | 0.05 | 500.00 | $1/\sqrt{50}$ | $\mathrm{CEM}_{\mathbf{b}}$ | 0.049 | 0.001 | 498.10 | 1.90 | 1.00 |
| | | | | | $\mathrm{EM}_{\mathbf{b}}$ | 0.05 | 0.00 | 501.12 | 1.12 | 1.00 |

the same row partition that results from ten iterations of Skmeans started using a random initial point. For our algorithms, we further initialize randomly the column partition. Each algorithm is run until there is no significant increase of the likelihood with $\mathrm{EM}_{\mathbf{b}}$ and EM or the complete data likelihood with $\mathrm{CEM}_{\mathbf{b}}$ and CEM. Moreover, we conducted two sided paired t-tests between each algorithm and $\mathrm{EM}_{\mathbf{b}}$.

Tables 3 and 4, summarize the results of the different methods in terms of NMI and ARI, over all datasets. All results are averaged over thirty different starting points, obtained using the initialization strategy described above. Between brackets, we report the results corresponding to the trial with the highest criterion. As it is evident from these tables, both $\mathrm{EM}_{\mathbf{b}}$ and $\mathrm{CEM}_{\mathbf{b}}$, exhibit better performances as opposed to the other approaches. In fact, $\mathrm{EM}_{\mathbf{b}}$ and $\mathrm{CEM}_{\mathbf{b}}$ achieve the best performances, in almost all situations, except in terms of ARI on CLASSIC4 and NMI on SPORTS, as statistical tests state, however, the difference is not significant. Again, we note only a slight difference between $\mathrm{EM}_{\mathbf{b}}$ and $\mathrm{CEM}_{\mathbf{b}}$, which is statistically not significant. Moreover, both tables show that our algorithms provide high performances in terms of both NMI and ARI, while the other movMFs-based methods sometimes provide good NMI but low ARI as this is the case on WEBACE, SPORTS and TDT2. The reason is that, movMFs-based clustering methods have a tendency to merge small clusters and split larger ones into comparably sized clusters (see, Table 5), as it has been already emphasized in [Banerjee et al., 2005]. In fact, unlike ARI, the NMI measure is less sensitive to clusters merging and/or splitting. Our dbmovMFs-based algorithms, however, thanks to the centroids orthonormality assumption, avoid the above difficulty, and are able to discover large as well as small clusters. Thereby, dbmovMFs is more desirable when dealing with unbalanced clusters as this is the case with SPORTS, TDT2 and WEBACE datasets. For instance, on SPORTS, the balance of clusters produced by $\mathrm{EM}_{\mathbf{b}}$ (see, Table 5) is equal to 0.024 while the balance of clusters resulting from Skmeans is 0.453 which is far from the true balance coefficient of SPORTS.

Another notable result, is that the Skmeans algorithm which is based on a restricted version of movMFs where all clusters share the same proportion and same concentration parameter, yields better results than the other movMFs-based algorithms, as it has been already emphasized by Gopal and Yang [2014]. The low performance of EM and CEM as opposed to Skmeans, is due to high concentration $\kappa_h$'s in the normalization terms $c_d(\kappa_h)$'s that involve Bessel functions. As it has been highlighted by Banerjee et al. [2005], in the case of large positive matrices, all the data points lie on the first orthant of a $d$-dimensional hypersphere, thereby the concentration of such data points is implicitly high and increases exponentially with the dimensionality of the hypersphere. As a result the concentration parameters of vMF distributions are high and increase exponentially with the dimensionality of the data. Once again, dbmovMFs, thanks to its implicitly adaptive dimensionality reduction property, alleviates this issue. In Figure 4, we reported the distribution of the concentration parameters estimated by dbmovMFs-based algorithms and movMFs-based methods. We clearly observe that $\mathrm{EM}_{\mathbf{b}}$ and $\mathrm{CEM}_{\mathbf{b}}$ lead to substantially smaller concentration parameters than EM and CEM.

Table 3: Comparison of Average: NMI and ARI on small text datatsets. Significance results of statistical tests against $\mathbf{EM_b}$ are denoted by * for significance at 1% level and by † for significance at 5% level.

| | CSTR | | CLASSIC4 | | WEBACE | |
|---|---|---|---|---|---|---|
| | NMI | ARI | NMI | ARI | NMI | ARI |
| Skmeans | 0.732±0.026<br>(0.759)† | 0.772±0.025<br>(0.807)* | 0.591±0.020<br>(0.595)* | 0.468±0.011<br>(**0.476**) | 0.613±0.008<br>(0.620)† | 0.423±0.026<br>(0.394)† |
| CEM | 0.734±0.025<br>(0.759)† | 0.774±0.024<br>(0.807)* | 0.413±0.011<br>(0.410)* | 0.199±0.018<br>(0.194)* | 0.619±0.011<br>(0.623) | 0.398±0.021<br>(0.412)* |
| EM | 0.741±0.026<br>(0.768) | 0.777±0.026<br>(0.808)* | 0.406±0.013<br>(0.403)* | 0.190±0.015<br>(0.184)* | 0.614±0.014<br>(0.623) | 0.385±0.034<br>(0.397)* |
| $\mathrm{CEM}_{\mathbf{b}}$ | 0.754±0.024<br>(0.789) | 0.804±0.022<br>(0.830) | 0.660±0.003<br>(0.665) | 0.467±0.003<br>(0.473) | 0.623±0.011<br>(0.637) | 0.479±0.038<br>(0.519) |
| $\mathrm{EM}_{\mathbf{b}}$ | 0.754±0.022<br>(**0.792**) | 0.803±0.022<br>(**0.837**) | 0.660±0.002<br>(**0.668**) | 0.466±0.002<br>(0.473) | 0.624±0.008<br>(**0.639**) | 0.481±0.025<br>(**0.523**) |

Table 4: Comparison of Average: NMI and ARI on large datasets.

| | NG20 | | SPORTS | | TDT2 | |
|---|---|---|---|---|---|---|
| | NMI | ARI | NMI | ARI | NMI | ARI |
| Skmeans | 0.542±0.013<br>(0.555)* | 0.375±0.016<br>(0.379)† | 0.614±0.044<br>(**0.627**)* | 0.405±0.053<br>(0.442)* | 0.790±0.012<br>(0.801)† | 0.492±0.031<br>(0.514)* |
| CEM | 0.467±0.013<br>(0.484)* | 0.149±0.021<br>(0.150)* | 0.446±0.048<br>(0.455)* | 0.151±0.067<br>(0.177)* | 0.750±0.021<br>(0.769)* | 0.436±0.048<br>(0.466)* |
| EM | 0.465±0.013<br>(0.481)* | 0.143±0.020<br>(0.141)* | 0.444±0.049<br>(0.453)* | 0.149±0.067<br>(0.174)* | 0.751±0.019<br>(0.771)* | 0.438±0.041<br>(0.489)* |
| $\mathrm{CEM}_{\mathbf{b}}$ | 0.582±0.011<br>(0.594) | 0.388±0.024<br>(0.403) | 0.558±0.039<br>(0.608)† | 0.508±0.080<br>(0.602) | 0.799 ±0.014<br>(0.817) | 0.657±0.032<br>(**0.719**) |
| $\mathrm{EM}_{\mathbf{b}}$ | 0.585±0.010<br>(**0.601**) | 0.390±0.025<br>(**0.425**) | 0.564±0.037<br>(0.617) | 0.517±0.072<br>(**0.605**) | 0.799±0.015<br>(**0.821**) | 0.658±0.034<br>(0.699) |

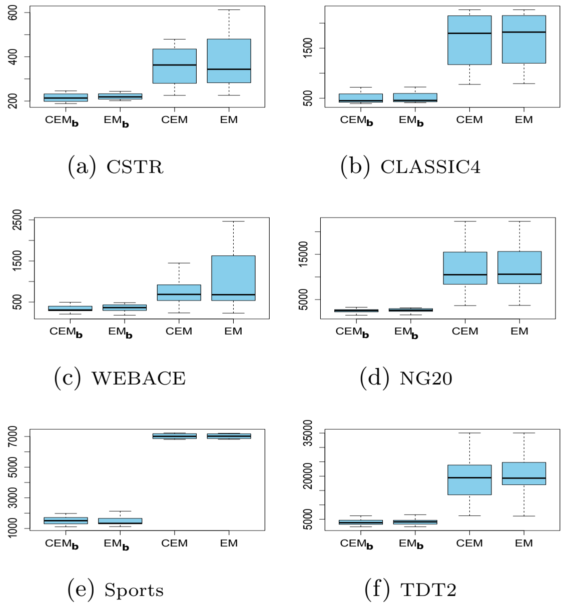
Table 5: Sports dataset: confusion matrices crossing the row clusters obtained by both algorithms (rows) and the true row clusters (columns). The column $z_{.h}$ indicates the cardinalities of clusters.

| | $\mathrm{EM}_{\mathbf{b}}$ (NMI = 0.617, ARI = 0.605) | | | | | | | | Skmeans (NMI = 0.627 , ARI = 0.442) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $z_{.h}$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $z_{.h}$ |
| 1 | 3338 | 219 | 41 | 27 | 487 | 48 | 82 | 4242 | 1705 | 1 | 0 | 1 | 1 | 0 | 0 | 1708 |
| 2 | 8 | 1008 | 0 | 0 | 16 | 0 | 0 | 1032 | 0 | 904 | 0 | 0 | 1 | 0 | 1 | 906 |
| 3 | 24 | 22 | 5 | 0 | 39 | 5 | 7 | 102 | 1 | 1 | 0 | 0 | 798 | 1 | 0 | 801 |
| 4 | 36 | 138 | 0 | 0 | 449 | 2 | 2 | 627 | 1326 | 0 | 0 | 0 | 6 | 0 | 0 | 1332 |
| 5 | 3 | 14 | 0 | 1 | 1345 | 1 | 10 | 1374 | 81 | 30 | 24 | 6 | 1171 | 5 | 4 | 1321 |
| 6 | 3 | 8 | 99 | 94 | 10 | 280 | 0 | 494 | 298 | 473 | 121 | 115 | 369 | 330 | 23 | 1729 |
| 7 | 0 | 1 | 0 | 0 | 0 | 0 | 708 | 709 | 1 | 1 | 0 | 0 | 0 | 0 | 781 | 783 |

## 5 Conclusion

We proposed dbmovMFs a novel generative mixture model based on the vMF distribution, for co-clustering high dimensional sparse data. Unlike existing vMF-based mixture models, which focus only on clustering along one dimension, dbmovMFs acts simultaneously on both dimensions of a data matrix. Thereby, dbmovMFs has the advantage of exploiting the clear duality between the rows and columns of a data matrix, which improves clustering performance. Setting dbmovMFs under the ML and CML approaches we derived two algorithms. Experiments conducted on numerous synthetic and real-world datasets, provide empirical evidence about the effectiveness of our algorithms and the benefits of dbmovMFs for modelling high dimensional sparse data. The proposed algorithms leads to substantially better performances than movMFs-based methods which are known to perform better than several existing methods, in the context of high dimensionality and sparsity. Moreover, thanks to the dimensionality reduction characteristic of our formulation, dbmovMFs alleviates the problem of high concentration parameters $\kappa_h$'s, a well known difficulty in vMF-based models.

The number of clusters remains one of the widely debated topics in clustering. In our context, this work is under investigation—due to the number of free parameters to be estimated—with information criteria such as BIC [Schwarz et al., 1978].



(a) CSTR

(b) CLASSIC4

(c) WEBACE

(d) NG20

(e) Sports

(f) TDT2

Figure 4: Distribution of concentration parameters.

## References

Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *J. Mach. Learn. Res.*, 6:1345–1382, 2005.

Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, Srujana Merugu, and Dharmendra S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximations. *J. Mach. Learn. Res.*, 8, 2007.

Hans-Hermann Bock. Convexity based clustering criteria: theory, algorithm and applications in statistics. *Statistical Methods and Applications*, 12:293–318, 2003.

Gilles Celeux and Gérard Govaert. A classification EM algorithm for clustering and two stochastic versions. *Computational statistics & Data analysis*, 14 (3):315–332, 1992.

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)*, pages 1–38, 1977.

Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Mach. Learn.*, 42(1-2):143–175, 2001.

Siddharth Gopal and Yiming Yang. von mises-fisher clustering models. In *Proceedings of the 31st ICML*, pages 154–162, 2014.

Gérard Govaert and Mohamed Nadif. *Co-Clustering*. John Wiley & Sons, 2013.

John A Hartigan. Direct clustering of a data matrix. *Journal of the american statistical association*, 67 (337):123–129, 1972.

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

Tao Li. A general model for clustering binary data. In *Proceedings of the eleventh ACM SIGKDD*, pages 188–197, 2005.

Sara C Madeira and Arlindo L Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM TCBB*, 1(1):24–45, 2004.

Kanti V Mardia and Peter E Jupp. *Directional statistics*. Wiley series in probability and statistics. Wiley, 2000.

Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

Glenn W Milligan and Martha C Cooper. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research*, 21(4):441–458, 1986.

Mohamed Nadif and Gérard Govaert. Model-based co-clustering for continuous data. In *ICMLA, 2010*, pages 175–180, 2010.

Radford Neal and Geoffrey E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12: 2825–2830, 2011.

Joseph Reisinger, Austin Waters, Bryan Silverthorn, and Raymond J Mooney. Spherical topic models. In *Proceedings of the 27th ICML*, pages 903–910, 2010.

Roberto Rocci and Maurizio Vichi. Two-mode multi-partitioning. *Computational Statistics & Data Analysis*, 52(4):1984–2003, 2008.

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

Allen J Scott and Michael J Symons. Clustering methods based on likelihood ratio criteria. *Biometrics*, pages 387–397, 1971.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

Michael J Symons. Clustering criteria and multivariate normal mixtures. *Biometrics*, pages 35–43, 1981.

Iven Van Mechelen, Hans-Hermann Bock, and Paul De Boeck. Two-mode clustering methods: a structured overview. *Statistical methods in medical research*, 13(5):363–394, 2004.

Jason Wyse and Nial Friel. Block clustering with collapsed latent block models. *Statistics and Computing*, 22(2):415–428, 2012.

Shi Zhong and Joydeep Ghosh. Generative model-based document clustering: a comparative study. *Knowledge and Information Systems*, 8(3):374–384, 2005.