Directional Clustering through Matrix Factorisation

Thomas Blumensath *Member, IEEE*

***Abstract*—This paper deals with a clustering problem where feature vectors are clustered depending on the angle between feature vectors, that is, feature vectors are grouped together if they point roughly in the same *direction*. This *directional* distance measure arises in several applications, including document classification and human brain imaging. Using ideas from the field of constrained low-rank matrix factorisation and sparse approximation, a novel approach is presented that differs from classical clustering methods, such as semi-Nonnegative Matrix Factorisation (semi-NMF), K-EVD or k-means clustering, yet combines some aspects of all of these. As in NMF and K-EVD, the matrix decomposition is iteratively refined to optimise a data fidelity term, however, no positivity constraint is enforced directly nor do we need to explicitly compute eigenvectors. As in k-means and K-EVD, each optimisation step is followed by a hard cluster assignment. This leads to an efficient algorithm that is here shown to outperform common competitors in terms of clustering performance and/or computation speed. In addition to a detailed theoretical analysis of some of the algorithm's main properties, the approach is evaluated empirically on a range of toy problems, several standard text clustering data-sets and a high dimensional problem in brain imaging, where functional MRI data is used to partition the human cerebral cortex into distinct functional regions.**

***Index Terms*—Clustering, Iterative Hard Thresholding, K-EVD, K-SVD, semi-NMF, k-means**

I. Introduction

Clustering [1], [2], has a long history in statistics and data analysis and a wide range of approaches have been proposed over the years, from generic algorithms to problem specific solutions. We are here interested in what we will call *directional clustering* problems. In directional clustering, a set of vectors is partitioned into several groups. Importantly, the distance used to group the vectors is the angle between the vectors, that is they are grouped depending on the direction into which they point, but the overall feature vector length does not influence the clustering result. The goal of directional clustering is thus to find a partition in which clusters are made up of vectors that roughly point in the same *direction*. This type of clustering is important in applications in which feature vectors are only given up to an unknown scaling. For example, in text clustering, feature vectors are often made up of the count of words in the document. Longer documents have naturally more words and individual words are likely to occur more often. As we want to compare documents in terms of content but not length, the overall length of the feature vectors is thus not important. Another example is functional human brain imaging using functional Magnetic Resonance Imaging data, which is the motivating application for the work reported here. In this application, time-series of brain activity are measured at different spatial locations within the brain and these are used as the feature vectors, however, due to the way in which neural function is measured, the scaling of these features is arbitrary so that if we want to group brain areas that have similar activity patterns, we need to ignore overall feature scaling. More formally, assume that we are given a set of M-dimensional feature vectors $\mathbf{x}_i$ that we want to group into disjoint sets, however, the features are only given up to an unknown positive scaling s_i. Due to this unknown scaling, in order to cluster the vectors $\mathbf{x}_i$, we can only rely on the direction of the vector, but not its length.

To derive our approach to solve this problem, we formulate the following optimisation problem, where features are modelled as a perturbed instance of an unknown scaled cluster centre $\mathbf{d}_k$

$$\mathbf{x}_i = s_{i,k}\mathbf{d}_k + \mathbf{e}_i, \tag{1}$$

where one could think of $\mathbf{e}_i$ as a 'noise' term. With this formulation, unsupervised clustering can be achieved by an estimation of the cluster centres $\mathbf{d}_k$ together with the assignment of each feature $\mathbf{x}_i$ to one of these centres. One way to achieve this would be based on traditional minimum mean square error (MMSE) estimation under appropriate constraints. The MMSE estimation problem can be formulated as

$$\min_{\{\mathbf{d}_k\},\{s_i\},\{\mathcal{C}_k\}} \sum_{k=1}^{K} \sum_{i\in\mathcal{C}_k} \|\mathbf{x}_i - s_{i,k}\mathbf{d}_k\|^2 \tag{2}$$

where we introduce the sets $\mathcal{C}_k$ which partition the feature vectors into the individual clusters, that is the sets $\mathcal{C}_k \subset [1,2,\ldots,N]$ are such that $\mathcal{C}_k \bigcap \mathcal{C}_{\hat{k}} = \emptyset$ for all $k \neq \hat{k}$ and $\bigcup \mathcal{C}_k = [1,2,\ldots,N]$. In words, we have to search over all partitions of the input feature vectors and over all possible vectors $\mathbf{d}_k$ and scalars s_i to optimise the distance between the cluster centres $\mathbf{d}_k$ and the feature vectors assigned to these centres. Our approach thus tries to find a cluster assignment, cluster centres and weights that directly minimise the euclidean error $\mathbf{e}_i$ in (1).

This formulation can be seen as a matrix factorisation problem [3]. Assume that the N column vectors $\mathbf{x}_i$ are stacked into a matrix $\mathbf{X}$. Let the K centres be stacked into a matrix $\mathbf{D}$ and let the errors $\mathbf{e}_i$ make up the columns of a matrix $\mathbf{E}$. With this notation, we can then write

$$\mathbf{X} = \mathbf{DS} + \mathbf{E}, \tag{3}$$

T. Blumensath is with the ISVR Signal Processing and Control Group, University of Southampton, SO17 1BJ, UK, Tel.: +44 (0) 23 8059 3224 ,e-mail: thomas.blumensath@soton.ac.uk

This research was supported by EPSRC grants EP/K037102/1 and EP/J005444/1 and through a Faculty of Engineering and the Environment funded New Frontiers Fellowship.

Data and code generated during this study are openly available from the University of Southampton repository at http://dx.doi.org/10.5258/SOTON/384519. Third party datasets are available following the links provided throughout the manuscript.

The author would like to thank the HCP consortium for the provision of the pre-processed fMRI data.

where $\mathbf{S}$ is a coefficient matrix, which, to be equivalent to the model in (1), will have to be a matrix with 1-sparse columns, that is each column is constrained to have a single non-zero entry.

We thus consider the following optimisation problem:

$$\min_{\mathbf{D},\mathbf{S}} \|\mathbf{X} - \mathbf{D}\mathbf{S}\|_F \quad : \mathbf{S} \text{ has } 1-\text{sparse columns}, \tag{4}$$

where the constraint on the sparsity of $\mathbf{S}$ enforces hard cluster assignment. For directional clustering, we assume that the columns of $\mathbf{X}$ are normalised to unit length[1].

A. Relationship to spherical k-means

Our generative model (1) is a generalisation of the classical approach used to deal with directional clustering, such as spherical k-means [4], [5], [6], which is based around a feature similarity measure

$$2 - 2\frac{\langle \mathbf{x}_i, \mathbf{d}_k \rangle}{\|\mathbf{x}_i\|\|\mathbf{d}_k\|} = \left\| \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} - \frac{\mathbf{d}_k}{\|\mathbf{d}_k\|} \right\|^2. \tag{5}$$

This cost function is typically optimised by scaling features and cluster centres to unit length as is done in spherical k-means. Using a probabilistic approach, (5) is proportional to the log-likelihood of the von Mieses-Fisher distribution. A von Mieses-Fisher mixture model has thus been used together with an Expectation Maximisation algorithm for directional clustering in [7].

Figure 1 highlights the main differences between the two cost functions. As can be seen, the cost function we propose is closely related to subspace clustering [8] and, in particular, to the one-dimensional subspace clustering problem. However, in our method, we do not try to find a subspace but a one-dimensional half-spaces defined through the vectors $\mathbf{d}_k$ as $s_k\mathbf{d}_k$ for $0 \leq s_k \leq \infty$. The difference is best understood in terms of a generative data model. Both cost functions model each feature vector as a perturbed scaled version of the cluster centre. However, the cost function on the right of figure 1 has a perturbation that is independent from the scaling of the cluster centre vector whilst the cost on the left has a perturbation that grows with the scaling. The model on the right can be understood as a model where each feature vector is a scaled version of a cluster centre vector with added uniform Gaussian noise. Obviously, if the scaled feature is very small, then the addition of the noise means that there is a lot of uncertainty in the angle of the original feature, whilst if the feature length is large, then there is less uncertainty and it is easier to cluster that feature.

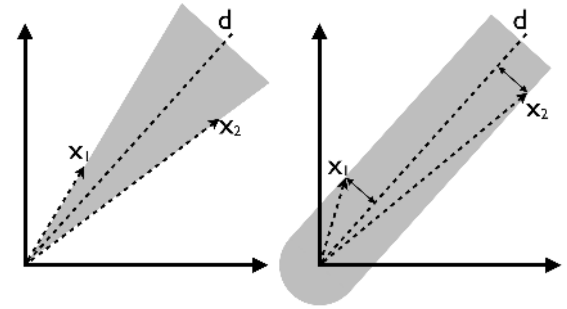

Fig. 1. Two vectors ($\mathbf{x}_1$ and $\mathbf{x}_2$) with the same distance from vector $\mathbf{d}$ as measured in terms of the angle (left) and in terms of our measure (right). The shaded regions indicate vectors with a smaller distance.

Importantly, our approach reduces to the spherical k-means cost function when we scale all features $\mathbf{x}_i$ to unit length. The advantage of our method however is that it allows us to incorporate additional prior information. This can be down by weighting certain features differently. In essence, by scaling $\mathbf{x}_i$, we simultaneously scale the noise term $c_i\mathbf{x}_i = s_i\mathbf{d}_k + c_i\mathbf{e}_i$, and, as our cost function penalises each feature equally, this allows us to compensate for known differences in noise variance.

B. Relationship to non-negative matrix factorisation

Formulating clustering problems as a matrix factorisation as in (3) is not new [3]. To estimate both $\mathbf{D}$ and $\mathbf{S}$ several constraints can be brought to bear, leading to different approaches. In many clustering problems, $\mathbf{S}$ can be constrained to be a nonnegative matrix in which case clustering becomes a semi-Nonnegative Matrix Factorisation (semi-NMF) problem [9]. Semi-NMF is a relaxation of more classical Nonnegative Matrix Factorisation (NMF)[10], [11]. In Semi-NMF, instead of constraining both $\mathbf{D}$ and $\mathbf{S}$ to be positive [12], only $\mathbf{S}$ is forced to be positive. Semi-NMF is used for clustering [13] by *first* computing the decomposition $\mathbf{DS}$ and *then*, in a second step, deriving a hard cluster assignment.

C. Relationship to subspace clustering

Sparsity constrained matrix factorisations are used in many Independent Component Analysis (ICA) methods [14], which are, for example, used in the fMRI literature for soft cluster assignments [15]. For hard sparsity constraints, such as those computed in the K-SVD algorithm and the related K-EVD [16], [17], this becomes equivalent to *subspace clustering* [17], [18].

Subspace clustering is an active research area. See for example [19] for a review of model-based approaches. Our approach is however more similar to K-SVD and K-EVD approaches, which, given a set of feature vectors, try to determine subspaces and cluster assignments such that the distance between features and their assigned subspaces is minimised. For one dimensional subspaces, both methods boil down to the same two steps.

1) For fixed subspaces, assign each feature to the subspace that is closest to the feature. If subspaces are defined as the lines $c_i\mathbf{d}_i$, where c_i can be both positive and negative, then the distance of feature $\mathbf{x}_k$ to the subspace is proportional to $1 - |\langle \mathbf{d}_i, \mathbf{x}_k \rangle| / (\|\mathbf{d}_i\|\|x_k\|)$.
2) For fixed cluster assignment, let $\mathbf{X}_i$ be the sub-matrix of $\mathbf{X}$ with feature vectors in cluster i, then the new subspace is defined through the vector $\mathbf{d}_i$ which is parallel to the singular vector (or eigenvector) associated

[1] Having features $\mathbf{x}_i$ to differ in length will giving more importance to certain features during clustering.

with the largest singular value of $\mathbf{X}_i$ (or eigenvalue of $\mathbf{X}_i^T\mathbf{X}_i$).

The above steps try to solve the one-dimensional subspace clustering problem. For directional clustering, step 1 would instead need to use the distance measure:

$$1 - \langle \mathbf{d}_i, \mathbf{x}_k \rangle / (\|\mathbf{d}_i\| \|x_k\|), \tag{6}$$

where the only difference is the missing $|\cdot|$ around the angle between features and directions. The other difference is that in the calculation of the new cluster centre, the direction has to be chosen carefully.

II. Our approach

We use an approach that iterates through three main steps: 1) Update of $\mathbf{D}$, 2) update of $\mathbf{S}$ without cluster assignment and 3) hard cluster assignment.

A. Updating $\mathbf{D}$, $\mathbf{S}$ and cluster assignment

For fixed $\mathbf{S}$, the optimal choice of $\mathbf{d}_k$ is calculated by minimising (4)

$$\mathbf{D} = \mathbf{X}\mathbf{S}^T(\mathbf{S}\mathbf{S}^T)^{-1}. \tag{7}$$

Note that $\mathbf{S}$ has 1-sparse columns, making the inversion trivial. If there are rows in which all entries are zero (i.e. we have empty cluster centres), we estimate all those columns in $\mathbf{D}$ for which there are non-zero rows in $\mathbf{S}$ whilst the remaining rows are re-instantiated. We here take an approach in which we set these columns randomly to elements from $\mathbf{X}$.

If we knew $\mathbf{D}_n$ and if we were to ignore the hard cluster assignment requirement, then the optimal $\mathbf{S}$ that solves (4) can be computed as $\mathbf{S}_{n+1} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{X}$ assuming that $(\mathbf{D}^T\mathbf{D})$ is invertible. To avoid matrix inversion, an alternative approach replaces the exact optimisation above with a single gradient step [20].

$$\mathbf{S}_{n+1} \quad = \quad \mathbf{S}_n + \mu_n \mathbf{D}_n^T(\mathbf{X} - \mathbf{D}_n\mathbf{S}_n), \tag{8}$$

where μ_n is a step size chosen appropriately (see below and the discussion in [21]).

To compute cluster assignment, we threshold $\mathbf{S}$, that is, we keeping only the largest entry of each column of $\mathbf{S}$, thus ensuring that $\mathbf{S}$ has 1-sparse columns. For simplicity, we write this non-linear operation as $\bar{\mathbf{S}}_n = P(\mathbf{S}_n)$, where the notation $\bar{\mathbf{S}}_n$ reminds us that this matrix has 1-sparse columns.

Problem (2) has several indeterminacies, which are common to most matrix factorisation problems. A re-scaling of $\mathbf{d}_k$ can always be counteracted by an appropriate inverse scaling of the associated $s_{i,k}$, a direct consequence of our desire that the cost is invariant to scaling. Due to this ambiguity, cluster assignment (i.e. the thresholding step), which is based on a comparison between the entries in $\mathbf{S}$, also faces ambiguities. To overcome this we use a re-scaling step that either normalises the columns in $\mathbf{D}$ after each update or normalises the rows in $\mathbf{S}$.

B. Algorithm

The algorithm is summarised below.

```
1) INPUT: data matrix $\mathbf{X}$, number of clusters $K$
2) initial decomposition of $\mathbf{X}$ into low rank factorisation
   (e.g. using an SVD or some initial cluster assignment)
   $\mathbf{X} = \mathbf{DS} + \mathbf{E}$.
3) iterate until the change in the cost (2) is small
    a) Calculate cluster assignment: $\bar{\mathbf{S}} = P(\mathbf{S})$
    b) Check for empty clusters and randomly re-initialise
    c) Update cluster centres: $\mathbf{D} = \mathbf{X}\bar{\mathbf{S}}^T(\bar{\mathbf{S}}\bar{\mathbf{S}}^T)^{-1}$ (and
       optionally normalise columns of $\mathbf{D}$)
    d) Update cluster weights: $\mathbf{S} = \mathbf{S} + \mu\mathbf{D}^T(\mathbf{X} - \mathbf{DS})$
       or $\mathbf{S} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{X}$ (optionally normalise $\mathbf{S}$)
```

C. Comparison to other approaches

Many matrix factorisation approaches, including semi-NMF, do not produce hard cluster assignment and so, a two stage approach is typically used in which the matrix decomposition is followed by a *single* cluster assignment step. This is in contrast to clustering approaches such as K-EVD and k-means, which make this assignment in each *iteration*. Our approach is similar in this respect to K-EVD. One important difference is, however, the way in which clusters are assigned. Instead of a k-means/k-EVD type approach that would pick the *closest* element, we first calculate a least squares type estimate of $\mathbf{S}$. We thus not only look at how close one feature is to one cluster centre, but take account of all cluster centres when assigning feature vectors. As a simple example that highlights this difference, assume we have three 2 dimensional cluster centres, $\mathbf{d}_1 = [1\ 0]^T$, $\mathbf{d}_2 = [0.999\ 0.1]^T$ and $\mathbf{d}_3 = [0.707\ 0.707]^T$ and a feature $\mathbf{x}_i = [0.9239\ 0.3827]^T$. The closest cluster centre in angle is actually $\mathbf{d}_2$, however, because $\mathbf{d}_1$ and $\mathbf{d}_2$ are very similar and $\mathbf{d}_3$ nearly orthogonal, if our method uses the pseudo inverse of $\mathbf{D}$ to calculate $\mathbf{S}$, then the feature will be assign to cluster $\mathbf{d}_3$. Our method thus has a build in mechanism that takes into account the fact that certain cluster centres are similar and that there is thus higher uncertainty in the assignment of a features to these centres as compared to more isolated centres which are more distinct.

III. Global minima, fixed points and convergence

We here concentrate on the analysis of one variant of the algorithm. Assume we use the steps

$$\mathbf{S}_{n+1/2} = \mathbf{S}_n + \mu_n \mathbf{D}_n^T(\mathbf{X} - \mathbf{D}_n\mathbf{S}_n), \tag{9}$$

$$\bar{\mathbf{S}}_{n+1/2} = P(\mathbf{S}_{n+1/2}) \tag{10}$$

and

$$\mathbf{D}_{n+1/2} = \mathbf{X}\bar{\mathbf{S}}_{n+1/2}^T(\bar{\mathbf{S}}_{n+1/2}\bar{\mathbf{S}}_{n+1/2}^T)^{-1} \tag{11}$$

and normalise $\mathbf{D}_{n+1/2}$ and $\bar{\mathbf{S}}_{n+1/2}$ after each update of $\mathbf{D}_{n+1/2}$. Let $\bar{\mathbf{S}}_{n+1}$ and $\mathbf{D}_{n+1}$ be the rescaled versions of $\mathbf{D}_{n+1/2}$ and $\bar{\mathbf{S}}_{n+1/2}$, such that $\mathbf{D}_{n+1}$ has unit norm columns and such that $\mathbf{D}_{n+1}\bar{\mathbf{S}}_{n+1} = \mathbf{D}_{n+1/2}\bar{\mathbf{S}}_{n+1/2}$.

A. Notation

In this section we will make use of the following notation.

- Let $\mathbf{s}_i$ be a column-vector containing the non-zero entries in $\bar{\mathbf{S}}$ for which the non-zero coefficient is in row i of $\bar{\mathbf{S}}$.
- Let $\mathbf{X}_k$ be the sub matrix of $\mathbf{X}$ containing those columns for which the columns in $\bar{\mathbf{S}}$ have a non-zero entry in row k.
- Let $\mathbf{\Phi}$ be a positive, diagonal matrix.
- Let q_i be the i^{th} diagonal element of the matrix $(\bar{\mathbf{S}}\bar{\mathbf{S}}^T)^{-1}$ and define p_i in the same way for matrix $\mathbf{\Phi}$. Note that $p_i = 1/(\|q_i\mathbf{X}_i\mathbf{s}_i\|)$, so that $q_i p_i = 1/\|\mathbf{X}_i\mathbf{s}_i\|$.
- Let $\mathbf{X}_i = \mathbf{U}_i\Sigma_i\mathbf{V}_i^T$ be the singular value decomposition (SVD) of $\mathbf{X}_i$, which is a sub-matrix of $\mathbf{X}$ containing those columns in $\mathbf{X}$ clustered into cluster i.
- Let $\mathbf{s}_i^T = \alpha_i^n\mathbf{V}_i^T$ be the expansion of the cluster coefficients $\mathbf{s}_i^T$ in the svd basis $\mathbf{V}_i$.
- For two matrices $\mathbf{A}$ and $\mathbf{B}$, we will use the inner product notation $\langle\mathbf{A},\mathbf{B}\rangle = \sum_{i,j} a_{i,j}b_{i,j}$, where the $a_{i,j}$ and $b_{i,j}$ are the elements in the i^{th} row and j^{th} column of $\mathbf{A}$ and $\mathbf{B}$ respectively. Note that this is the inner product that induces the Frobenius norm, making the space of matrices a Hilbert space.

With this notation, assume we have clustered $\mathbf{X}$ into some decomposition $\mathbf{DS}$, where $\mathbf{S}$ has one sparse columns. For the i^{th} cluster, the feature in that cluster are modelled with a single cluster centre, the i^{th} column in $\mathbf{D}$. This column is multiplied by all those elements in $\mathbf{S}$ that have a non-zero entry in row i. Thus, if $\mathbf{d}_i$ is the i^{th} column in $\mathbf{D}$, then the features in cluster i are approximated with scaled versions of $\mathbf{d}_i$, i.e. $\mathbf{X}_i \approx \mathbf{d}_i\mathbf{s}_i^T$. Furthermore, $\mathbf{d}_i$ itself is a function of $\mathbf{X}_i$ and $\mathbf{s}_i^T$, that is

$$\mathbf{d}_i = \frac{\mathbf{X}_i\mathbf{s}_i}{\mathbf{s}_i^T\mathbf{s}_i}, \tag{12}$$

or, if we normalise $\mathbf{d}_i$, then

$$\mathbf{d}_i = \frac{\mathbf{X}_i\mathbf{s}_i}{\|\mathbf{X}_i\mathbf{s}_i\|}, \tag{13}$$

However, as in the normalisation step, both $\mathbf{D}$ and $\mathbf{S}$ are scaled, the normalisation constant cancels in the product $\mathbf{d}_i\mathbf{s}_i^T$, which we thus write as

$$\mathbf{d}_i\mathbf{s}_i^T = \mathbf{X}_i\frac{\mathbf{s}_i\mathbf{s}_i^T}{\mathbf{s}_i^T\mathbf{s}_i}, \tag{14}$$

B. Summary of main results

We start with a characterisation of the global minimum of the clustering cost function. In fact, the minimum over $\mathbf{D}$ and $\mathbf{S}$ is found for some partition of $\mathbf{X}$ into sub matrices $\mathbf{X}_i$ such that the non-zero elements in $\mathbf{S}$, that is, the $\mathbf{s}_i^T$ are right singular vectors of the sub matrices $\mathbf{X}_i$ associated with the largest singular value. We then show that fixed points of the algorithm are also associated with $\mathbf{s}_i^T$ that are right singular vectors of the feature matrix $\mathbf{X}_i$.

We finally look at convergence and show that the algorithm converges to some cluster assignment, where cluster weights converge to the singular subspace of $\mathbf{X}_i$ associated with the largest singular value. This convergence depends on the choice of the step size μ.

C. The global minima

We start our analysis of the cost function by assuming that the cluster assignment, and thus the position of the non-zero elements in $\mathbf{S}$ is fixed. Under this condition, we have the following result.

Lemma 1. *For a fixed cluster assignment the minimal cost is achieved for $\mathbf{s}_i^T$ which are scaled versions of the right singular vector (or an element in the subspace spanned by the singular vectors) associated with the larges singular value(s) of $\mathbf{X}_i$.*

Proof: Using the notation above we note that the cost function

$$\|\mathbf{X} - \mathbf{D}_{n+1/2}\mathbf{S}_{n+1/2}\|_F^2 = \|\mathbf{X} - \mathbf{D}_{n+1}\mathbf{S}_{n+1}\|_F^2 \tag{15}$$

can be written as (dropping the iteration subscript n)

$$\sum_i \left\|\mathbf{X}_i - \mathbf{X}_i\frac{\mathbf{s}_i\mathbf{s}_i^T}{\mathbf{s}_i^T\mathbf{s}_i}\right\|_F^2. \tag{16}$$

Importantly, we recognise that $\frac{\mathbf{s}_i\mathbf{s}_i^T}{\mathbf{s}_i^T\mathbf{s}_i}$ is an orthogonal projection of the rows of $\mathbf{X}$ onto the one dimensional subspace spanned by $\mathbf{s}_i^T$, the minimum over all $\mathbf{s}_i^T$ is thus found if $\mathbf{s}_i^T$ lies in the subspace spanned by the right singular vectors of $\mathbf{X}_i$ associated with the largest singular values. ■

As there are only finitely many ways to assign features to clusters, we have thus proven the following result.

Theorem 2. *The global minima of the clustering cost function is achieved for $\mathbf{s}_i^T$ that lie in the subspace spanned by the right singular vectors of $\mathbf{X}_i$ associated with the largest singular values, where the $\mathbf{X}_i$ are non-empty sub-matrices of $\mathbf{X}$, such that each column in $\mathbf{X}$ is in exactly one sub-matrix.*

D. Stationary points

Let us next turn to the fixed points of the algorithm, that is, to an analysis of those $\mathbf{S}$ that satisfy the following condition

$$\mathbf{\Phi}\bar{\mathbf{S}} = P(\bar{\mathbf{S}} + \mu_n((\bar{\mathbf{S}}\bar{\mathbf{S}}^T)^{-1}\bar{\mathbf{S}}\mathbf{X}^T\,(\mathbf{X} - \mathbf{X}\bar{\mathbf{S}}^T(\bar{\mathbf{S}}\bar{\mathbf{S}}^T)^{-1}\bar{\mathbf{S}}), \tag{17}$$

where $\mathbf{\Phi}$ is a diagonal matrix (a function of $\bar{\mathbf{S}}$) that normalises the columns of the matrix $\mathbf{X}\bar{\mathbf{S}}^T(\bar{\mathbf{S}}\bar{\mathbf{S}}^T)^{-1}$. Note that $\bar{\mathbf{S}}\bar{\mathbf{S}}^T$ is diagonal and so is $\mathbf{\Phi}$. Because $\bar{\mathbf{S}}$ is one-column sparse, it is again instructive to re-write the above condition in terms of the vectors $\mathbf{s}_i^T$.

We then have the following stationarity condition

$$\mu\frac{\mathbf{s}_i^T\mathbf{X}_i^T}{\|\mathbf{X}_i\mathbf{s}_i\|}\mathbf{X}_k(\mathbf{I} - \frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{s}_k^T\mathbf{s}_k})\begin{cases} = p_i\mathbf{s}_i^T, & \text{if } i = k \\ < (1+p_i)\mathbf{s}_i^T, & \text{otherwise.}\end{cases} \tag{18}$$

Here, the p_i are scalars (the diagonal elements in $\mathbf{\Phi}$), $P_{\mathbf{s}_k^T} = \frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{s}_k^T\mathbf{s}_k}$ is a projection and $P_{\mathbf{s}_k^T}^{\perp} = (\mathbf{I} - \frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{s}_k^T\mathbf{s}_k})$ its orthogonal complement. The strict inequality is due to the fact that at a stationary points, the thresholding operation must set all $\mathbf{s}_i$ to zero apart from the largest one, which implies that before thresholding, $\mathbf{s}_i + \mu\frac{\mathbf{s}_i^T\mathbf{X}_i^T}{\|\mathbf{X}_i\mathbf{s}_i\|}\mathbf{X}_k(\mathbf{I} - \frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{s}_k^T\mathbf{s}_k})$ must be smaller than $p_k\mathbf{s}_k$ for $k \neq i$.

Because $(\mathbf{I} - \frac{\mathbf{s}_i\mathbf{s}_i^T}{\mathbf{s}_i^T\mathbf{s}_i})$ is a projection onto the row space orthogonal to $\mathbf{s}_i^T$, the row vectors of $X_k(\mathbf{I} - \frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{s}_k^T\mathbf{s}_k})$ are orthogonal to $\mathbf{s}_i^T$. Thus, $\frac{\mathbf{s}_i^T \mathbf{X}_i^T}{\|\mathbf{X}_i \mathbf{s}_i\|} \mathbf{X}_k (\mathbf{I} - \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{s}_k})$ is a sum over vectors that are orthogonal to $\mathbf{s}_i^T$, which implies that the constant p_i above has to be zero.

We have thus shown the following.

Lemma 3. *The stationary points* $\mathbf{S}$ *satisfy the following condition*

$$\mu \frac{\mathbf{s}_i^T \mathbf{X}_i^T}{\|\mathbf{X}_i \mathbf{s}_i\|} \mathbf{X}_k (\mathbf{I} - \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{s}_k}) \begin{cases} = 0, & \textit{if } i = k \\ < \mathbf{s}_i^T, & \textit{otherwise.} \end{cases} \tag{19}$$

To get an even better understanding of the fixed point condition above, let us write $\mathbf{X}_i^{\mathbf{s}_i^T} = \mathbf{X}_i \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{s}_k}$ and let $\mathbf{E}_i = \mathbf{X}_i - \mathbf{X}_i^{\mathbf{s}_i^T}$. Note that $\mathbf{X}_i^{\mathbf{s}_i^T}$ and $\mathbf{E}_i$ have orthogonal rows. Thus, the above lemma shows that $\mathbf{s}_i^T$ is a fixed point if and only if

$$\langle \mathbf{X}_i \mathbf{s}_i, \mathbf{E}_i \rangle = 0 \tag{20}$$

and

$$\langle \mathbf{X}_i \mathbf{s}_i, \mathbf{E}_k \rangle < \mathbf{s}_i^T, \tag{21}$$

where the inequality must hold element wise and for all $k \neq i$. Importantly, the first equality above can also be stated as

$$\mathbf{s}_i^T \mathbf{X}_i^T \mathbf{X}_i = \gamma_i \mathbf{s}_i^T, \tag{22}$$

for some c_i. As this is a typical eigenvalue problem we have proven the following lemma

Lemma 4. *The stationary points of the algorithm provide a partition of the data set such that the non-zero elements in* $\mathbf{S}$ *associated with cluster* i *are eigenvectors of the matrix* $\mathbf{X}_i^T \mathbf{X}_i$.

E. Convergence, preliminary results

To derive convergence results for the algorithm, we first derive a range of results that show the convergence of several related quantities. We first show that our algorithm is optimising the following majorized cost function, which is optimised under the constraint that the columns of the solution have to be 1 column sparse:

$$\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{D}_n \mathbf{A}\|_F^2 + \frac{1}{\mu_n} \|\bar{\mathbf{S}}_n - \mathbf{A}\|_F^2 - \|\mathbf{D}_n(\bar{\mathbf{S}}_n - \mathbf{A})\|_F^2, \tag{23}$$

where the minimisation is done over all matrices $\mathbf{A}$ that have 1-sparse columns. The argument for this basically follows that in [22]. We can re-write this as

$$\begin{aligned} & \min\nolimits_{\mathbf{A}} && \|\mathbf{X} - \mathbf{D}_n \mathbf{A}\|_F^2 + \frac{1}{\mu_n} \|\bar{\mathbf{S}}_n - \mathbf{A}\|_F^2 \\ & && -\|\mathbf{D}_n(\bar{\mathbf{S}}_n - \mathbf{A})\|_F^2, \\ = & \min\nolimits_{\mathbf{A}} && \|\mathbf{X}\|_F^2 + \|\mathbf{D}_n \mathbf{A}\|_F^2 - 2\langle \mathbf{D}_n^T \mathbf{X}, \mathbf{A} \rangle \\ & && + \frac{1}{\mu_n} \|\bar{\mathbf{S}}_n\|_F^2 + \frac{1}{\mu_n} \|\mathbf{A}\|_F^2 - \frac{1}{\mu_n} 2\langle \bar{\mathbf{S}}_n, \mathbf{A} \rangle \\ & && -\|\mathbf{D}_n \bar{\mathbf{S}}_n\|_F^2 - \|\mathbf{D}_n \mathbf{A}\|_F^2 + 2\langle \mathbf{D}_n^T \mathbf{D}_n \bar{\mathbf{S}}_n, \mathbf{A} \rangle \\ = & \min\nolimits_{\mathbf{A}} && -2\langle \mathbf{D}_n^T \mathbf{X}, \mathbf{A} \rangle + \frac{1}{\mu_n} \langle \mathbf{A}, \mathbf{A} \rangle - \frac{1}{\mu_n} 2\langle \bar{\mathbf{S}}_n, \mathbf{A} \rangle + \\ & && 2\langle^T \mathbf{D}_n^T \mathbf{D}_n \bar{\mathbf{S}}_n, \mathbf{A} \rangle \end{aligned}$$

Thus, we need to minimise

$$\left\langle \frac{1}{\mu_n} \mathbf{A} - \frac{2}{\mu_n} \bar{\mathbf{S}}_n - 2\mathbf{D}_n^T (\mathbf{X} - \mathbf{D}_n \bar{\mathbf{S}}_n), \mathbf{A} \right\rangle.$$

Taking derivatives w.r.t. the elements in $\mathbf{A}$ and setting to zero, we re-derive our update equation (8)

$$\mathbf{A} = \bar{\mathbf{S}}_n + \mu \mathbf{D}_n^T (\mathbf{X} - \mathbf{D}_n \bar{\mathbf{S}}_n), \tag{24}$$

which, to impose the sparsity constraint on the columns of $\mathbf{A}$ has to be thresholded appropriately.

For the majorised cost function to bound the original clustering cost function, we need to choose μ_n such that for all $\mathbf{A}$ with one sparse columns, the majorisation term

$$\frac{1}{\mu_n} \|\bar{\mathbf{S}}_n - \mathbf{A}\|_F^2 - \|\mathbf{D}_n(\bar{\mathbf{S}}_n - \mathbf{A})\|_F^2 > 1/c \|\bar{\mathbf{S}}_n - \mathbf{A}\|_F^2 \tag{25}$$

for some constant $c > 0$ independent of n. As the columns of $\mathbf{D}_n$ are normalised and as columns in $(\bar{\mathbf{S}}_n - \mathbf{A})$ are two sparse, $\|\mathbf{D}_n(\bar{\mathbf{S}}_n - \mathbf{A})\|^2 \leq 4\|(\bar{\mathbf{S}}_n - \mathbf{A})\|^2$ so we can choose $\mu < 1/4$. In fact, equality only holds if there are two columns in $\mathbf{D}_n$ that are equal in which case we can combine these two clusters and re-initialise the empty cluster. W.l.g we can thus assume that $\mu = 0.25$. Also note that, if cluster assignment does not change between iterations, then $\bar{\mathbf{S}}_n - \mathbf{A}$ will have one sparse columns, in which case we can choose $\mu < 1$. This suggests a line search approach as suggested in [21]. Where we initially try $\mu < 1$, which is used as long as the cluster assignment does not change, but if it leads to a changing cluster assignment, we instead use $\mu = 0.25$.

Under this condition on μ, $\bar{\mathbf{S}}_{n+1/2}$ satisfies

$$\begin{aligned} & && \|\mathbf{X} - \mathbf{D}_{n+1} \bar{\mathbf{S}}_{n+1}\|_F^2 = \|\mathbf{X} - \mathbf{D}_{n+1/2} \bar{\mathbf{S}}_{n+1/2}\|_F^2 \\ & \leq && \|\mathbf{X} - \mathbf{D}_n \bar{\mathbf{S}}_{n+1/2}\|_F^2 \\ & \leq && \|\mathbf{X} - \mathbf{D}_n \bar{\mathbf{S}}_{n+1/2}\|_F^2 + \frac{1}{\mu_n} \|\bar{\mathbf{S}}_n - \bar{\mathbf{S}}_{n+1/2}\|_F^2 \\ & && -\|\mathbf{D}_n(\bar{\mathbf{S}}_n - \bar{\mathbf{S}}_{n+1/2})\|_F^2 \\ & \leq && \|\mathbf{X} - \mathbf{D}_n \bar{\mathbf{S}}_n\|_F^2. \end{aligned} \tag{26}$$

so that

$$\begin{aligned} & \frac{1}{\mu_n} \|\bar{\mathbf{S}}_n - \bar{\mathbf{S}}_{n+1/2}\|_F^2 - \|\mathbf{D}_n(\bar{\mathbf{S}}_n - \bar{\mathbf{S}}_{n+1/2})\|_F^2 \\ & \qquad \leq \|\mathbf{X} - \mathbf{D}_n \mathbf{S}_n\|^2 - \|\mathbf{X} - \mathbf{D}_{n+1} \bar{\mathbf{S}}_{n+1}\|_F^2 \end{aligned} \tag{27}$$

where we used the minimality of $\mathbf{D}_{n+1/2}$ and the fact that $\mathbf{D}_{n+1} \bar{\mathbf{S}}_{n+1} = \mathbf{D}_{n+1/2} \bar{\mathbf{S}}_{n+1/2}$. This shows that our algorithm reduces the cost function in each iteration

$$\|\mathbf{X} - \mathbf{D}_{n+1} \bar{\mathbf{S}}_{n+1}\|_F^2 \leq \|\mathbf{X} - \mathbf{D}_n \mathbf{S}_n\|_F^2. \tag{28}$$

Thus, the sequence $\mathbf{X} - \mathbf{D}_{n+1} \bar{\mathbf{S}}_{n+1}$ is bounded and thus by the Bolzano-Weierstrass theorem will have a convergent subsequence. Note that boundedness holds also if we re-initialise empty clusters in step b) of the algorithm, as long as we do this as discussed above. Because $\mathbf{X}$ is fixed, boundedness of $\mathbf{X} - \mathbf{D}_{n+1} \bar{\mathbf{S}}_{n+1}$ also implies boundedness of $\mathbf{D}_{n+1} \bar{\mathbf{S}}_{n+1}$, i.e.

$$\begin{aligned} \|\mathbf{X}\|_F + M & \geq && \|\mathbf{X}\|_F + \|\mathbf{X} - \mathbf{D}_{n+1} \bar{\mathbf{S}}_{n+1}\|_F \\ & \geq && \|\mathbf{D}_{n+1} \bar{\mathbf{S}}_{n+1}\|_F \\ & = && \|\mathbf{D}_{n+1/2} \bar{\mathbf{S}}_{n+1/2}\|_F \\ & = && \|\mathbf{X} \bar{\mathbf{S}}_{n+1/2}^T (\bar{\mathbf{S}}_{n+1/2} \bar{\mathbf{S}}_{n+1/2}^T)^{-1} \bar{\mathbf{S}}_{n+1/2}\|_F. \end{aligned}$$

Note that the last line also implies boundedness, as $\bar{\mathbf{S}}^T_{n+1/2}(\bar{\mathbf{S}}_{n+1/2}\bar{\mathbf{S}}^T_{n+1/2})^{-1}\bar{\mathbf{S}}_{n+1/2}$ is a projection operator projecting the rows of $\mathbf{X}$ (the inverse always exists by construction). Thus

$$\|\mathbf{D}_{n+1/2}\bar{\mathbf{S}}_{n+1/2}\|_F \leq \|\mathbf{X}\|_F \tag{29}$$

and

$$\begin{aligned} &\|\mathbf{X} - \mathbf{D}_{n+1}\bar{\mathbf{S}}_{n+1}\|_F \\ &= \|\mathbf{X}(\mathbf{I} - \bar{\mathbf{S}}^T_{n+1/2}(\bar{\mathbf{S}}_{n+1/2}\bar{\mathbf{S}}^T_{n+1/2})^{-1}\bar{\mathbf{S}}_{n+1/2})\|_F \\ &\leq \|\mathbf{X}\|_F \end{aligned} \tag{30}$$

Thus, using the Bolzano-Weierstrass theorem, we have proven the following lemma.

Lemma 5. *There exist an $\mathbf{X}^\star$ such that for all ϵ, we can choose an $N_\epsilon < \infty$ such that*

$$\|\mathbf{X}^\star - \mathbf{D}_{n_i}\bar{\mathbf{S}}_{n_i}\|_F \leq \epsilon, \tag{31}$$

hold for infinitely many $n_i > N_\epsilon$.

Assume the accumulation point $\mathbf{X}^\star$ in the above lemma is unique, that is, for all ϵ in the above lemma, let n_j be the indices such that $\|\mathbf{X}^\star - \mathbf{D}_{n_j}\bar{\mathbf{S}}_{n_j}\| \geq \epsilon$. If the set n_j is finite, then there will be a maximal n_j and we can choose $N_\epsilon > n_j$ and find that for all $n > N_\epsilon$ $\|\mathbf{X}^\star - \mathbf{D}_n\bar{\mathbf{S}}_n\| \leq \epsilon$. This implies convergence of $\mathbf{D}_n\bar{\mathbf{S}}_n$ to $\mathbf{X}^\star$. Thus, either $\mathbf{D}_n\bar{\mathbf{S}}_n$ converges or there are at least two accumulation points.

We can also establish the following lemma.

Lemma 6. *Assume that μ_n is chosen such that*

$$\frac{1}{\mu_n}\|\mathbf{S}_n - \bar{\mathbf{S}}_{n+1/2}\|^2 - \|\mathbf{D}_n(\mathbf{S}_n - \bar{\mathbf{S}}_{n+1/2})\|^2 > \frac{1}{c}\|\mathbf{S}_n - \bar{\mathbf{S}}_{n+1/2}\|^2 \tag{32}$$

for some positive constant c. The matrix factorisation algorithm then produces a sequence of estimates $\bar{\mathbf{S}}_n$ that satisfy: $\|\bar{\mathbf{S}}_{n+1/2} - \bar{\mathbf{S}}_n\|^2 \to 0$.

Furthermore, the sum $\sum_{n=1}^N \|\bar{\mathbf{S}}_{n+1/2} - \bar{\mathbf{S}}_n\|^2$ converges and thus, by the Cauchy's Convergence Criterion, so do the partial sums $\sum_{n=N}^{N+p} \|\bar{\mathbf{S}}_{n+1/2} - \bar{\mathbf{S}}_n\|^2$, where $p \geq 1$ is arbitrary.

Proof: *Convergence follows from the fact that the series $\sum_{n=1}^N \|\bar{\mathbf{S}}_{n+1/2} - \bar{\mathbf{S}}_n\|^2$ is monotonically increasing and bounded. Monotonicity is obvious, to show boundedness, write*

$$\begin{aligned} & \sum_{n=1}^N \|\bar{\mathbf{S}}_{n+1/2} - \bar{\mathbf{S}}_n\|^2 \\ \leq\ & c\sum_{n=1}^N \frac{1}{\mu}\|\bar{\mathbf{S}}_{n+1/2} - \bar{\mathbf{S}}_n\|^2 - \|\mathbf{D}_n(\bar{\mathbf{S}}_{n+1/2} - \bar{\mathbf{S}}_n)\|^2 \\ \leq\ & c\sum_{n=1}^N \left(\|\mathbf{X} - \mathbf{D}_n\bar{\mathbf{S}}_n\|^2 - \|\mathbf{X} - \mathbf{D}_n\bar{\mathbf{S}}_{n+1}\|^2\right) \\ \leq\ & c\sum_{n=1}^N \left(\|\mathbf{X} - \mathbf{D}_n\bar{\mathbf{S}}_n\|^2 - \|\mathbf{X} - \mathbf{D}_{n+1}\bar{\mathbf{S}}_{n+1}\|^2\right) \\ =\ & c\left(\|\mathbf{X} - \mathbf{D}_1\bar{\mathbf{S}}_1\|^2 - \|\mathbf{X} - \mathbf{D}_{N+1}\bar{\mathbf{S}}_{N+1}\|^2\right) \\ \leq\ & c\|\mathbf{X} - \mathbf{D}_1\bar{\mathbf{S}}_1\|^2 \end{aligned} \tag{33}$$

where the first inequality is due to the choice of μ_n, the second inequality is (26) *and where the third inequality is due to the optimality of $\mathbf{D}_{n+1}$ (i.e. $\|\mathbf{X} - \mathbf{D}_{n+1}\bar{\mathbf{S}}_{n+1}\|^2 \leq \|\mathbf{X} - \mathbf{D}_n\bar{\mathbf{S}}_{n+1}\|^2$).*

■

Lemma 7. *Assume that μ_n is chosen such that*

$$\frac{1}{\mu_n}\|\mathbf{S}_n - \bar{\mathbf{S}}_{n+1/2}\|^2 - \|\mathbf{D}_n(\mathbf{S}_n - \bar{\mathbf{S}}_{n+1/2})\|^2 > \frac{1}{c}\|\mathbf{S}_n - \bar{\mathbf{S}}_{n+1/2}\|^2 \tag{34}$$

for some positive constant c. Assume there are no empty clusters in $\bar{\mathbf{S}}^n$. The matrix factorisation algorithm then produces a sequence of estimates $\mathbf{D}_n\bar{\mathbf{S}}_n$ that satisfy:

$$\|\mathbf{D}_{n+1}\bar{\mathbf{S}}_{n+1} - \mathbf{D}_n\bar{\mathbf{S}}_n\|^2 \to 0. \tag{35}$$

Proof:

Note that $\|\bar{\mathbf{S}}_{n+1/2} - \bar{\mathbf{S}}_{n+}\|^2 \to 0$ implies that $\|\bar{\mathbf{S}}^T_{n+1/2}(\bar{\mathbf{S}}_{n+1/2}\bar{\mathbf{S}}^T_{n+1/2})^{-1}\bar{\mathbf{S}}_{n+1/2} - \bar{\mathbf{S}}^T_n(\bar{\mathbf{S}}_n\bar{\mathbf{S}}^T_n)^{-1}\bar{\mathbf{S}}_n\|^2 \to 0$, [23], which in turn implies that $\|\mathbf{D}_{n+1}\bar{\mathbf{S}}_{n+1} - \mathbf{D}_n\bar{\mathbf{S}}_n\|^2 \to 0$ (Remember, $\mathbf{D}_{n+1}\bar{\mathbf{S}}_{n+1} = \mathbf{D}_{n+1/2}\bar{\mathbf{S}}_{n+1/2} = \mathbf{X}\bar{\mathbf{S}}^T_{n+1/2}(\bar{\mathbf{S}}_{n+1/2}\bar{\mathbf{S}}^T_{n+1/2})^{-1}\bar{\mathbf{S}}_{n+1/2}$).

■

F. Convergence

We have the following theorem.

Theorem 8. *The algorithm produces a sequence of $\mathbf{D}_n\bar{\mathbf{S}}_n$, such that either $\|\mathbf{X} - \mathbf{D}_n\bar{\mathbf{S}}_n\|_F \to 0$ or such that $\bar{\mathbf{S}}_n \to \bar{\mathbf{S}}^\star$, where the nonzero elements in row i converge to an element in the space spanned by the right singular vectors associated with the largest singular values of the feature vector matrix $\mathbf{X}_i$ containing those columns in $\mathbf{X}$ for which the i^{th} row in $\bar{\mathbf{S}}^\star$ has non-zero entries.*

In other words, the algorithm either finds K vectors $\mathbf{d}_i$ such that each feature $\mathbf{x}_i$ is a multiple of one $\mathbf{d}_i$ or it partitions the features into distinct clusters such that the feature vectors of each cluster are modelled with left and right eigenvectors associated to the largest eigenvalue of the feature sub matrix.

To proof this theorem, we distinguish three cases and proof convergence for each of these cases independently..

- After some n, cluster assignment does not change.
- Cluster assignment changes infinitely often due to changes in sparsity pattern in $\bar{\mathbf{S}}_{n_i}$, that is, there is an infinite sequence of $\bar{\mathbf{S}}_{n_i}$ such that $\bar{\mathbf{S}}_{n_i}$ and $\bar{\mathbf{S}}_{n_i+1/2}$ have different cluster assignments.
- Cluster assignment changes infinitely often due to empty clusters appearing after thresholding.

G. Case 1: Convergence for fixed cluster assignment

Assume that cluster assignment does no longer change after some iteration. In this case, we can treat the algorithm for each cluster independently. To do this, let us write the algorithm in terms of singular values of $\mathbf{X}_i$. In this case, we can re-write the update as

$$(\mathbf{s}_i^{n+1})^T = (\mathbf{s}_i^n)^T + \mu\frac{(\mathbf{s}_i^n)^T\mathbf{X}_i^T}{\|(\mathbf{s}_i^n)^T\mathbf{X}_i^T\|}\mathbf{X}_i\left(\mathbf{I} - \frac{\mathbf{s}_i^n(\mathbf{s}_i^n)^T}{(\mathbf{s}_i^n)^T\mathbf{s}_i^n}\right) \tag{36}$$

as (dropping the subscript i from α and Σ)

$$\alpha^{n+1} = \alpha^n + \mu\frac{\alpha^n}{\|\alpha^n\Sigma\|}\Sigma^2\left(\mathbf{I} - \frac{(\alpha^n)^T\alpha^n}{\|\alpha\|^2}\right), \tag{37}$$

where we have right multiplied the equation by $\mathbf{V}$ and used the fact that due to orthonormality of $\mathbf{U}$, $\|\alpha^n\Sigma\| = \|\alpha^n\Sigma\mathbf{U}\|$. Writing this update element wise, we see that the k^{th} element in α^n (i.e. α_k) is updated as

$$\alpha_k^{n+1} = \alpha_k^n + \mu\left(\frac{\sigma_k^2}{\|\alpha^n\Sigma\|} - \frac{\alpha^n\Sigma^2(\alpha^n)^T}{\|\alpha^n\Sigma\|\|\alpha^n\|^2}\right)\alpha_k^n. \tag{38}$$

i.e.

$$\alpha_k^{n+1} = \left(1 + \mu\left(\frac{\sigma_k^2}{\|\alpha^n\Sigma\|} - \frac{\|\alpha^n\Sigma\|}{\|\alpha^n\|^2}\right)\right)\alpha_k^n. \tag{39}$$

Let σ_j be the diagonal elements of Σ_i, which we will assumed are ordered $\sigma_j \geq \sigma_k$ whenever $j < k$. For each cluster, the algorithm produces a cluster centre $\mathbf{d}_i$ and cluster weight vectors $\mathbf{s}_i^T$ such that $\mathbf{X}_i \approx \mathbf{d}_i\mathbf{s}_i^T$, where $\mathbf{d}_i \propto \mathbf{X}_i\mathbf{s}_i/(\mathbf{s}_i^T\mathbf{s}_i)$. In the svd basis, this can be expressed as $\mathbf{s}_i^T = \alpha\mathbf{V}_i^T$ such that $\mathbf{d}_i = \mathbf{U}_i\Sigma_i\alpha^T/(\alpha\alpha^T)$ and $\mathbf{X}_i \approx \mathbf{U}_i\Sigma_i\alpha^T/(\alpha\alpha^T)\alpha\mathbf{V}_i^T$. That is, α is the representation of the cluster weights $\mathbf{s}_i^T$ in the right singular vector basis $\mathbf{V}_i$. Also, let α_j be the j^{th} element of α. We then have the following important result

Theorem 9. *Assume there is an iteration N, such that the cluster assignment stays fixed for all iterations $n > N$. Assume that at iteration $N+1$ the vectors α^{N+1} are such that the element $\alpha_{i_{max}}^{N+1} \neq 0$. Let $\mathcal{I}$ be the index set of the largest singular values, that is, $\sigma_{I_{max}} > \sigma_j$ whenever $i_{max} \in \mathcal{I}$ and $j \notin \mathcal{I}$. The algorithm then converges to a representation with $\sum_{i\in\mathcal{I}}\alpha_i \neq 0$ and $\alpha_j = 0$ for all $j \notin \mathbf{I}$. In other words, if the algorithm reaches an iteration after which cluster assignment no longer changes, and if at that iteration, the cluster weight vector $\mathbf{s}_i^T$ is not orthogonal to the subspace spanned by the right singular vectors of the feature matrix $\mathbf{X}_i$ associated with the largest singular values, then the weight vector $\mathbf{s}_i^T$ will converge to a vector that lies in this subspace. In particular, if the largest singular value is unique, then the algorithm converges to a vector collinear to the associated singular vector with $\alpha_{i_{max}} = \sigma_{i_{max}}$.*

Proof: Let us first recall that, due to normalisation of $\mathbf{d}_i$ and $\mathbf{s}_i^T$, we have $\frac{\|\alpha\Sigma\|}{\|\alpha\|^2} = 1$. Thus, the update of the k^{th} element in vector α

$$\alpha_k^{n+1} = \left(1 - \mu\frac{\|\alpha^n\Sigma\|}{\|\alpha^n\|^2} + \mu\frac{\sigma_k^2}{\|\alpha^n\Sigma\|}\right)\alpha_k^n \tag{40}$$

simplifies to

$$\alpha_k^{n+1} = \left(1 - \mu + \mu\frac{\sigma_k^2}{\|\alpha^n\Sigma\|}\right)\alpha_k^n \tag{41}$$

Without loss of generality assume that $\alpha_k^n > 0$ (Note that the update does not change the sign of α_k so we can repeat the same argument for negative α_k. Note however (see also below) that $\alpha_k^n = 0$ is not allowed as α_k will then remain constant.). Let us use the shorthand $c_k = \left(1 - \mu + \mu\frac{\sigma_k^2}{\|\alpha^n\Sigma\|}\right)$. Note that $0 < \mu \leq 1$ implies that c_k is positive. Looking at the normalised update we then have

$$\frac{(\alpha_k^{n+1})^2}{\|\alpha_k^{n+1}\|^2} = \frac{c_k^2}{\sum_i c_i^2(\alpha_i^n)^2}(\alpha_k^n)^2, \tag{42}$$

which can be rewritten as

$$\frac{(\alpha_k^{n+1})^2}{\|\alpha_k^{n+1}\|^2} = \frac{c_k^2}{\sum_i \lambda^n c_i^2}\frac{(\alpha_k^n)^2}{\|\alpha^n\|^2}, \tag{43}$$

where $\lambda_i = (\alpha_i^n)^2/\|\alpha^n\|^2$, so that $\sum_i \lambda_i c_i^2$ is a convex combination of the positive values c_i^2 (i.e.$\sum_i \lambda_i = 1$, $\lambda_i \geq 0$). We have thus shown that, for all i for which $c_i^2 > \sum_i \lambda_i c_i^2$, the normalised α_i increase (i.e. $\frac{c_k^2}{\sum_i \lambda^n c_i^2} > 1$), whilst for those $c_i^2 < \sum_i \lambda_i c_i^2$, we have a relative decrease. Furthermore, if all $a_i \neq 0$, then the largest relative increase is for the α_i associated with the largest singular values (as for those elements c_i^2 is maximal and as the maximum value of a set of positive numbers must be larger than any convex combination of the elements).

If we write $\tilde{c}_i^n = \frac{c_k^2}{\sum_i \lambda^n c_i^2}$, then we have the recursion

$$\frac{(\alpha_k^{n+1})^2}{\|\alpha_k^{n+1}\|^2} = \tilde{c}_i^n\frac{(\alpha_k^n)^2}{\|\alpha^n\|^2} = \prod_{N=0}^{n}\tilde{c}_k^N\frac{(\alpha_k^0)^2}{\|\alpha^0\|^2}, \tag{44}$$

Assume the singular values σ_i are ordered such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M$. This implies the same ordering on the c_i^n, i.e. $c_1^n \geq c_2^n \geq \cdots \geq c_M^n$ for all n.

Note that $\frac{(\alpha_k^{n+1})^2}{\|\alpha^{n+1}\|^2} \leq 1$ and thus, the sequences $\frac{(\alpha_k^{n+1})^2}{\|\alpha^{n+1}\|^2}$ are bounded. Furthermore, for those k associated with the largest singular values, the sequence is increasing, as for those k $\tilde{c}_k^n \geq 1$. This implies that for those k, the sequence $\frac{(\alpha_k^{n+1})^2}{\|\alpha^{n+1}\|^2}$ converges.

If $\alpha_k^0 \neq 0$, then for $\mu < 1$ $\tilde{c}_k^n \neq 0$. Alternatively, for $\mu = 1$, if there are singular values that are zero (i.e. $\sigma_m = 0$), then $\alpha_m^n = 0$ for all $n > 1$. In this case, we apply the following argument only to those α_i for which $\sigma_i \neq 0$. Thus, without loss of generality assume that $\alpha_k^n \neq 0$ and that $\tilde{c}_k^n \neq 0$. In this case, for all i for which σ_i is maximal, $1 \geq \frac{(\alpha_i^n)^2}{\|\alpha^n\|^2} \geq \frac{(\alpha_i^0)^2}{\|\alpha^0\|^2}$ for all n. Convergence of $\frac{(\alpha_i^n)^2}{\|\alpha^n\|^2}$ then also implies that the sequences $\tilde{c}_i^n$ converge to 1. (Because $\lim_{n\to\infty}\frac{(\alpha_i^n)^2}{\|\alpha^n\|^2} = \lim_{n\to\infty}\tilde{c}_i^n\frac{(\alpha_i^n)^2}{\|\alpha^n\|^2} = (\lim_{n\to\infty}\tilde{c}_i^n)(\lim_{n\to\infty}\frac{(\alpha_i^n)^2}{\|\alpha^n\|^2})$.) Thus in the limit, $\tilde{c}_1^n = \frac{(c_1^n)^2}{\sum_i \lambda_i^n(c_i^n)^2} \to 1$. However, as $\sum_i \lambda_i^n = 1$, we also have the requirement that

$$\tilde{c}_1^n = \frac{(c_1^n)^2}{\sum_i \lambda_i^n(c_i^n)^2} = \frac{1}{\sum_i \lambda_i^n(c_i^n/c_1^n)^2} > \frac{1}{\sum_i \lambda_i^n} = 1 \tag{45}$$

unless $\lambda_i^n = 0$ for all $c_i^n < c_1^n$. Thus convergence of $\tilde{c}_i^n$ to zero implies convergence of λ_i^n to zero for all i other than those i associated with the largest singular values. But this implies that $\lambda_i^n = (\alpha_i^n)^2/\|\alpha^n\|^2 \to 0$ for those i which in turn implies that $\sum_k(\alpha_k^n)^2/\|\alpha^n\|^2 \to 1$, where we sum over those k associated with the largest singular values. ■

H. Case 2: infinite changes in sparsity pattern

By Lemma 6 there is an N such that $\|\bar{\mathbf{S}}_{n_i} - \bar{\mathbf{S}}_{n_i+1/2}\| \leq \epsilon$ for all $\epsilon > 0$. Let $n_i > N$ be an infinite sequence of indices such that $\bar{\mathbf{S}}_{n_i}$ has a different support to $\bar{\mathbf{S}}_{n_i+1/2}$. Let $\mathcal{I}$ be the set of indices of columns in $\mathbf{S}$ that have elements that change infinitely often from zero to a non-zero value and vice versa. As the difference $\|\bar{\mathbf{S}}_{n_i} - \bar{\mathbf{S}}_{n_i+1/2}\|_F^2 \to 0$, this implies that $\|\mathbf{S}_{\mathcal{I}}\|_F^2 \to 0$, where $\mathbf{S}_{\mathcal{I}}$ is the sub matrix made of columns of $\mathbf{S}$ indexed by $\mathcal{I}$. Thus, $\mathbf{S}_{\mathcal{I}} \to 0$, whilst the columns in $\bar{\mathbf{S}}$ not indexed by $\mathcal{I}$, say $\bar{\mathbf{S}}_{\mathcal{I}^c}$ will converge to right singular vectors of the feature vector matrix using arguments that mirror those described above.

I. Case 3: infinitely many empty clusters

Assume there is an infinite sequence of $\bar{\mathbf{S}}_{n_i}$ for which $\bar{\mathbf{S}}_{n_i}$ has empty clusters. We know that after empty cluster re-initialisation and normalisation,

$$\begin{aligned}&\|\mathbf{X}-\mathbf{D}_{n_i+1}\bar{\mathbf{S}}_{n_i+1}\|_F\\ &\leq \|\mathbf{X}(\mathbf{I}-\bar{\mathbf{S}}_{n_i+1/2}^T(\bar{\mathbf{S}}_{n_i+1/2}\bar{\mathbf{S}}_{n_i+1/2}^T)^{-1}\bar{\mathbf{S}}_{n_i+1/2})\|_F\\ &\leq \|\mathbf{X}\|_F \end{aligned} \quad (46)$$

(where we set the inverse of the zero element in $\bar{\mathbf{S}}_{n_i+1/2}\bar{\mathbf{S}}_{n_i+1/2}^T$) to zero) so that there is an infinite sequence $\|\mathbf{X}-\mathbf{D}_{n_i+1}\bar{\mathbf{S}}_{n_i+1}\|_F$ that is bounded. The Bolzano-Weierstrass theorem then implies the existence of an infinite convergent subsequence. $\|\mathbf{X}^\star-\mathbf{D}_{\tilde{n}_i}\bar{\mathbf{S}}_{\tilde{n}_i}\|_F \to 0$, where the $\bar{\mathbf{S}}_{\tilde{n}_i-1}$ have empty clusters.

As our arguments are independent of exactly which of the columns in $\mathbf{X}$ we use to re-initialise the empty cluster (as long as we don't choose one from a cluster with a single element), we assume that w.l.g. we take that element for which $\|\mathbf{x}_i-\mathbf{d}_i s_j\|$ is maximal. But the fact that $\|\mathbf{X}-\mathbf{D}_n\bar{\mathbf{S}}_n\|_F$ converges for all n then implies that $\|\mathbf{x}_i-\mathbf{d}_i s_j\| \to 0$, that is, $\|\mathbf{X}-\mathbf{D}_n\bar{\mathbf{S}}_n\|_F \to 0$.

IV. Numerical Results

The performance of our new approach was evaluated using artificial data as well as real data-sets. We evaluated our results using a selection of popular metrics (maximum cluster overlap (as in [9]), Dice similarity (DICE) [24], Normalised Mutual Information (NMI) [25] and Adjusted Rand Index (RI) [26]). All of these measure show qualitatively similar results. We thus report most of our results in terms of normalised mutual information. Only the results on brain parcellation will be reported in terms of Dice similarity as this is the more common measure in the brain imaging literature.

Dice's similarity is defined as $DICE(A,B)=\frac{2|A\bigcap B|}{|A|+|B|}$. The notation $|A|$ refers to the number of elements in set A. Dice's similarity only measure similarity between two clusters and to compute a measure that can compare entire clusterings, we 1) calculate the similarity between any pair of clusters taken from the two clusterings 2) permute this similarity matrix greedily, so that each entry along the diagonal is no smaller than any other entry in the sub-matrix formed from the elements that are below and to the right of that diagonal element, 3) average over the matrix diagonal.

Normalised Mutual Information (NMI) compares clusterings directly. Let $\mathcal{C}_1$ be a partitioning of a set of N features into k_1 distinct clusters and $\mathcal{C}_2$ a partitioning of the same features into k_2 clusters. Let n_i^1 be the number of features in cluster i in clustering 1 and n_j^2 the number of features in cluster j in clustering 2. Similarly, let $n_{i,j}$ be the number of features that are both, in cluster i in partition 1 and in cluster j in partition 2. The NMI is then

$$NMI(\mathcal{C}_1,\mathcal{C}_2)=\frac{\sum_{i=1}^{k_1}\sum_{j=1}^{k_2} n_{i,j}\log\left(\frac{Nn_{i,j}}{n_i^1 n_j^2}\right)}{\sqrt{\left(\sum_{i=1}^{k_1} n_i^1\log\frac{n_i^1}{N}\right)\left(\sum_{i=1}^{k_1} n_j^2\log\frac{n_j^2}{N}\right)}} \quad (47)$$

A. Comparison of Different Versions of Our Approach using a Synthetic Data Set

The synthetic data sets in the first set of test problems were generated by randomly generating matrices $\mathbf{D}^\star$ and binary $\mathbf{S}^\star$ from which the observations were constructed as $\mathbf{X}=\mathbf{D}^\star\mathbf{S}^\star+\mathbf{E}$, where $\mathbf{E}$ is an i.i.d. Gaussian noise term. $\mathbf{D}$ was a 1000 by 10 matrix (i.e. we generated 10 cluster centres) and $\mathbf{S}$ was of dimension 10 by 100 (that is, we generated 100 observations).

We varied the standard deviation of $\mathbf{E}$ from $0.01, 0.1, 1$ and 10 and contrasted two different regimes, one, in which the average number of features in each cluster were identical and one in which one cluster had 91 features and all other clusters had a single feature.

We compared our method with several variations and averaged the results over 1000 random problem instances. The results are shown in Tables I and II, where, for each noise level, we have highlighted the best performing algorithm version in bold. Results are here reported in terms of NMI, as the other measures gave qualitatively similar results (see also the comparison of performance metric in the next subsection).

TABLE I
Performance with equally sized clusters in terms of NMI.

Update of $\mathbf{S}$: normalisation:	$\mathbf{S}+\mu\mathbf{D}^T(\mathbf{X}-\mathbf{DS})$	$(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{X}$
std	0.01, 0.1, 1, 2	0.01, 0.1, 1, 2
$\mathbf{D}$	0.951, 0.939, 0.841, 0.564	0.984, 0.965, 0.861, 0.577
$\mathbf{S}$	0.948, 0.931, 0.841, **0.587**	**0.989, 0.976, 0.872,** 0.584
NONE	0.951, 0.936, 0.842, 0.571	0.982, 0.967, 0.866, 0.584

TABLE II
Performance with varying cluster sizes in terms of NMI.

Update of $\mathbf{S}$: normalisation:	$\mathbf{S}+\mu\mathbf{D}^T(\mathbf{X}-\mathbf{DS})$	$(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{X}$
std	0.01, 0.1, 1, 2	0.01, 0.1, 1, 2
$\mathbf{D}$	0.445, 0.388, 0.329, 0.270	**0.876, 0.730, 0.408,** 0.260
$\mathbf{S}$	0.491, 0.367, 0.325, 0.266	0.834, 0.665, 0.308, 0.252
NONE	0.616, 0.378, 0.334, **0.273**	0.57, 0.679, 0.343, 0.256

From these results we see that, apart from the condition with very high noise, an update of $\mathbf{S}$ based on the pseudo-inverse of $\mathbf{D}$ is advantageous. If cluster size is roughly equal between clusters, then a pre-thresholding normalisation of the rows of $\mathbf{S}$ seems to perform better, whilst for clusters of varying size, normalisation of columns of $\mathbf{D}$ works best. Interestingly, if we use the gradient type update $\mathbf{S}+\mu\mathbf{D}^T(\mathbf{X}-\mathbf{DS})$, then an algorithm without column normalisation seems to be the best choice in both conditions.

B. Comparison of Different Algorithms on Synthetic Data Sets

Synthetic data sets were again generated by randomly generating matrices $\mathbf{D}^\star$ and binary $\mathbf{S}^\star$ and i.i.d. Gaussian noise $\mathbf{E}$. Three different datasets were generated simulating different clustering scenarios:

1) **Dataset 1:** $\mathbf{D}\in\mathbb{R}^{M\times K}$ was generated with i.i.d Gaussian zero-mean unit-variance entries. $\mathbf{S}\in\mathbb{R}^{K\times N}$ was generated with each column set to zero apart from one entry whose location was chosen at random and

whose value was set to 1. For this data-set, all clusters had thus roughly the same number of observations per cluster.

2) **Dataset 2:** As dataset 1, but with $\mathbf{S}$ generated so that each cluster had different numbers of observations $\mathbf{x}_i$. We here used an extreme example, where there were 3 clusters with only 1 observation, 2 clusters with 3 observations, and 1 cluster each with 6, 10, 14, 24 and 36 observations respectively.
3) **Dataset 3:** As dataset 2, but with cluster centres in $\mathbf{D}$ each scaled by a random scaling drawn uniformly from 0 to 1. Thus each cluster did have a different level of noise compared to the size of the cluster centre (or, after normalisation of each $\mathbf{x}_i$ each cluster had a different amount of within cluster variance).

To each of these datasets, four different levels of noise were added with the entries in $\mathbf{E}$ having a variance of 0 (no noise, i.e. the $\mathbf{x}_i$ are cluster centres), a variance of 1, a variance of 4 and a variance of 9 (See figures (2) to (4) for average SNR values for each condition). Noise was added before normalisation of the observations and results are averaged over 500 different realisations of each condition.

The results for the three datasets are shown in Figures 2, 3 and 4, where we compare our method to a range of alternative approaches. The k-means and spherical k-means are standard implementations of the algorithms, where empty clusters are re-initialised with randomly selected feature vectors. Semi-NMF clustering uses the semi-Nonnegative Matrix Factorisation method of [9], followed by a hard cluster assignment based on the magnitude of the entries in the Nonnegative matrix in the matrix decomposition. The next two approaches, CLUTO [27] and gmeans [28], are advanced, bespoke spherical clustering algorithms developed for text clustering. We used CLUTO's vcluster command with default settings and gmeans with the spherical clustering flag. All algorithms were initialised with the same initial cluster assignment. When analysing the results, the following should be kept in mind. CLUTO uses a "randomized incremental optimization algorithm" [27] and by default compares 10 solutions to report the result with the minimal cost, whilst gmeans uses an additional refinement step that uses a local search strategy to further refine an initial spherical k-means clustering result. Both of these refinement approaches, Monte Carlo search and local refinement, are potential strategies that might be combined with our approach also. This will increase computational complexity but is likely to increase performance in the same way in which these strategies are able to increase performance of, for example, spherical k-means.

It is clear that for the experiments reported here, our approach outperforms all other reproaches over all datasets and noise conditions in terms of computation time and outperforms most other methods in terms of clustering performance. Only CLUTO, gmeans and K-EVD perform better in some of the experiments, though they are also much slower. For CLUTO and gmeans some of the difference in speed are due to file i/o overhead required by the standalone algorithms, but this does not explain all of the difference in speed as CLUTO uses ten independent cluster runs and gmeans adds an additional local search procedure. K-EVD only outperforms our approach for two noise conditions in experiment one, but this comes at the cost of substantially increase computation cost (two orders of magnitude), due to the repeated requirement to estimate the singular vector associated with the largest singular value. Other observations worth pointing out include

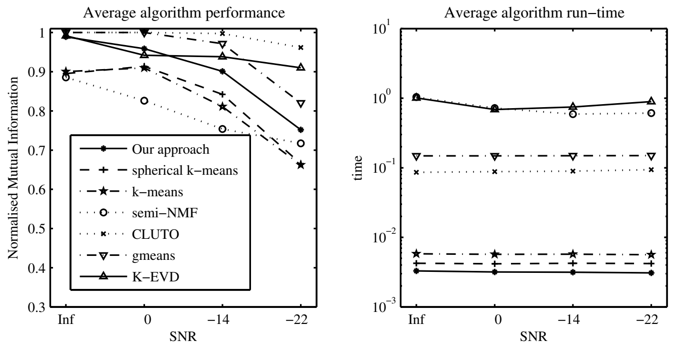

Fig. 2. Performance for artificial dataset 1 and for 4 noise levels. Performance is shown in terms of NMF (left) and computation time (right).

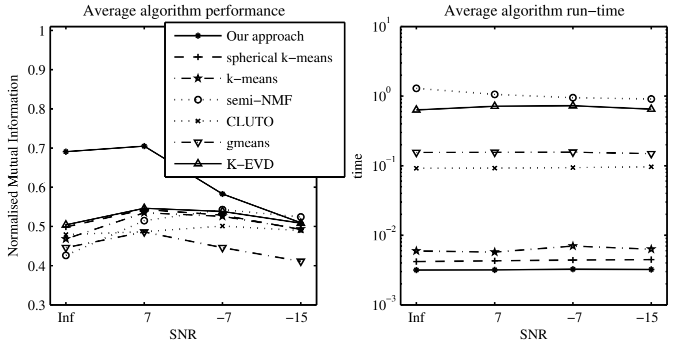

Fig. 3. Performance for artificial dataset 2 and for 4 noise levels. Performance is shown in terms of NMF (left) and computation time (right).

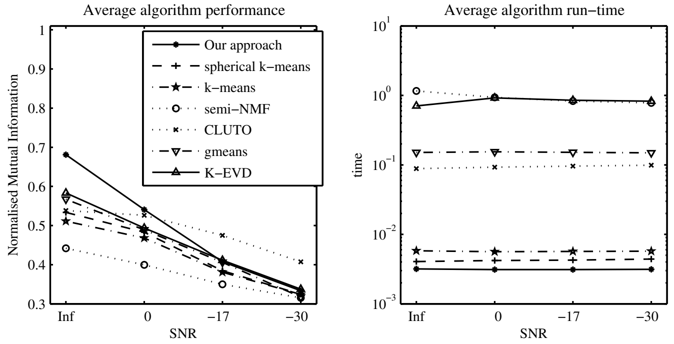

Fig. 4. Performance of our algorithm, spherical k-means, k-means, semi-NMF, CLUTO, gleans and K-EVD clustering for artificial dataset 3 and for 4 noise levels. Performance is shown in terms of NMF (left) and computation time (right).

1) The semi-NMF algorithm sometimes performs better than k-means and sometimes it performs worse.
2) Spherical k-means performs better than non-spherical k-means run on normalised vectors but the difference is small.
3) Unsurprisingly, increasing the noise clearly reduces per-

formance.

4) There is a clear performance decrease when going from dataset 1 to dataset 2, though going from dataset 2 to dataset 3 only reduces performance slightly.

We also tried two EM algorithms, one based on a Gaussian mixture model and one based on a von Mises-Fisher Mixture model (using the code on http://suvrit.de/work/progs/movmf/), but these methods performed poorly[2] and we do not show the results here.

C. Synthetic functional Brain Data

We developed the approach for a specific problem in brain imaging and the next artificial data sets simulate this. We are interested in the clustering of a spatial data-set, where each spatial location has an associated time-series (the feature vector). The aim is then to cluster the time-series or features to recover the spatial clusters. To simulate such a data-set, we generated a spatial grid (64 × 64) and split this grid into 40 spatially connected regions. This was done by randomly selecting 40 cluster seed locations on the grid. The seeds are then grown by adding one randomly chosen spatial neighbourhood point to one randomly chosen cluster. This is repeated until the entire spatial grid is covered. An example can be seen in the top left of Figure 5. Whilst these clusters have clear spatial structure, this was not used in the clustering itself, where features were grouped based on the similarity of their time-series (or feature vector).

For each cluster, these feature vectors were drawn from different distributions. We thus generated three different datasets.

1) Features within each cluster were generated from an i.i.d. Gaussian, with a mean that was itself drawn from an i.i.d. zero-mean, unit variance Gaussian. The within cluster variance was varied between 1 and 3, producing SNR values of 0dB, -3 dB and -9dB respectively. This is intended as a very rough simulation of a *functional* Magnetic Resonance Imaging dataset (see below).
2) The data was generated as in 1) above, but additional spatial smoothing was applied to simulate spatial correlation between features as observed in real brain imaging data. Smoothing was achieved by averaging spatially close feature vectors using a Gaussian smoothing kernel. The amount of smoothing varied within each data-set and the Gaussian kernel had a standard deviation that varied from 0.2 to 5 pixels.
3) Cluster centres were generated from a Beta distribution with both parameters set to 2. For each of the clusters, observations were then drawn from a Beta distribution whose parameters were calculated such that the distribution had a variance of 0.2 and a mean equal to the cluster's mean. Each observation $\mathbf{x}_i$ thus had entries between 0 and 1. This data is a rough approximation simulating brain connectivity data as estimated using *diffusion* Magnetic Resonance Imaging techniques (see below).

We again evaluate the cluster assignment using Normalised Mutual Information (the other measures again show similar differences between approaches). The results for the three different datasets are shown in figure 6, with a visual representation of the spatial clusters and their estimates for the data set (1) with -3dB of SNR shown in figure 5. Figure 6 plots the Normalised Mutual Information agains computation time (on a logarithmic scale). We here compared our approach to semi-NMF, CLUTO, gmeans and K-EVD. In general, our approach outperforms the other approaches, especially for moderate to low noise. Only the -9dB SNR condition does not show our method as a clear winner, with the gmeans algorithms performing slightly better and working faster.

We also run our method on the same datasets using a recursive scheme in which we changed the number of clusters to optimise the Akaike information Criterion (AIC). This method was able to correctly estimate the number of clusters (±2) and AIC optimal clusters were found to have a NMI similarity to the original clusters comparable to those observed when specifying the correct number of clusters.

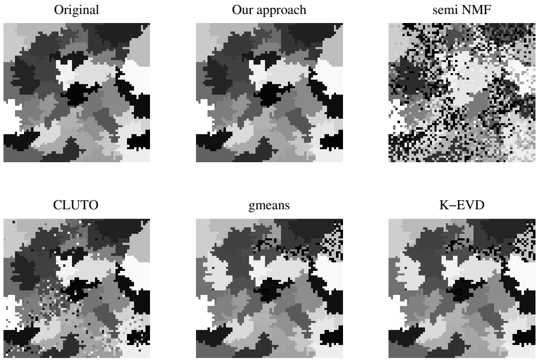

Fig. 5. Example of spatial distribution of feature vectors (top left) and estimates calculated with different methods.

D. Application to brain parcellation

Our third experiment evaluates our new clustering method on actual brain imaging data. Neuroscientists are interested in a detailed understanding of connections in the human brain and modern Magnetic Resonance Imaging (MRI) techniques offer two complementary methods to study these brain connections [29], [30]. Diffusion MRI [30] methods allow estimates of major fibre bundles to be computed and, by tracking individual fibres, the connection between distant brain parts can be studied (so called structural connectivity). An alternative view of brain connectivity is offered by functional MRI studies. For example, by measuring blood oxygenation changes in the brain during rest, statistical relationships between the activation of different brain regions can be estimated [29]. If brain activation in distinct regions shows statistical dependancy, then these regions must exchange information and must therefore be connected in some way (so called functional connectivity). MRI studies often measure brain properties on three dimensional spatial grids of 2 to 4 millimetres and, for the average

[2] This was mainly due to the methods difficulty in estimating within-cluster variance, a problem that could potentially be overcome with a full Bayesian model, though this would further increase the computational burden for this rather slow approach.

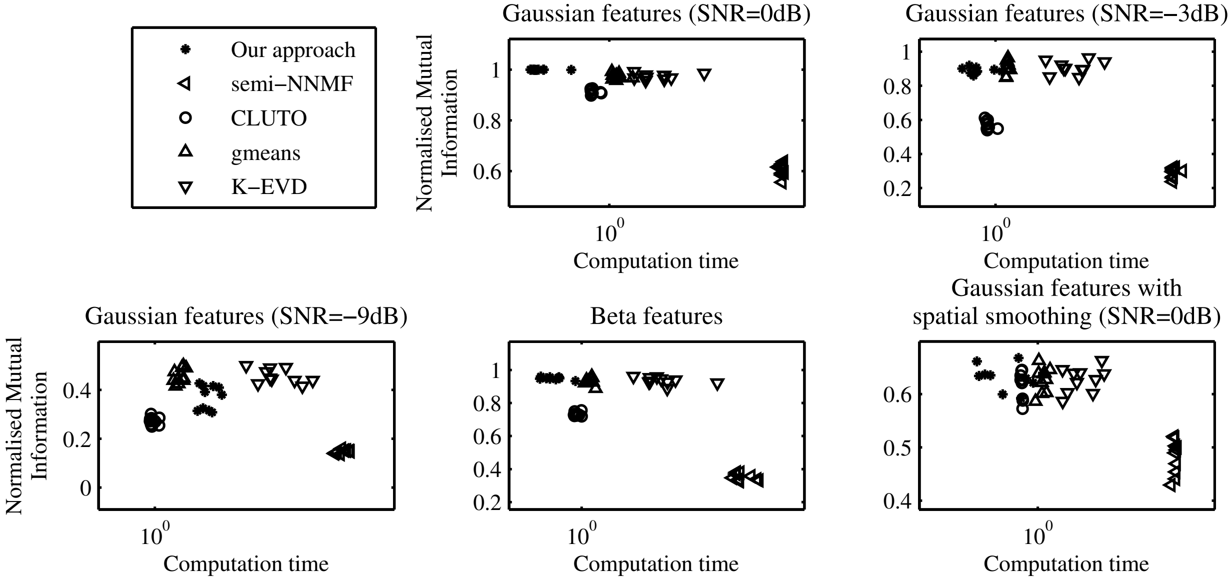

Fig. 6. Computation time vs. Normalised Mutual Information between original and estimated spatial cluster assignments (using Gaussian distributed feature vectors with 3 different levels of variance, using Gaussian distributed feature vectors and spatial Gaussian smoothing with spatially varying variance and additional noise leading to an SNR of 0 and using Beta distributed feature vectors).

human brain, this leads to very high dimensional problems, where the connection between hundreds of thousands of brain areas has to be estimated. This cannot be done reliably and a fundamental first step is the decomposition of the brain into a smaller set of brain areas. Whilst regions can be defined based on neural anatomy found in post-mortem studies, or through the agglomeration of large brain imaging studies that use specific cognitive tasks to study a specific brain region, there are many reasons (such as the large variability in functional brain anatomy between people) why these partitions are not ideal substrates on which to base connectivity analysis. There is thus now an extensive literature on the development of algorithms to partition the human brain based on functional MRI data acquired during rest [31], [32], [33], [34], [35], [36], [37], [38], [39]. We test our algorithm on the same problem.

We used fMRI and structural MRI data from 66 subjects, collected during the initial stages of phase 2 of the human connectome project (http://humanconnectome.org/). The data had 2mm isotropic spatial resolution and a temporal resolution of 1.4 seconds. The data was processed using a preliminary version of the Human Connectome Project's structural and functional minimal preprocessing pipelines, final versions to be published separately (Glasser et al. unpublished). Briefly, this involved brain extraction, registration of different MRI modalities, bias field correction, registration to a standard brain template and cortical surface modelling. Functional data were motion corrected, distortion corrected, mean normalized and resampled to the cortical surface. Standard surface smoothing and temporal filtering was applied and ICA based noise reduction used.

For each of the 66 subjects, the dataset consisted of a set of approximately 64000 functional MRI time series, each with approximately 1000 temporal samples each. We split the dataset into two, with 33 subjects each. For each of these splits, we combined the data across subjects by estimating the 1000 left singular vectors of the spatio-temporal data matrix (concatenated in the temporal direction over the 33 subjects). We thus produced two sets of feature vectors, where each vector had a length of 1000 and was associated with one of the vertex locations on the cortical grid representation.

As there is no ground truth available for this experiment, we estimate the performance based on the ability of an algorithm to reliably identify clusters in each of the two split datasets. The results are compared visually in figure 7, where we show an inflated representation of the left and right cortical surface and the estimated clusters from the two data-sets (left vs. right). Grey levels were matched to ease visual comparison.

A numerical evaluation in terms of the Dice[3] similarity between the clusters derived form each of the two datasets is shown in Figure 8. The results obtained for different number of clusters and different methods is shown. Before calculating dice similarity, we split all clusters we estimated into spatially contiguous regions and then discarded very small clusters (we here removed clusters that had less than 20 features, though the flavour of the results does not vary much if we use another threshold). Also shown are results directional k-means and a recently developed region growing based method that explicitly enforces clusters to be spatially connected [39].

We also tried the normalised cuts spectral clustering method of [40] on this problem. As it is not feasible to calculate and save the entire similarity matrix for all features, we here generated sparse versions by thresholding the correlation at 0.5 and 0.4. However, the results did not compare well to the other methods tested and are thus omitted.

We can see that our method performs much better than the region growing approach and better than the semi-NMF

[3] We used dice similarity here, as this is a common measure used in the field

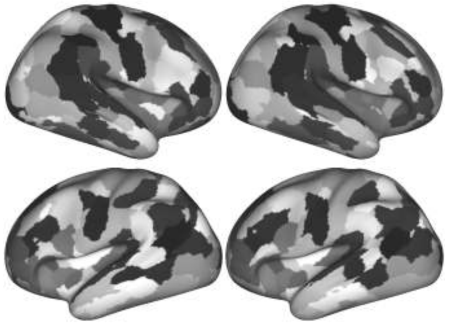

Fig. 7. Repeatability of clustering of the cortical surface based on resting-state fMRI data. Clusters derived from two different groups of 33 subjects each are shown on the left and right on an inflated rendering of the cortical surface. Right hemisphere (top) and left hemisphere (bottom).

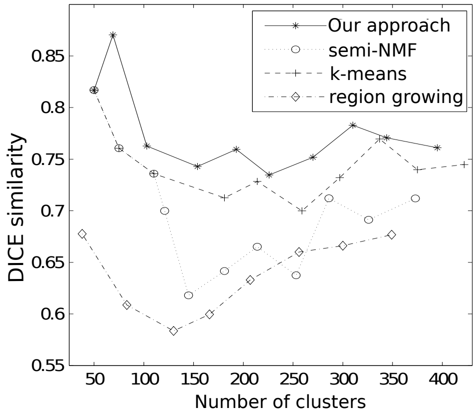

Fig. 8. Comparison of four different approaches for clustering of the cortical surface based on resting-sate fMRI data. Repeatability measured in terms of average Dice similarity between cluster regions plotted for different numbers of clusters. For each approach, the clusters were derived from two different groups of 33 subjects. Before the calculation of Dice similarity, clusters were split into spatially homogeneous regions and small clusters were removed.

algorithm. To interpret these results, it must be remembered that the region growing algorithm enforces clusters to be spatially connected. This is known to introduce additional biases into the estimated clusters, which in turn generally means that clusters are more repeatable. Our approach does not include such an additional spatial constraint and is thus not affected by the associated bias and is thus a more reliable indicator of intrinsic data-structure.

E. Performance on standard data sets

We conclude this section with an analysis of more general data-sets used elsewhere in the clustering literature. In particular, we used the following 3 datasets:

- Data set 1 **WAVE**: This data-set, generated for [41, p. 49-55, 169] can be retrieved from http://archive.ics.uci.edu/ml/datasets/Waveform+Database+Generator+(Version+1)). The data-set consisted of 5000 features each with 21 elements. Features were form 3 different classes and contained gaussian noise.
- Data set 2a,b **NEWS**: is a text analysis data-set consisting of bag of words feature vectors, generated originally for [42]. We used the version of the database in which there are 20 Newsgroups sorted by date (retrieved from http://qwone.com/ jason/20Newsgroups/). We used subsets of this data with 500 features of length 53975 and clustered these into the 20 classes. Different subsets were used with version (a) of the dataset generated by randomly taking 25 features from each newsgroup whilst dataset (b) was generated by randomly taking subsets of varying size from each news group (the number of features varied exponentially between 1 and 102).
- Data set 3 **MXM**: was a subset of the bag of word features generated for [43], (retrieved from http://labrosa.ee.columbia.edu/millionsong/musixmatch). This dataset contains bag of words representations for the lyrics from a music database. We extracted a subset of the BoW features, corresponding to music from 6 different musical genres (techno, rock, pop, punk, country and hip hop). There were 5000 BoW features in this data-set of length 12921. We used these features to see if we could use a blind clustering approach to distinguish the different musical genres based on the lyrics alone.

The result of the analysis of the three data-sets are shown in table III, measured using Normalised Mutual Information and contrasting our approach to semi-NMF and spherical k-means. Whilst overall performance on these data-sets is low (they are difficult data-sets to cluster), it is evident that our approach outperforms the other approaches.

TABLE III
COMPARISION OF DIFFERENT METHODS ON DIFFERENT DATA SETS MEASURED IN NMI

	WAVE	NEWS(a)	NEWS(b)	MXM
Our approach	**0.3676**	**0.1864**	**0.1447**	**0.0545**
semi-NMF	0.3466	0.1148	0.0947	0.0489
spherical k-means	0.2801	0.1636	0.1253	0.0507

V. CONCLUSIONS

We have here proposed a simple and fast algorithm that can efficiently cluster feature vectors based on their direction. The approach is based on matrix factorisation ideas and these allowed us to design an algorithm that is applicable to relatively large, non-sparse clustering problems where hundreds of thousands of feature vectors are clustered into hundreds of clusters, even in settings where features are not sparse. Our method was shown to outperform a wide range of competing techniques. Only three other algorithms were found to sometimes perform better, but these were also much slower. Two of these approaches used either Monte Carlo search strategies or additional local search post-processing, two approaches which are likely to also improve the performance of our method, though at similar computational costs. There remain several aspect of the method that require further investigation. Of theoretical interest are conditions on the original cluster features that would guarantee the algorithm to cluster the features correctly. Of interest to our brain imaging work is another extension that uses of additional constraints in cluster assignment. Here additional spatial neighbourhood constraints might be of particular relevance. Whilst we tried to evaluate our approach on a large set of clustering problems from different application domains, there remain many application areas that we did not evaluate. Whether or not our approach is beneficial in these settings remains to be seen.

REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[2] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA, Jun. 1967, pp. 281–297.

[3] R. Arora, M. R. Gupta, A. Kapila, and M. Fazel, "Similarity-based clustering by left-stochastic matrix factorization," *Journal of Machine Learning Research*, vol. 14, pp. 1715–1746, 2013.

[4] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, no. 1, pp. 143–175, 2001.

[5] S. Zhong, "Efficient online spherical k-means clustering," in *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN 2005)*, Montreal, QC, Canada, Aug. 2005, pp. 3180–3185.

[6] K. Hornik, I. Feinerer, M. Kober, and C. Buchta, "Spherical k-means clustering," *Journal of Statistical Software*, vol. 50, no. 10, pp. 1–22, 2012.

[7] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von Mises-Fisher distributions," *The Journal of Machine Learning Research*, vol. 6, pp. 1345–1382, Jun. 2005.

[8] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 2765–2781, Nov. 2013.

[9] C. Ding, L. T., and M. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 45–55, Jan. 2010.

[10] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons, "Document clustering using nonnegative matrix factorization," *Information Processing and Management: an International Journal archive*, vol. 42, no. 2, pp. 373–386, 2006.

[11] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proc. SIAM Data Mining Conf.*, Newport Beach, CA, Apr. 2005, pp. 606–610.

[12] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[13] C. Li and C. Ding, "The relationships among various nonnegative matrix factorization methods for clustering," in *Proceedings of the sixth International Conference on Data Mining*, Hong Kong, China, Dec. 2006, pp. 362–371.

[14] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: John Wiley & Sons, 2001.

[15] C. Beckmann and S. Smith, "Probabilistic independent component analysis for functional magnetic resonance imaging," *IEEE Trans. on Medical Imaging*, vol. 23, pp. 137–152, Feb. 2004.

[16] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, pp. 4311–4322, Nov. 2006.

[17] Z. He and A. Cichocki, "K-EVD clustering and its applications to sparse component analysis," in *Proceedings of the 6th international conference on Independent component analysis and signal separation*, Charleston, CS, Mar. 2006, pp. 90–97.

[18] P. Georgiev, F. Theis, and A. Ralescu, "Identifiability conditions and subspace clustering in sparse BSS," in *Proceedings of the 7th international conference on Independent component analysis and signal separation*, London, UK, Sep. 2007, pp. 357–364.

[19] C. Bouveyron and C. Brunet-Saumard, "Model-based clustering of high-dimensional data: A review," *Computational Statistics & Data Analysis*, vol. 71, pp. 52–78, Mar. 2014.

[20] T. Blumensath and M. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[21] ——, "Iterative hard thresholding for compressed sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 298 – 309, Feb. 2010.

[22] ——, "Iterative thresholding for sparse approximations," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 629–657, 2008.

[23] G. W. Stewart, "On the perturbation of pseudo-inverses, projections and linear least squares problems," *SIAM Review*, vol. 19, pp. 634–662, 1977.

[24] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[25] A. Strehl and J. Ghosh, "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, Dec. 2002.

[26] L. Hubert and P. Arabic, "Comparing partitions," *Journal of Classification*, vol. 2, pp. 193–218, 1985.

[27] G. Karypis, "Cluto: A clustering toolkit," Department of Computer Science, University of Minnesota, Tech. Rep. 02-017, 2003. [Online]. Available: http://glaros.dtc.umn.edu/gkhome/fetch/sw/cluto/manual.pdf

[28] I. S. Dhillon, Y. Guan, and J. Kogan, "Iterative clustering of high dimensional text data augmented by local search," in *Proceedings of the 2002 IEEE International Conference on Data Mining*, Maebashi City, Japan, Dec. 2002.

[29] B. Biswal, F. Z. Yetkin, V. M. Haughton, and J. S. Hyde, "Functional connectivity in the motor cortex of resting human brain using echo-planar mri," *Magn Reson Med*, vol. 34, no. 4, pp. 537–541, 1995.

[30] J. C. Klein, T. E. Behrens, M. D. Robson, C. Mackay, D. J. Higham, and H. Johansen-Berg, "Connectivity-based parcellation of human cortex using diffusion mri: Establishing reproducibility, validity and observer independence in ba 44/45 and sma/pre-sma," *Neuroimage*, vol. 34, no. 1, pp. 204–211, 2007.

[31] J. A. Mumford, S. Horvath, M. C. Oldham, P. Langfelder, D. H. Geschwind, and R. A. Poldrack, "Detecting network modules in fmri time series: a weighted network analysis approach," *Neuroimage*, vol. 52, no. 4, pp. 1465–1476, 2010.

[32] R. C. Craddock, G. A. James, P. E. Holtzheimer, X. P. Hu, and H. S. Mayberg, "A whole brain fmri atlas generated via spatially constrained spectral clustering," *Hum. Brain Mapp.*, vol. 33, no. 8, pp. 1914–1928, 2012.

[33] J. D. Power, A. L. Cohen, S. M. Nelson, G. S. Wig, K. A. Barnes, J. A. Church, A. C. Vogel, T. O. Laumann, F. M. Miezin, B. L. Schlaggar, and S. E. Petersen, "Functional network organization of the human brain," *Neuron*, vol. 72, pp. 665–678, 2011.

[34] J. Zhang, X. Tuo, Z. Yuan, W. Liao, and H. Chen, "Analysis of fmri data using an integrated principal component analysis and supervised affinity propagation clustering approach," *IEEE Trans. on biomedical engineering*, vol. 58, pp. 3184–3196, Nov. 2011.

[35] D. Lashkari, E. Vul, N. Kanwisher, and P. Golland, "Discovering structure in the space of fmri selectivity profiles," *NeuroImage*, vol. 50, pp. 1085–1098, 2010.

[36] P. Bellec, P. Rosa-Neto, O. C. Lyttelton, H. Benali, and A. C. Evans, "Multi-level bootstrap analysis of stable clusters in resting-state fmri," *NeuroImage*, vol. 51, pp. 1126–2239, 2010.

[37] X. Shen, X. Papademetris, and R. T. Constable, "Graph-theory based parcellation of functional subunits in the brain from resting-state fmri data," *NeuroImage*, vol. 50, pp. 1027–1035, 2010.

[38] B. T. Yeo, F. M. Krienen, J. Sepulcre, M. R. Sabuncu, D. Lashkari, M. Hollinshead, J. L. Roffman, J. W. Smoller, L. Zllei, J. R. Polimeni, B. Fischl, H. Liu, and R. L. Buckner, "The organization of the human cerebral cortex estimated by intrinsic functional connectivity," *J. Neurophysiol.*, vol. 106, no. 3, pp. 1125–1165, 2011.

[39] T. Blumensath, S. Jbabdi, M. F. Glasser, D. C. Van Essen, K. Ugurbild, T. E. J. Behrens, and S. M. Smith, "Spatially constrained hierarchical parcellation of the brain with resting-state fmri," *NeuroImage*, vol. 76, pp. 313–324, 2013.

[40] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Learning*, vol. 22, pp. 888–905, Aug. 2000.

[41] L. Breiman, J. H. Friedman, A. Olshen, and J. Stone, *Classification and Regression Trees*. Boca Raton: Chapman and Hall/CRC, 1984.

[42] K. Lang, "Newsweeder: Learning to filter netnews," in *Proceedings of the Twelfth International Conference on Machine Learning*, Kittilä, Finland, Aug. 1995, pp. 331–339.

[43] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, Miami, FL, Oct. 2011.