GRASSMANN CLUSTERING

Peter Gruber and Fabian J. Theis

Institute of Biophysics, University of Regensburg
93040 Regensburg, Germany
phone: +49 941 943 2924, fax: +49 941 943 2479
email: fabian@theis.name, web: http://fabian.theis.name

ABSTRACT

An important tool in high-dimensional, explorative data mining is given by clustering methods. They aim at identifying samples or regions of similar characteristics, and often code them by a single codebook vector or centroid. One of the most commonly used partitional clustering techniques is the k-means algorithm, which in its batch form partitions the data set into k disjoint clusters by simply iterating between cluster assignments and cluster updates. The latter step implies calculating a new centroid within each cluster. We generalize the concept of k-means by applying it not to the standard Euclidean space but to the manifold of subvectorspaces of a fixed dimension, also known as the Grassmann manifold. Important examples include projective space i.e. the manifold of lines and the space of all hyperplanes. Detecting clusters in multiple samples drawn from a Grassmannian is a problem arising in various applications. In this manuscript, we provide corresponding metrics for a Grassmann k-means algorithm, and solve the centroid calculation problem explicitly in closed form. An application to nonnegative matrix factorization illustrates the feasibility of the proposed algorithm.

1. INTRODUCTION

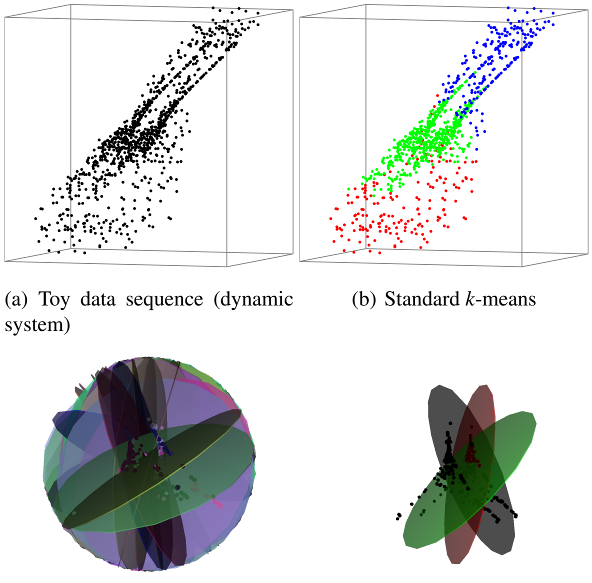
Clustering denotes the detection of common features within a data set. It has many applications in fields as varied as signal processing, telecommunications, biomedical data analysis and financial markets. Clustering is typically performed in the data space itself [8]. Some extensions, namely subspace clustering allow for additional indeterminacies in some directions by fitting subspaces into the sample sets [4]. Our contribution here is different: We do not directly consider the data space as subset of $\mathbb{R}^n$. Instead we consider a set of subspaces, which for example could have been extracted from the experiment itself. Our goal is to find clusters within this set of subspaces. The space of all subspaces is known as the Grassmann manifold, so we call our clustering algorithm Grassmann clustering. Figure 1 illustrates the difference between standard k-means and Grassmann clustering — clearly k-means fails to detect the structure of the time series, whereas the relaxed conditions of Grassmann clustering allow for a more precise fit of the data set.

2. PARTITIONAL CLUSTERING

In the literature Many algorithms for clustering are discussed. In the following, we will study clustering within the framework of k-means [2].

In general, its goal can be described as follows: Given a set A of points in some metric space (M,d), find a partition of A into disjoint non-empty subsets B_i, $\bigcup_i B_i = A$, together with *centroids* $\mathbf{c}_i \in M$ so as to minimize the sum of the squares of the distances of each point of A to the centroid $\mathbf{c}_i$ of the cluster B_i containing it. In other words, minimize

$$E(B_1,\mathbf{c}_1,\ldots,B_k,\mathbf{c}_k) := \sum_{i=1}^{k}\sum_{\mathbf{a}\in B_i} d(\mathbf{a},\mathbf{c}_i)^2. \tag{1}$$


(a) Toy data sequence (dynamic system)

(b) Standard k-means

(c) Data hyperplanes

(d) Cluster centers

Figure 1: Illustration of the differences of standard k-means clustering and Grassmann clustering. The hyperplanes for the Grassmann clustering are the spanned by each of 3 consecutive samples of the sequence.

If the set A contains only finitely many elements $\mathbf{a}_1,\ldots,\mathbf{a}_N$, then this can be easily re-formulated as constrained non-linear optimization problem: minimize

$$E(\mathbf{W},\mathbf{C}) := \sum_{i=1}^{k}\sum_{t=1}^{T} w_{it}\, d(\mathbf{a}_i,\mathbf{c}_i)^2. \tag{2}$$

subject to

$$w_{it} \in \{0,1\}, \sum_{i=1}^{k} w_{it} = 1 \quad \text{for} \quad 1 \le i \le k, 1 \le t \le T. \tag{3}$$

Here $\mathbf{C} := \{\mathbf{c}_1,\ldots,\mathbf{c}_k\}$ are the centroid locations, and $\mathbf{W} := (w_{it})$ is the partition matrix corresponding to the partition B_i of A.

A common approach to minimizing (2) subject to (3) is partial optimization for $\mathbf{W}$ and $\mathbf{C}$, i.e. alternating minimization of either $\mathbf{W}$ and $\mathbf{C}$ while keeping the other one fixed. The batch k-means algorithm employs precisely this strategy: After an initial, random choice of centroids $\mathbf{c}_1,\ldots,\mathbf{c}_k$, it iterates between the following two steps until convergence measured by a suitable stopping criterion:

- *cluster assignment*: $\mathbf{a}_t$ determine an index $i(t)$ such that

$$i(t) = \operatorname{argmin}_i d(\mathbf{a}_t,\mathbf{c}_i) \tag{4}$$

- *cluster update*: within each cluster $B_i := \{\mathbf{a}_t | i(t) = i\}$ determine the centroid $\mathbf{c}_i$ by minimizing

$$\mathbf{c}_i := \operatorname{argmin}_{\mathbf{c}} \sum_{\mathbf{a}\in B_i} d(\mathbf{a},\mathbf{c})^2 \qquad (5)$$

The cluster assignment step corresponds to minimizing (2) for fixed $\mathbf{C}$, which means choosing the partition $\mathbf{W}$ such that each element of A is assigned to the i-th cluster if $\mathbf{c}_i$ is the closest centroid. In the cluster update step, (2) is minimized for fixed partition $\mathbf{W}$, implying that $\mathbf{c}_i$ is constructed as centroid within the i-th cluster; this indeed corresponds to minimizing $E(\mathbf{W},\mathbf{C})$ for fixed $\mathbf{W}$ because in this case the cost function is a sum of functions depending different parameters, so we can minimize them separately leading to the centroid equation (5). This general update rule converges to a local minimum under rather weak conditions [3, 8].

An important special case is given by $M := \mathbb{R}^n$ and the Euclidean distance $d(\mathbf{x},\mathbf{y}) := \|\mathbf{x}-\mathbf{y}\|$. The centroids from equation (5) can then be calculated in closed form, and each centroid is simply given by the cluster mean $\mathbf{c}_i := (1/|B_i|)\sum_{\mathbf{a}\in B_i}\mathbf{a}$; this follows directly from

$$\sum_{\mathbf{a}\in B_i} \|\mathbf{a}-\mathbf{c}_i\|^2 = \sum_{\mathbf{a}\in B_i}\sum_{j=1}^{n}(a_j - c_{ij})^2 = \sum_{j=1}^{n}\sum_{\mathbf{a}\in B_i}(a_j^2 - 2a_j c_{ij} + c_{ij}^2),$$

which can be minimized separately for each coordinate j and is minimal with respect to c_{ij} if the derivative of the quadratic function is zero, so if $|B_i| c_{ij} = \sum_{\mathbf{a}\in B_i} a_j$.

In the following, we are interested in more complex metric spaces. Typically, k-means can be implemented efficiently, if the cluster centroids can be calculated quickly. In the example of $\mathbb{R}^n$, we saw that it was crucial to use minimize the square distances and to use the Euclidean distance. Hence we will study metrics which also allow a closed-form centroid solution.

The data space of interest will consist of subspaces of $\mathbb{R}^n$, and the goal is to find subspace clusters. We will only be dealing with sub-vector-spaces; extensions to the affine case are discussed in section 4.3.

A somewhat related method is the so-called k-plane clustering algorithm [4], which does not cluster subspaces but solves the problem of fitting hyperplanes in $\mathbb{R}^n$ to a given point set $A \subset \mathbb{R}^n$. A hyperplane $H \subset \mathbb{R}^n$ can be described by $H = \{\mathbf{x}|\mathbf{c}^\top\mathbf{x} = 0\} = \mathbf{c}^\perp$ for some normal vector $\mathbf{c}$, typically chosen such that $\|\mathbf{c}\| = 1$. Bradley and Mangasarian [4] essentially choose the pseudo-metric $d(\mathbf{a},\mathbf{b}) := |\mathbf{a}^\top\mathbf{b}|$ on the sphere $S^{n-1} := \{\mathbf{x} \in \mathbb{R} | \|\mathbf{x}\| = 1\}$ — the data can be assumed to lie on the sphere after normalization, which does not change cluster containment. They show that the centroid equation (5) is solved by any eigenvector of the *cluster correlation* $B_i B_i{}^\top$ corresponding to the minimal eigenvalue, if by abuse of notation B_i is to indicate the $(n \times |B_i|)$-matrix containing the elements of the set B_i in its columns. Alternative approaches to this subspace clustering problem are reviewed in [7].

3. PROJECTIVE CLUSTERING

A first step towards general subspace clustering is to consider one-dimensional subspace i.e. lines. Let $\mathbb{RP}^n$ denote the space of one-dimensional real vector subspaces of $\mathbb{R}^{n+1}$. It is equivalent to S^n after identifying antipodal points, so it has the quotient representation $\mathbb{RP}^n = S^n/\{-1,1\}$. We will represent lines by their equivalence class $[\mathbf{x}] := \{\lambda\mathbf{x} | \lambda \in \mathbb{R}^n\}$ for $\mathbf{x} \neq 0$. A metric can be defined by

$$d_0([\mathbf{x}],[\mathbf{y}]) := \sqrt{1 - \left(\frac{\mathbf{x}^\top\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}\right)^2} \qquad (6)$$

Clearly d is symmetric, and positive definite according to the Cauchy-Schwartz's inequality.

Conveniently, the cluster centroid of cluster B_i is given by any eigenvector of the cluster correlation $B_i B_i{}^\top$ corresponding to the largest eigenvalue. In section 4.1, we will show that projective clustering is a special case of a more general clustering and hence the derivation of the corresponding centroid clustering algorithm will be postponed until later.

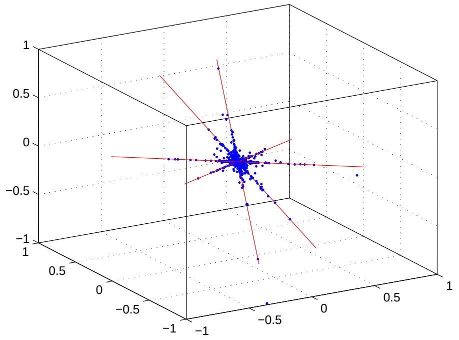


Figure 2: Illustration of projective k-means clustering in three dimensions. 10^5 samples from a 4-dimensional strongly supergaussian distribution are projected onto three dimensions and serve as the generators of the lines. These were nicely clustered into $k = 4$ centroids, located at the density axes.

Figure 2 shows an example application of the projective k-means algorithm. Note that the projective k-means can be directly applied to the dual problem of clustering hyperplanes by using the description via their normal 'lines'.

4. GRASSMANN CLUSTERING

More interestingly, we would like to perform clustering in the *Grassmann manifold* $\mathbb{G}_{n,p}$ of p-dimensional vector subspaces of $\mathbb{R}^n$ for $0 \leq p \leq n$. If $\mathbb{V}_{n,p}$ denotes the *Stiefel manifold* consisting of orthonormal matrices for $n \geq p$, then $\mathbb{G}_{n,p}$ has the natural quotient representation $\mathbb{G}_{n,p} = \mathbb{V}_{n,p}/\mathbb{O}_p$, where $\mathbb{O}_p := \mathbb{V}_{p,p}$ denotes the *orthogonal group*. This representation simply means that any p-dimensional subspace of $\mathbb{R}^n$ is given by p orthonormal vectors, i.e. by a basis $\mathbf{V} \in \mathbb{V}_{n,p}$, which is unique except for right multiplication by an orthogonal matrix. We will also write $[\mathbf{V}]$ for the subspace.

The geometric properties of optimization algorithms on $\mathbb{G}_{n,p}$ are nicely discussed by Edelman et al. [6]. They also summarize various metrics on the Grassmann manifold, which can all be naturally derived from the geodesic metric (arc length) induced by the natural Riemannian structure of $\mathbb{G}_{n,p}$. Some equivalence relations between the metrics are known, but for computational purposes, we choose the very easy to calculate so-called *projection F-norm* given by

$$d([\mathbf{V}],[\mathbf{W}]) := 2^{-1/2}\|\mathbf{V}\mathbf{V}^\top - \mathbf{W}\mathbf{W}^\top\|_F \qquad (7)$$

where $\|\mathbf{V}\|_F := \sqrt{\operatorname{tr}(\mathbf{V}\mathbf{V}^\top)}$ denotes the *Frobenius-norm* of a matrix. Note that the projection F-norm is indeed well-defined, as (7) does not depend on the choice of class representatives.

In order to perform k-means clustering on $(\mathbb{G}_{n,p}, d)$, we have to solve the centroid problem (5). One of our main results is that the centroid $[\mathbf{C}_i]$ of subspaces of some cluster B_i is spanned by p eigenvectors corresponding to the smallest eigenvalues of the *generalized cluster covariance* $(1/|B_i|)\sum_{[\mathbf{V}]\in B_i}\mathbf{V}\mathbf{V}^\top$. This generalizes the projective and the hyperplane k-means algorithm from above.

4.1 Calculating the optimal centroids

For the cluster update step of the batch k-means algorithm we need to find $[\mathbf{C}]$ such that

$$f(\mathbf{C}) := \sum_{i=1}^{l} d([\mathbf{V}_i],[\mathbf{C}])^2$$

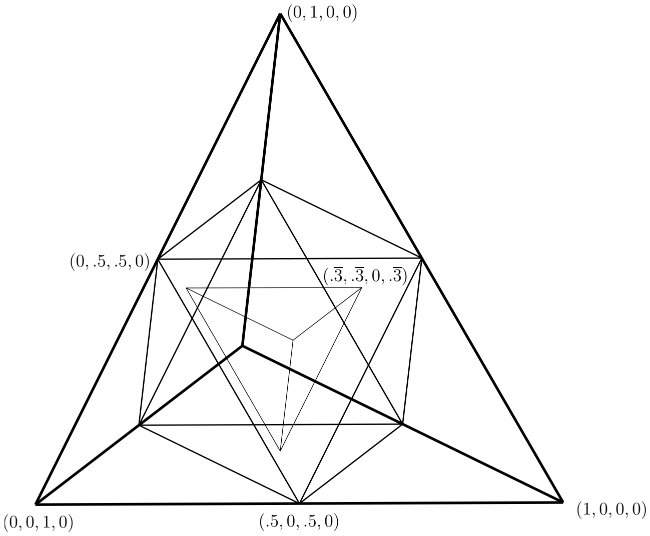


Figure 3: Illustration of the convex subsets on which the equation $\sum_{i=1}^{n} d_{ii}x_i$ for given $\mathbf{D}$ is optimized. Here $n=4$ and the surfaces for $p=1,\ldots,3$ are depicted (normalized onto the standard simplex).

for l subspaces $[\mathbf{V}_i]$ represented by $\mathbf{V}_i \in \mathbb{V}(n,p)$ is minimal, subject to $g(\mathbf{C}) := \mathbf{C}^\top\mathbf{C} = \mathbf{I}_p$ (pseudo orthogonality). We may also assume that the $\mathbf{V}_i$ are pseudo-orthonormal $\mathbf{V}_i{}^\top\mathbf{V}_i = \mathbf{I}_p$.

It is easy to see that:

$$\begin{aligned} f(\mathbf{C}) &= 2^{-1/2}\,\mathrm{tr}(\sum_i \mathbf{V}_i\mathbf{V}_i{}^\top) + \mathrm{tr}(l\mathbf{C}\mathbf{C}^\top - 2\mathbf{C}\mathbf{C}^\top \sum_i \mathbf{V}_i\mathbf{V}_i{}^\top) \\ &= 2^{-1/2}\,\mathrm{tr}\,\mathbf{D} + \mathrm{tr}((l\mathbf{I}_n - 2\mathbf{V})\mathbf{C}\mathbf{C}^\top) \end{aligned}$$

where

$$\mathbf{V} := \sum_i \mathbf{V}_i\mathbf{V}_i{}^\top \qquad \text{and} \qquad \mathbf{E}\mathbf{D}\mathbf{E}^\top = \mathbf{V}$$

denote the eigenvalue decomposition of $\mathbf{V}$ with $\mathbf{E}$ orthonormal and $\mathbf{D}$ diagonal. This means that

$$\begin{aligned} f(\mathbf{C}) &= 2^{-1/2}\sum_{i=1}^{n} d_{ii} + lp - 2\,\mathrm{tr}(\mathbf{D}\mathbf{E}^\top\mathbf{C}\mathbf{C}^\top\mathbf{E}) \\ &= 2^{-1/2}\sum_{i=1}^{n} d_{ii} + lp - 2\sum_{i=1}^{n} d_{ii}x_{ii} \end{aligned}$$

where d_{ij} are the matrix elements of $\mathbf{D}$, and x_{ij} of $\mathbf{X} = \mathbf{E}^\top\mathbf{C}\mathbf{C}^\top\mathbf{E}$.

Here $\mathrm{tr}\,\mathbf{X} = \mathrm{tr}(\mathbf{C}\mathbf{C}^\top) = p$ for pseudo orthogonal $\mathbf{C}$ (p eigenvectors $\mathbf{C}$ with eigenvalue 1) and all $0 \le x_{ii} \le 1$ (again pseudo orthogonality). Hence this is a linear optimization problem on a convex set (see also figure 3) and therefore any optimum is located at the corners of the convex set, which in our case are $\{x \in \{0,1\}^n \mid \sum_{i=1}^{n} x_i = p\}$. If we assume that the d_{ii} are ordered in descending order, then a minimum of f is given by

$$\mathbf{C}\mathbf{C}^\top = \mathbf{E}\mathbf{X}\mathbf{E}^\top = \mathbf{E}\begin{pmatrix} \mathbf{I}_p & 0 \\ 0 & 0 \end{pmatrix}\mathbf{E}^\top,$$

which corresponds to

$$\mathbf{C} = \begin{pmatrix} \mathbf{I}_p \\ 0 \end{pmatrix}.$$

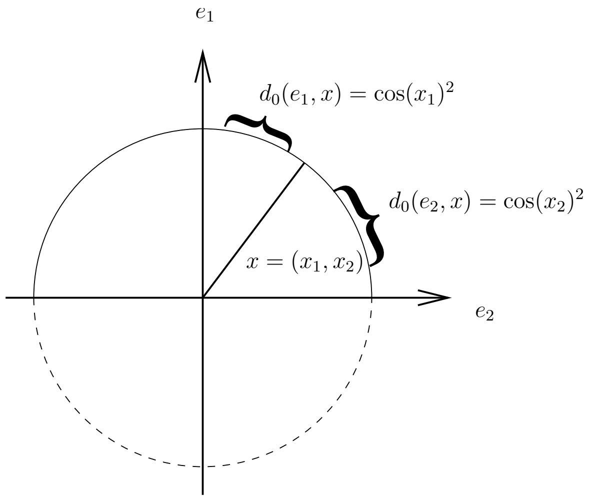


Figure 4: Let $\mathbf{V}_i$ be two samples which are orthogonal (w.l.o.g. we can assume $\mathbf{V}_i = e_i$ represented by the unit vectors). Hence $\mathbf{V} = \sum \mathbf{V}_i\mathbf{V}_i{}^\top$ has degenerate eigenstructure. Then the quantisation error is given by $d(e_1,x)^2 + d(e_2,x)^2$ which is here $\frac{1}{2}(2+2-2\,\mathrm{tr}\,\mathbf{I}\mathbf{X}\mathbf{X}^\top) = 2 - x_1^2 - x_2^2 = 1$ for $\mathbf{X}$ represented by $x = (x_1,x_2)$. Hence any $\mathbf{X}$ is a centroid in the sense of the batch k-means algorithm.

In this calculation we can also see the indeterminacies of the optimization:

1. If two or more eigenvalues of $\mathbf{V}$ are equal, any point on the corresponding edge of the convex set is optimal and hence the centroid can vary along the subspace generated by the corresponding eigenvectors $\mathbf{E}$
2. If some eigenvalues of $\mathbf{V}$ are zero, a similar indeterminacy occurs.

An example in $\mathbb{RP}^2$ is demonstrated in figure 4.

4.2 Relationship to projective clustering

The distance d_0 on $\mathbb{RP}^n$ from above (equation (6)) was defined as

$$d_0(\mathbf{V},\mathbf{W}) = \sqrt{1 - \left(\frac{V^\top W}{\|V\|\|W\|}\right)^2},$$

if according to our previous notation $[\mathbf{V}],[\mathbf{W}] \in \mathbb{G}_{n,1} = \mathbb{RP}^n$. Note that if the two vectors represent time series, then this is the same as the correlation between the two.

It is now easy to see that this distance coincides with the definition of d on the general Grassmannian from above. Let $V,W \in \mathbb{V}(n,1)$ be two vectors. We may assume that $\mathbf{V}^\top\mathbf{V} = \mathbf{W}^\top\mathbf{W} = 1$. Then

$$\begin{aligned} 2d(\mathbf{V},\mathbf{W})^2 &= \mathrm{tr}(\mathbf{V}\mathbf{V}^\top + \mathbf{W}\mathbf{W}^\top - \mathbf{V}\mathbf{W}^\top - \mathbf{W}\mathbf{V}^\top) \\ &= \mathrm{tr}(\mathbf{V}\mathbf{V}^\top) + \mathrm{tr}(\mathbf{W}\mathbf{W}^\top) - 2\,\mathrm{tr}(\mathbf{V}(\mathbf{V}^\top\mathbf{W})\mathbf{W}^\top) \end{aligned}$$

All matrices have rank 1 and hence the trace is the sole nonzero eigenvalue. Since $\mathbf{V}\mathbf{V}^\top\mathbf{V} = \mathbf{V}$ it is 1 for the first matrix, similar for the second and $\mathbf{W}^\top\mathbf{V}$ for the third, because $\mathbf{V}\mathbf{W}^\top\mathbf{V} = (\mathbf{W}^\top\mathbf{V})\mathbf{V}$. Hence

$$\begin{aligned} 2d(\mathbf{V},\mathbf{W})^2 &= 2 - 2(\mathbf{W}^\top\mathbf{V})^2 \\ &= 2d_0(\mathbf{V},\mathbf{W})^2. \end{aligned}$$

4.3 Dealing with affine spaces

So far we only have dealt with the special case of clustering subspaces, i.e. linear subsets which contain the origin. But in practice the problem of clustering affine subspaces arises, see for example 6. This can be dealt with quite easily.

Let F be a p dimensional affine linear subset of $\mathbb{R}^n$. Then F can be characterized by $p+1$ points $v_0,\dots,v_p$ such that $v_1-v_0,\dots,v_n-v_0$ are linearly independent. Consider the following embedding

$$\mathbb{R}^n \to \mathbb{R}^{n+1} : (x_1,\dots,x_n) \mapsto (x_1,\dots,x_n,1).$$

We may therefore identify the p dimensional affine subspaces with the $p+1$ linear subspaces in $\mathbb{R}^{n+1}$ by embedding the generators and taking the linear closure. In fact it is easy to see that we obtain a 1-to-1 mapping between the p dimensional affine subspaces of $\mathbb{R}^n$ and the $p+1$ dimensional linear subspaces in $\mathbb{R}^{n-1}$, which intersect the orthogonal complement of $(0,\dots,0,1)$ only at the origin.

Hence we can reduce the affine case to calculations for linear subsets only. Note that since only eigenvectors of sums of projections onto the subsets $\mathbf{V}_i$ can become centroids in the batch version of the k-means algorithm, any centroid is also in the image of the above embedding and can be identified uniquely with a affine subspace of the original problem.

4.4 Convergence and computational complexity

Since the algorithm uses the well understood framework of (batch) k-means calculation, it is very easy to see that it also inherits the convergence properties [3]. Hence convergence after finite steps is guaranteed. The only difference lies in the centroid calculation. Therefore in each step we have to calculate k eigenvalue decompositions of a symmetric matrix — is also guaranteed to succeed.

For complexity considerations we note that the eigenvalue decomposition is in the worst case an $O(n^3)$ operation and a worst case upper bound for iterations of the k-means algorithm is of $O(l^k)$, where l is the number of samples [5]. Hence the complexity is usually by a factor n^2 higher than with the standard k-means algorithm.

In practice however the algorithm converges after only a few iterations and we can employ restart techniques to avoid local minima.

5. EXPERIMENTAL RESULTS

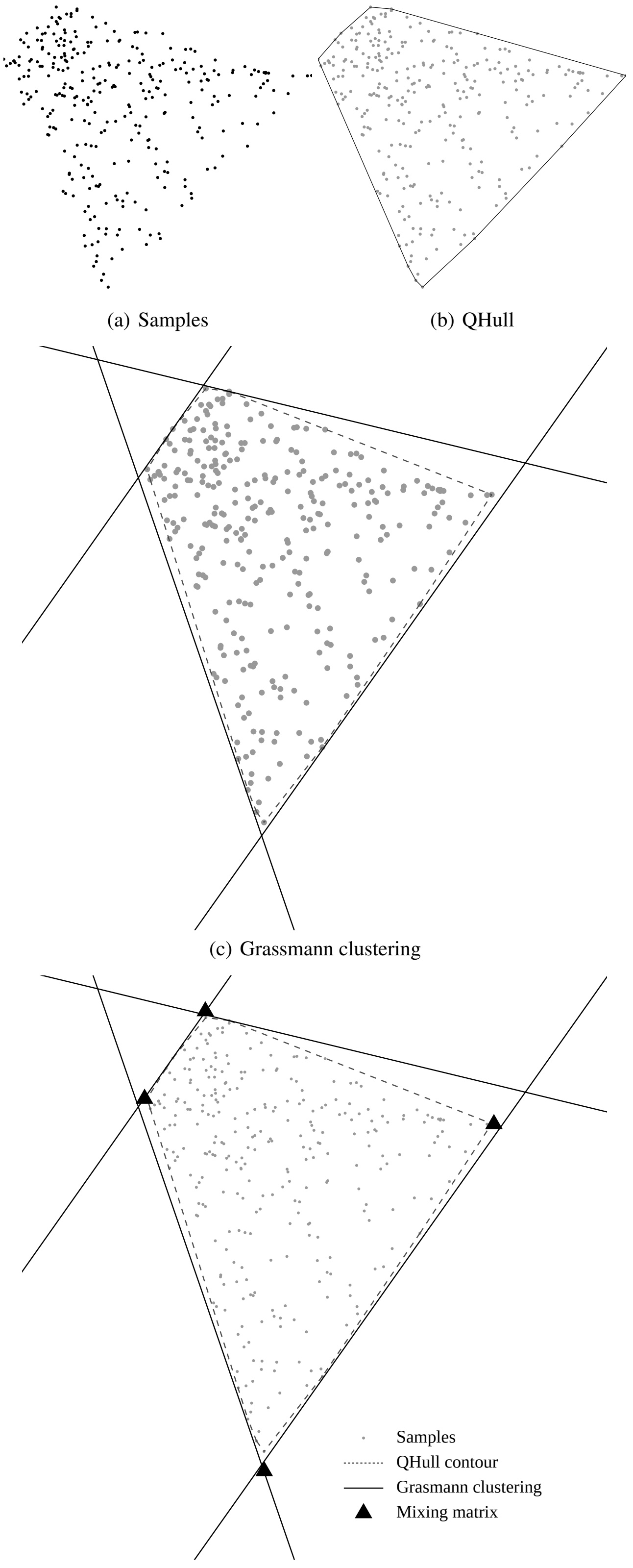
We finish by illustrating the algorithm in a few examples.

5.1 Toy example

As a toy example, let us first consider 10^4 samples of $\mathbb{G}_{4,2}$, namely uniformly randomly chosen from the 6 possible 2-dimensional coordinate planes. In order to avoid any bias within the algorithm, the non-zero coefficients from the plane-representing matrices have been chosen uniformly from $\mathbb{O}_2$. The samples have been deteriorated by Gaussian noise with a signal-to-noise ratio of $10dB$. Application of the Grassmann k-means algorithm with $k=6$ yields convergence after only 6 epochs with the resulting 6 clusters with centroids $[\mathbf{V}^i]$. The distance measure $\mu(\mathbf{V}) := (|v_{i1}+v_{i2}|+|v_{i1}-v_{i2}|)_i$ should be large only in two coordinates if $[\mathbf{V}]$ is close to the corresponding 2-dimensional coordinate plane. And indeed, the found centroids have distance measures $\mu(\mathbf{V}^i) =$

$$\begin{pmatrix}0.02\\0\\1.9\\1.9\end{pmatrix},\begin{pmatrix}1.7\\0.01\\0.01\\1.7\end{pmatrix},\begin{pmatrix}1.7\\0.01\\1.7\\0.02\end{pmatrix},\begin{pmatrix}0.01\\1.5\\1.5\\0\end{pmatrix},\begin{pmatrix}2.0\\2.0\\0\\0.01\end{pmatrix},\begin{pmatrix}0.01\\2.0\\0.01\\2.0\end{pmatrix}.$$

Hence, the algorithm correctly chose all 6 coordinate planes as cluster centroids.



(a) Samples

(b) QHull

(c) Grassmann clustering

(d) Result

Figure 5: An example of using hyperplane clustering ($p=n-1$) to identify the contour of a samples figure. QHull was used to find the outer edges then those are clustered into 4 clusters. The broken lines show the boundaries use to generate the 300 samples.

5.2 Polytope identification

As an example application of the Grassmann clustering algorithm, we want to solve the following approximation problem from computational geometry: given a set of points, identify the smallest convex polytope with a fixed number of faces k, containing the points. In two dimensions, this implies the task of finding the k edges of a polytope where only samples in the inside are known. We use QHull algorithm [1] to construct the convex hull thus identifying the possible edges of the polytope. Then, we apply affine Grassmann k-means clustering to these edges in order to identify the k bounding edges. Figure 5 shows an example. Generalization to arbitrary dimensions are straight-forward.

5.3 Nonnegative Matrix Factorization

(Overcomplete) Nonnegative Matrix Factorization (NMF) deals with the problem of finding a nonnegative decomposition $\mathbf{X} = \mathbf{AS} + \mathbf{N}$ of a nonnegative matrix $\mathbf{X}$, where $\mathbf{N}$ denotes unknown Gaussian noise. $\mathbf{S}$ is often pictured as a source data set containing samples along its columns. If we assume that $\mathbf{S}$ spans the whole first quadrant, then $\mathbf{X}$ is a conic hull with cone lines given by the columns of $\mathbf{A}$. After projection to the standard simplex, the conic hull reduces to the convex hull, and the projected, known mixture data set $\mathbf{X}$ lies within a convex polytope of the order given by the number of rows of $\mathbf{S}$. Hence we face the problem of identifying edges of a sampled polytope, and, even in the overcomplete case, we may tackle this problem by the Grassmann clustering-based identification algorithm from the previous section.

As an example, see figure 6, we choose a random mixing matrix

$$\mathbf{A} = \begin{pmatrix} 0.76 & 0.39 & 0.14 \\ 0.033 & 0.06 & 0.43 \\ 0.20 & 0.56 & 0.43 \end{pmatrix}$$

and sources $\mathbf{S}$ given by i.i.d. samples from a squared gaussian. 10^5 samples were drawn, and sample subsets containing 10 to 10^5 points where used for the comparison. We refer to the figure caption for further details.

6. CONCLUSION

We have studied k-means-style clustering problems on the non-Euclidean Grassmann manifold. In an adequate metric, we were able to reduce the arising centroid calculation problem to the calculation of eigenvectors of the cluster covariance, for which we gave a proof based on convex optimization. The algorithm was illustrated by applications to polytope fitting and to performing overcomplete nonnegative factorizations similar to NMF. In future work, besides extending the framework to other clustering algorithms and matrix manifolds together with proving convergence of the resulting algorithms, we plan on applying the algorithm for the stability analysis of multidimensional independent component analysis.

Acknowledgements

Partial financial support by the DFG (GRK 638) and the BMBF (project 'ModKog') is gratefully acknowledged.

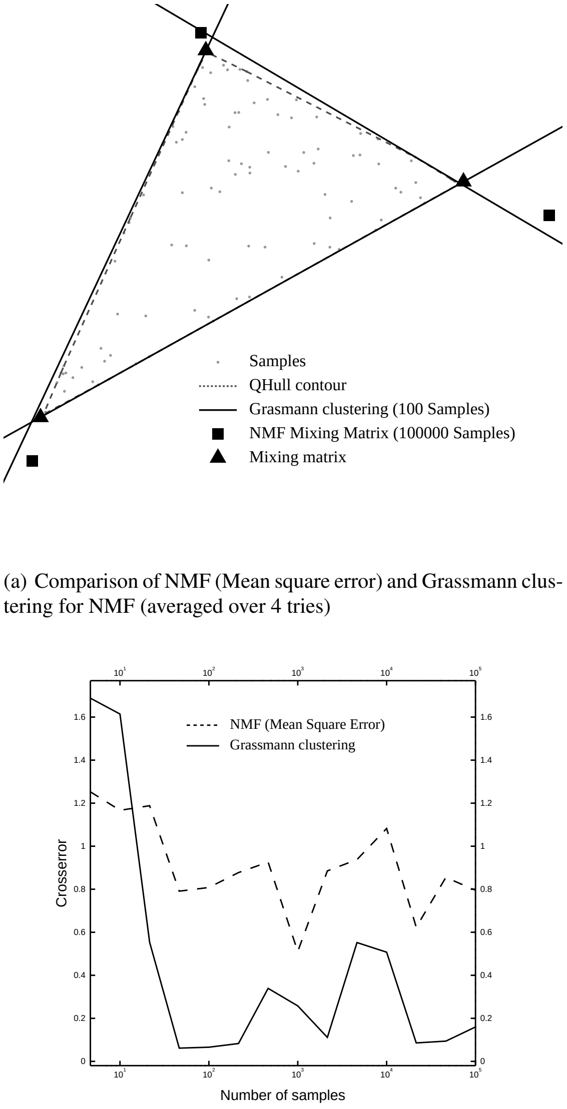
REFERENCES

[1] C.B. Barber, D.P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hull. Technical Report GCG53, The Geometry Center, University of Minnesota, Minneapolis, 1993.

[2] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] L. Bottou and Y. Bengio. Convergence properties of the k-means algorithms. In *Proc. NIPS 1994*, pages 585–592. MIT Press, 1995.

[4] P.S. Bradley and O.L. Mangasarian. k-plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000.



(a) Comparison of NMF (Mean square error) and Grassmann clustering for NMF (averaged over 4 tries)

(b) Illustration of the NMF algorithm: projection onto the standard simplex

Figure 6: Grassmann clustering can be used to solve the NMF problem. The mixed signal to be analyzed is a 3-dimensional toy signal with a positive 3×3 matrix. The resulting mixture was analyzed with a mean square error implementation of NMF and compared to Grassmann clustering. In the clustering algorithm the data is first projected onto the standard simplex. This translates the task to the polytope identification discussed in section 5.2.

[5] Sanjoy Dasgupta. How fast is k -means? *Computational Learning Theory*, 2777:735, 2003.

[6] A. Edelman, T.A. Arias, and S.T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1999.

[7] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newsl.*, 6(1):90–105, 2004.

[8] S.Z. Selim and M.A. Ismail. K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:81–87, 1984.