IIT@MIT, Massachussets Institute of Technology

Center for Biological and Computational Learning

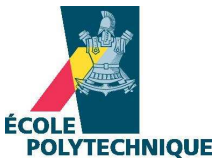
Research internship report

Learning data representation and piece-wise estimation

Author:
Sammy El Ghazzal

Supervisor:
Dr. Lorenzo Rosasco

August 28, 2012



Contents

Acknowledgment

This thesis could not have been written without the help of Dr. Lorenzo Rosasco and Dr. Guillermo D. Canas, who were not only my supervisors but also encouraged and challenged me through my research internship. Their infinite patience and availability have been of great help whenever I needed to be guided in my research, to get suggestions and ideas for the problems I was tackling, or simply to ask questions on specific points.
I also wish to thank Youssef Mroueh for checking some of the proofs of this report and Andrea Tacchetti for his valuable experience in computing.
I am truly grateful to Pr. Tomaso Poggio for having given me the opportunity to work in the laboratories of the Center for Biological Computational Learning and to the Istituto Italiano di Tecnologia, who provided me with the financial support to pursue this internship.
Lastly, I want to thank Kathleen Sullivan and Gadi Geiger for their help in the administrative follow-up.

Introduction

Unsupervised learning is traditionally defined as the problem of learning structure, or patterns in the data [1]. The importance of unsupervised learning can be seen by the abundance of unlabeled data. Indeed, in a wide range of settings, unlabeled data is much more common than labeled data, and may be vastly cheaper to acquire [2]. Additionally, while the algorithms for supervised learning are relatively mature [3, 4], a lot of work remains to be done on unsupervised learning algorithms. Much of the current work in machine learning has indeed focused on the unsupervised preprocessing of the data. Recent work such as HMAX [5], Super-Vector Coding [6], or Deep Belief Networks [7, 8], perform a hierarchical preprocessing of the unlabeled data, and produce an encoding that can be more easily classified using a standard classifier (typically a Support Vector Machine [9]). Since the classification stage is performed using standard algorithms, their state-of-the-art performance stems from an effective encoding/representation of the data.

One of the motivations for unsupervised learning has been in attempting to overcome, or alleviate the *curse of dimensionality* [10], in which the sample complexity of supervised learning, or classification algorithms typically depends exponentially on the dimension of the input. In cases in which the data may be high-dimensional, but is generated by a distribution with support on a lower-dimensional space (e.g. a low-dimensional manifold), a low-dimensional encoding of the data that preserves the metric structure of the original data (such as pairwise distances), can be used in place of the original data, possibly resulting in a lower sample complexity. Learning such a low-dimensional encoding, under the manifold input assumption, has been termed *manifold learning*, and there has been a large amount of interest in this topic in recent years [11, 12, 13, 14, 15, 16].

In this report, we study the problem of learning an efficient representation of unsupervised data, under the assumption that the data is generated from a distribution with support on a manifold (of dimension lower than that of the embedding space). Following [17], we extend the unsupervised learning framework first suggested by Maurer [18], in which a wide range of standard algorithms, such as k-means, PCA, sparse coding..., can be studied. In this setting, an unsupervised algorithm performs a minimization over a training set, while its performance is evaluated on a test data set. The quantity that is minimized is a measure of geometric approximation to the training set, which is minimal whenever the learned output representation is very close to the data.

Using this framework, we study k-means, the most widely-used algorithm for unsupervised learning, and k-flats, an unsupervised algorithm similar in its principle to k-means, but that represents data by a collection of affine spaces instead of points.

In this context, three main contributions of our work are:

1. **Kernel extensions**: we discuss the kernel extensions of these algorithms. Whereas kernel k-means has been studied [19], we propose a kernel extension of k-flats, *kernel k-flats*, which to the best of our knowledge has not been done before.

2. **Adaptive choice of parameter**: while Maurer [18] studies k-means for a fixed value of the parameter k, we analyze the performance of k-means and k-flats as functions of their free parameters. Indeed, while the dependence of the dimensionality of the output of k-means and k-flats on their free parameters is simple and well-understood, their performance in representing the data, for varying values of these parameters, is not well-understood. Indeed, the empirical error can be made arbitrarily low with a simple choice of the free parameters (such as setting the value of k in k-means to equal the training set size). However, a good approximation of the training set does not, in general, imply a similarly

good approximation of the test set. Indeed, lower, and particularly upper bounds [20, 21, 18] on the reconstruction performance suggest that there is a non-trivial choice of parameters that produces the best possible test set performance, a fact that may have important practical implications. Therefore, the problem of choosing the free parameters optimally as to minimize the *test set* performance naturally arises. Drawing ideas from [22], we propose a method based on complexity regularization to choose the number of means in k-means, and prove that this choice is relevant.

3. **Connections to other settings**: we show the connections between the k-means problem and well-studied problems: in particular, we show how the measure of risk we use relates to the notion of distortion in optimal quantization and to the cost of clustering when it is measured by summing intra-cluster distances.

The remainder of this report is organized as follows: in section 1, we recall the computational aspects of k-means and k-flats. In section 2, we discuss kernel extensions of k-means and k-flats. In section 3, we introduce the formal setting we use to analyze the statistical properties of k-means and k-flats and show the connections of k-means to existing problems. In section 4, we propose a method based on complexity regularization to choose the value of k in k-means. In section 5, we discuss experimental results.

1 Background

k-means and, to a lesser extent, k-flats are two widely used algorithms in unsupervised learning. We study in this section the computational aspects of these two algorithms before tackling their statistical analysis and specifically their asymptotic behavior.

Assume given a separable Euclidean space $\mathcal{X} = \mathbb{R}^d$, with inner product denoted by $\langle \cdot, \cdot \rangle$. We denote by $\|.\|$ the norm on $\mathcal{X}$, and by $d_{\mathcal{X}}$ the distance it induces. Further, assume that we are given data in the form of a set $X_n = (x_1, \ldots, x_n) \in \mathcal{X}^n$. Using this set, we define an empirical measure of risk called *empirical reconstruction error* as follows:

Definition 1 (Empirical reconstruction error)
Given a closed set S, the empirical reconstruction error of S is defined by:

$$\hat{\mathcal{E}}(S) = \frac{1}{n} \sum_{i=1}^{n} d^2_{\mathcal{X}}(x_i, S). \tag{1}$$

The algorithms that we consider minimize the error of equation (1) when restricting the minimization over sets in a certain class $\mathcal{H}$ called the *hypothesis space*[1], that is they compute the set $\hat{S}$ such that:

$$\hat{S} = \arg\min_{S \in \mathcal{H}} \hat{\mathcal{E}}(S). \tag{2}$$

The basic framework being set, we now present the two algorithms of interest, namely k-means and k-flats. We make particular emphasis on the computational aspects by providing pseudo-code, and theoretical justifications of the correctness of the algorithms.

[1] We will see in section 3.1 that this space plays an important role through the constraints it imposes on the problem.

1.1 k-means

k-means is one of the first and probably one of the most widely used algorithm for unsupervised learning tasks. It has been well-studied both from a theoretical [18, 23] and a practical point of view [24, 25]. We begin with a presentation of the algorithm introduced by Lloyd [26] and provide justifications to the way it is computed.

1.1.1 Algorithm

k-means is an algorithm whose input is a data set X_n and an integer k, and whose output is a set of k points (see figure 1). Given a value of k fixed a priori[2], k-means can be seen as minimizing the following objective function

$$\begin{aligned} \hat{\mathcal{E}}(S) &= \sum_{j=1}^{k} \sum_{i \in \mathcal{C}_j} \|x_i - \pi_S(x_i)\|^2 , \\ &= \sum_{i=1}^{n} \min_{1 \leq j \leq k} \|x_i - m_j\|^2 \text{ where } S = \{m_1, \ldots, m_k\} , \end{aligned} \tag{3}$$

over the following hypothesis space:

$$\mathcal{H} = \mathcal{P}_k \text{ where } \mathcal{P}_k \text{ is the class of sets of } k \text{ points,}$$

and where the projection π_S is defined for any closed set S by [3]:

$$\pi_S(x) = \arg\min_{s \in S} \|x - s\| . \tag{4}$$

In general, the k-means problem is NP-hard, even for $k = 2$ [27] and as soon as the ambient space has at least two dimensions [23]. Lloyd [26] proposed an algorithm that reaches a local minimum of the objective function. His algorithm alternates between two steps: given the assignments of data points to means, a minimization of the objective function over the means, and given the means, a minimization over the assignments.

To describe Lloyd's algorithm, we need the following notation:

$$\begin{aligned} C : \quad & [\![1, n]\!] \to [\![1, k]\!] \\ & i \mapsto j \Leftrightarrow \pi_S(x_i) = m_j, \end{aligned} \tag{5}$$

$$\mathcal{C}_j = \{i,\ C(i) = j\} , \tag{6}$$

$$n_j = \text{card}(\mathcal{C}_j). \tag{7}$$

C is the assignment vector and maps indices of data points to indices of closest means. $\mathcal{C}_j$ is the set of indices of data points that are closer to the j-th mean and n_j is the size of this set.

Algorithm 1 (Lloyd's algorithm)
Input: *a data set X_n, an integer k (number of means).*
Output: *set of k means.*

1. *Choose randomly the $(m_j)_{1 \leq j \leq k}$ among the $(x_i)_{1 \leq i \leq n}$ without replacement.*

[2] However, we will see in section 4 that it is possible to provide a choice of k a posteriori, for instance by using complexity regularization.

[3] The existence of the minimum in equation (4) follows from the convexity of the norm $\|\cdot\|$ and the closeness of S.

2. *Assignment update*[4]*:*

$$\forall i \in [\![1, n]\!],\ C(i) = \arg\min_{1 \leq j \leq k} \|x_i - m_j\|\,.$$

3. *Means update:*

$$\forall j \in [\![1, k]\!],\ m_j = \frac{\sum_{i \in \mathcal{C}_j} x_i}{n_j}.$$

4. *Iterate steps 2. and 3. until convergence.*

where convergence means that the assignment vector C does not change (and consequently, the means do not change either) from one iteration to the following.

Each iteration[5] (steps 2. and 3.) has a complexity $\mathcal{O}(knd)$ (an iteration needs to compute distances from the n data points to the k means, each compuatation of a distance taking a time proportional to the dimension d)[6].

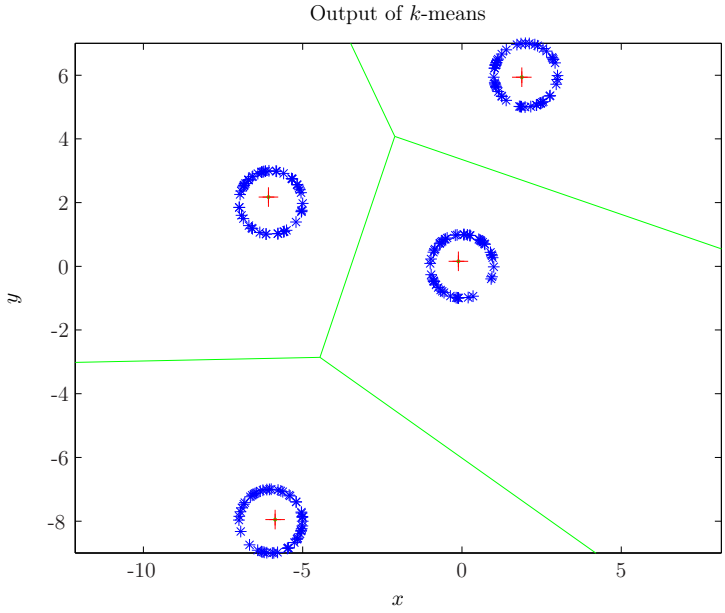


Figure 1: Output of k-means for k=4 on a set composed of four unitary circles (in blue). The red crosses are the means produced by k-means and the green lines delimit the Voronoi regions (see section 3.2.2) associated to the means.

1.1.2 Computational analysis

In this section, we prove that Lloyd's algorithm reaches a local minimum of the objective function (3) in a finite number of steps. We also explain why step 3. minimizes the objective function given the assignments.

[4]To manage cases where the minimum is reached for several indices, one can impose that the point is assigned to the mean with the smallest indice.

[5]We empirically see that between 3 and 10 iterations are needed to reach the convergence.

[6]In fact, some work [28] has been done to avoid the computation of all distances between data points and means.

The main point is to prove that given the assignment vector C, the means must be the centers of mass of the different group of points, which is done in lemma 1.

Lemma 1
The center of mass $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ *minimizes the sum of squared distances of a set of points* i.e.*:*

$$\overline{x} = \arg\min_{y \in \mathcal{X}} \sum_{i=1}^{n} \|x_i - y\|^2 .$$

Proof
The result follows by taking the derivative with respect to y:

$$\nabla_y \sum_{i=1}^{n} \|x_i - y\|^2 = 0 \Leftrightarrow 2\sum_{i=1}^{n} (y - x_i) = 0 \Leftrightarrow y = \frac{1}{n}\sum_{i=1}^{n} x_i = \overline{x}. \quad \blacksquare$$

Lloyd's algorithm is an iterative algorithm and thus, the question of whether it finishes in a finite time can be asked. The following proposition ensures that it does.

Proposition 1 (Convergence of Lloyd's algorithm)
Lloyd's algorithm converges in a finite number of steps.

Proof
The proof relies on two basic remarks:

- The empirical reconstruction error decreases at each step: indeed, step 2. of algorithm 1 makes the distance of each point to the closest code point $\|x_i - \pi_S(x_i)\|$ decrease, thus making the global objective function decrease. Step 3. allows the sum of inter-cluster distances $\sum_{i \in \mathcal{C}_j} \|x_i - m_j\|^2$ to decrease (cf lemma 1).
- There is only a finite number of ways of forming k groups with n data points. Indeed, this number is upper-bounded by n^k, which is the number of ways of forming at most k groups with n points.

The finite number of the ways of splitting the data points and the fact that each step makes the objective function decrease ensure the convergence. $\blacksquare$

Proposition 1 only ensures that Lloyd's algorithm finishes in a finite time and reaches a local minimum. It does not guarantee that the global optimum is reached. In fact, the local minimum reached at convergence can be arbitrarily far from the optimum.
In section 1.1.3, we describe k-means++, a particular initialization of k-means that addresses that provides a guarantee that the risk of the configuration reached at convergence is (in expectation) within a factor $\mathcal{O}(\ln k)$ of the best risk achievable.

1.1.3 k-means++

k-means++ is a special initialization of Lloyd's algorithm which had been proposed in [24]: it uses a careful seeding and this modification guarantees an upper bound on the proximity of the configuration reached at the convergence to the optimal configuration. Precisely [24]:

$$\mathbb{E}\left(\hat{\mathcal{E}}(S_{k++})\right) \leq 8(\ln k + 2)\hat{\mathcal{E}}(\hat{S}_k),$$

where the expectation is taken over the initializations (in particular not over the training set), S_{k++} is the set produced by the initialization of k-means++, and $\hat{S}_k$ the set optimizing empirical

risk of equation (2) over sets of size k. Note that this inequality holds as soon as the initialization is performed according to k-means++ (that is, even if Lloyd's algorithm is not run). If Lloyd's algorithm is run on top of it, this bound still holds as Lloyd's algorithm can only decrease the risk.

In this algorithm, the important step is the choice of the initial $(m_j)_{1\leq j\leq k}$. Here is how it proceeds:

Algorithm 2 (k-means++)
Input: *a data set X_n, an integer k (number of means).*
Output: *set of k means.*

1. *Randomly choose m_1 among the $(x_i)_{1\leq i\leq n}$.*
2. *$\forall j \in [\![2, k]\!]$ choose $m_j = x_i$ with probability*

$$\frac{D(x_i)^2}{\sum_{i=1}^{n} D(x_i)^2}, \tag{8}$$

   where $D(x)$ denotes the distance of x to the closest mean already found.
3. *(Optional) Run Lloyd's algorithm (see Algorithm 1) with the initialization previously obtained.*

The complexity of each iteration (step 2. and 3.) is[7] $\mathcal{O}(nd)$ and thus k-means++ without running Lloyd's algorithm has a complexity $\mathcal{O}(knd)$.

Remark 1 *Notice that, during the initialization, it is not possible that two means are placed on the same data point, that is $m_l \neq m_j$ for $l \neq j$. Indeed, at step $l \in [\![2, k]\!]$, if there is one of the mean $\{m_1, \ldots, m_{l-1}\}$ on a certain data point $x \in X_n$, the probability that m_l be x is zero (from equation (8)).*

Intuitively, the initialization provided k-means++ produces *a priori* better results than k-means because it places the means (with high probability) in such a way that it does not let a remote region of the training set without means and places (with high probability) more means in the regions that have weight. In this sense, it realizes a trade-off between the random initialization that places means mainly in regions with high weight and the ϵ-net initialization (see section 5.5 for the definition of the ϵ-net initialization) that places means in remote regions but does not take the weights of the regions into account.

1.2 k-flats

k-flats, although very similar in its principle to k-means, has been much less studied. It was introduced by Bradley and Mangasarian in [29] in an attempt to represent data distributions by affine subspaces[8] instead of points.

1.2.1 Algorithm

k-flats takes as input a data set $X_n \in \mathbb{R}^{d\times n}$, two integers k and m representing respectively the number of flats and the dimension of each flat. It ouputs a set of k flats of dimension m (see

[7] Indeed, contrary to Lloyd's algorithm, at step $l \in [\![2, k]\!]$, one does not need to compare the distance of a data point $x \in X_n$ to each of the l means: it is sufficient to compare $d_{\mathcal{X}}(x, m_l)$ and $d_{\mathcal{X}}(x, \{m_1, \ldots, m_{l-1}\})$.

[8] We use in this report the word m-flat to mention an affine space of dimension m.

figures 2 and 3). Given values of k and m chosen a priori, k-flats can be seen as minimizing

$$\begin{aligned}\hat{\mathcal{E}}(S) &= \sum_{j=1}^{k}\sum_{i\in\mathcal{C}_j}\|x_i-\pi_S(x_i)\|^2 \\ &= \sum_{i=1}^{n}\min_{1\le j\le k} d_{\mathcal{X}}(x_i,S)^2 \text{ where } S=\bigcup_{1\le j\le k}F_j,\end{aligned} \tag{9}$$

over the following hypothesis space

$$\mathcal{H}=\mathcal{F}_{k,m} \text{ where } \mathcal{F}_{k,m} \text{ is the class of sets of } k \text{ affine spaces of dimension } m \text{ each.}$$

and exactly like what happens with k-means, the user must specify the values of k and m before running the algorithm.
A practical algorithm, very similar to k-means, has been proposed in [30, 29]: it alternates between optimizing the flats given the assignments and updating the assignments given the flats. We present it in algorithm 3.

Algorithm 3
Input: *a data set X_n, two integers k (number of flats) and m (dimension of the flats).*
Output: *set of k affine spaces.*

1. *Initialize the assignment vector C.*
2. *$\forall j\in[\![1,k]\!]$ compute the $(F_j)_{1\le j\le k}$ by finding the best m-dimensional ($1\le m\le d$) flat (*i.e *the one that minimizes $\sum_{i\in\mathcal{C}_j} d_{\mathcal{X}}(x_i,F_j)^2$).*
3. *Assignment: assign each data point to the closest flat, that is*

$$\forall i\in[\![1,n]\!],\ C(i)=\arg\min_{1\le j\le k} d_{\mathcal{X}}(x_i,F_j).$$

4. *Repeat steps 2. and 3. until convergence.*

In the case $k=1$, algorithm 3 is Principal Component Analysis (PCA). As we will see in the next section, the relation between these two algorithms is even tighter, as algorithm 3 is basically obtained by running PCA independently on k groups of points.

1.2.2 How are flats computed ?

Given the assignments of the data points $C=(C(i))_{1\le i\le n}$, we can compute each flat *independently*, that is: for $j\in[\![1,k]\!]$, we consider the covariance matrix[9] restricted to the j-th group of points

$$Z_j=\tilde{X}_{n,j}\tilde{X}_{n,j}^\top \text{ where } \tilde{X}_{n,j}=((\tilde{X}_n)_{uv})_{1\le u\le d, v\in\mathcal{C}_j}\in\mathbb{R}^{d\times n_j}.$$

The j-th flat F_j is computed (see proposition 3 for the mathematical details and the justification of this result) as follows:

$$F_j=\frac{\sum_{i\in\mathcal{C}_j}x_i}{n_j}+\text{span}(e_1^{(j)},\dots,e_m^{(j)}),$$

[9]Here we have recentered the data *i.e.*, we replaced $X=(x_1,\dots,x_n)$ by $\tilde{X}_n$ such that $\tilde{X}_n=(x_1-\overline{x}_{C(1)},\dots,x_n-\overline{x}_{C(n)})$ with $\overline{x}_j=\frac{\sum_{i\in\mathcal{C}_j}x_i}{n_j}$.

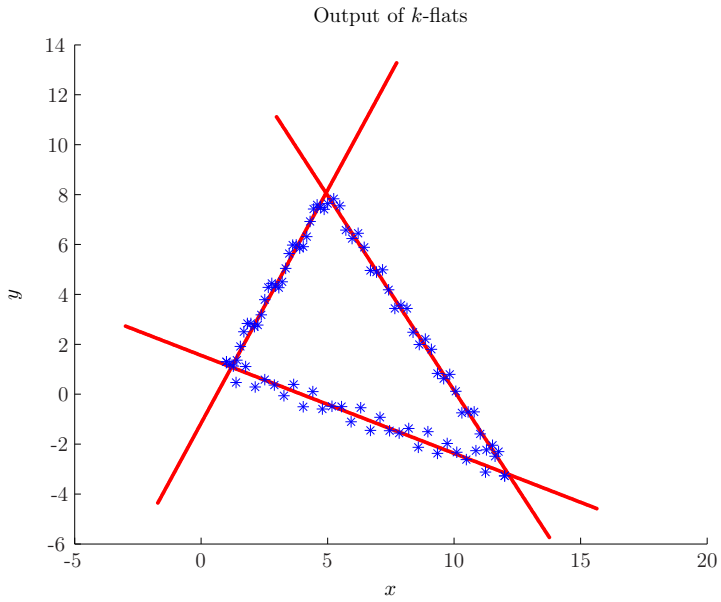


Figure 2: Output of k-flats (in red) with $k = 3$ and $m = 1$ on a triangle with some noise. The blue points compose the training set.

where the $(e_l^{(j)})_{1 \leq l \leq d}$ are the eigenvectors of Z_j associated respectively with eigenvalues $\lambda_1^{(j)} \geq \cdots \geq \lambda_d^{(j)}$, and where the formal addition of a point x and a vector space F is defined by

$$x + F = \{x + f, f \in F\}\,.$$

Notice that because of the symmetry of Z_j, the $(e_l^{(j)})_{1 \leq l \leq d}$ form an orthogonal basis and in the remainder of the section, we suppose that a renormalization has been performed, so that the basis is orthonormal.

The basic idea is that for a given group of points, the flat goes through the center of mass and is directed by the m principal components.

1.2.3 Computational analysis

In this section we provide a proof that the process mentioned in the previous section to compute the flat optimizes the sum of distances of data points to closest flats (equation (9)). For the sake of simplicity in the notation, we sketch the proof for each independent problem, that is we consider the data points of a certain group only $\{x_i, i \in \mathcal{C}_j\}$ for j between 1 and k. Most of the proofs are relegated in appendix B.2.

Using a similar argument as the one used in proposition 1, we can prove

Proposition 2 *Algorithm 3 converges in a finite number of steps.*

As the flats are affine spaces, we recall the link between the projection on an affine space and its direction (the vector space associated to it):

Lemma 2
If F is an affine space of direction G that is:

$$F = m + G,$$

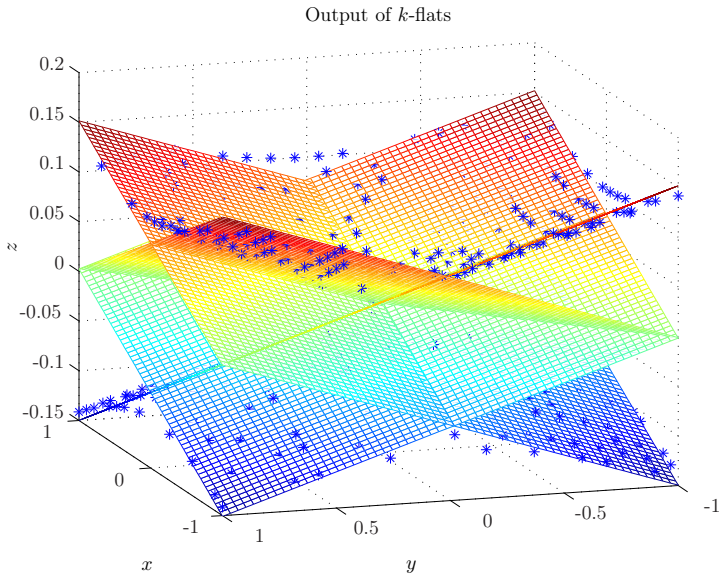


Figure 3: Output of k-flats (in red) with $k = 4$ and $m = 2$ on an elliptic paraboloid $x^2+y^2 = 10z^2$. The blue points compose the training set.

for $m \in F$ and G a vector space, then:

$$\forall x \in \mathcal{X},\ \pi_F(x) = m + \pi_G(x - m). \tag{10}$$

Proof
Let $x \in \mathcal{X}$.

$$\min_{f \in F} \|x - f\| \quad = \quad \|x - \pi_F(x)\| = \min_{g \in G} \|x - (m + g)\| = \left\| x - \underbrace{(m + \pi_G(x - m))}_{\in F} \right\|.$$

By uniqueness of the projection on a closed set, $\pi_F(x) = m + \pi_G(x - m)$. ∎

The following lemmas (3 and 4) justify the assertions we made in section 1.2.2 about the way flats are computed in practice.

Lemma 3
Let $X_n = (x_1, \ldots, x_n)$ be a data set ($X_n \in \mathbb{R}^{d \times n}$), m an integer between 1 and d, and $(g_1, \ldots, g_m)$ a linearly independent family that we will suppose orthonormal [10]*, $\mathcal{G} = span(g_1, \ldots, g_m)$ and $\mathcal{G}_y = y + \mathcal{G} = \{y + g, g \in \mathcal{G}\}$. The following problem:*

$$\begin{aligned} y^* &= \arg\min_y \sum_{i=1}^{n} d_{\mathcal{X}}(x_i, \mathcal{G}_y)^2 \\ &= \arg\min_y \sum_{i=1}^{n} \left(\|x_i - y\|^2 - \sum_{l=1}^{m} \langle x_i - y, g_l \rangle^2 \right), \end{aligned}$$

admits as a solution the center of mass of the $(x_i)_{1 \leq i \leq n}$, i.e*:*

$$y^* = \frac{\sum_{i=1}^{n} x_i}{n} = \bar{x}.$$

The minimum is not unique: every point from the affine space $\overline{x} + \mathcal{G}$ is a solution.

[10] This assumption is not restrictive because each set of linearly independent vectors can be made orthonormal (for instance by using Gram Schmidt algorithm).

Lemma 4
Let X_n be a data set ($X_n \in \mathbb{R}^{d\times n}$), m an integer between 1 and d, and $Z = \tilde{X}_n \tilde{X}_n^\top$ where $\tilde{X}_n$ is the recentered data set ($\tilde{X}_n = (x'_1, \dots, x'_n)$ with $x'_i = x_i - \overline{x}$). The set of m vectors $(f_1, \dots, f_m)$ (we will suppose that the family is orthonormal) that maximizes the projection on the space it generates $F = span(f_1, \dots, f_m)$ is exactly the set of the m eigenvectors associated with the largest eigenvalues of Z.

More precisely, if $(e_1, \dots, e_d)$ is a set of eigenvectors of Z associated respectively to $(\lambda_1, \dots, \lambda_d)$ with $\lambda_1 \geq \dots \geq \lambda_d$, then the solution to

$$\begin{aligned}(f_1^*, \dots, f_m^*) &= \arg \max_{F,\ dim(F)=m} \sum_{i=1}^{n} \left\|\pi_F(x'_i)\right\|^2 \\ &= \arg \max_{(f_1,\dots,f_m)\in O_{dn}} \sum_{i=1}^{n}\sum_{l=1}^{m} \langle x'_i, f_l\rangle^2\end{aligned}$$

is

$$(f_1^*, \dots, f_m^*) = (e_1, \dots, e_m).$$

By putting together the results of lemmas 3 and 4, we obtain the main result on k-flats.

Proposition 3
Let X_n be a data set ($X_n \in \mathbb{R}^{d\times n}$), $\tilde{X}_n$ the recentered data set, m an integer between 1 and d. Let Ω_m be the set of m-dimensional vector subspaces of $\mathcal{X}$, $\mathcal{L}$ the loss function:

$$\begin{aligned}\mathcal{L}: \quad \mathcal{X} \times \Omega_m &\to \mathbb{R} \\ (y, F) &\mapsto \sum_{i=1}^{n} \|x_i - \pi_{y+F}(x_i)\|^2 = \sum_{i=1}^{n} \|x_i - y - \pi_F(x_i - y)\|^2\end{aligned}$$

Let $(e_l)_{1\leq l\leq d}$ be the family of eigenvectors of $\tilde{X}_n \tilde{X}_n^\top$ ranked in the order of decreasing eigenvalues. Then the solution to

$$(y^*, F^*) = \arg\min_{y,F} \mathcal{L}(y, F),$$

is

$$(y^*, F^*) = \left(\frac{\sum_{i=1}^{n} x_i}{n}, span\ (e_1, \dots, e_m)\right).$$

Proposition 3 provides the mathematical justification to the process we mentioned in 1.2.2: indeed, by restricting the original problem to each of the k group points, and by noticing that $\mathcal{L}(y, F) = \hat{\mathcal{E}}(y + F)$, we can apply directly the result of the proposition.

2 Kernel extensions

In supervised learning, kernels are used to allow more flexibility than linear models [31]. In this section, we present kernels in order to define kernel versions of k-means and k-flats. Kernel k-means has been introduced to clusterize data that were not linearly separable in the input space [19]. Based on the same principle, we propose a kernel version of k-flats that we call kernel k-flats. In the whole section, we use the same notations as in section 1.1 and 1.2.

2.1 Reproducing Kernel Hilbert Spaces

We begin by recalling the basic definitions of kernels and Reproducing Kernel Hilbert Spaces (RKHS). For further reference, see [9].

Definition 2 (Feature map, feature space),[9]
Let S be a non-empty set. A function $k : S \times S \to \mathbb{K}$ is said to be a kernel on S if there exists a $\mathbb{K}$-Hilbert[11] *space H and a map $\phi : S \to H$ such that:*

$$\forall x, x' \in S, \quad k(x, x') = \left\langle \phi(x), \phi(x') \right\rangle_H ,$$

where $\langle \cdot, \cdot \rangle_H$ denotes the inner product of H.
ϕ is called a feature map and H a feature space of k.

The following proposition is useful to determine whether a function is a kernel or not.

Proposition 4 (Characterization of a kernel),[9]
A function $k : S \times S \to \mathbb{K}$ is a kernel if and only if it is symmetric and positive definite, i.e.*:*

$$\begin{aligned} &i. \quad \forall x, x' \in S,\ k(x, x') = k(x', x). \\ &ii. \quad \forall n \in \mathbb{N}, \forall \alpha_1, \ldots, \alpha_n \in \mathbb{R}, \forall y_1, \ldots, y_n \in S,\ \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(y_i, y_j) \geq 0. \end{aligned}$$

Examples of kernels

- Gaussian kernel:
$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2} \right).$$
- Polynomial kernel:
$$k(x, y) = \left(\langle x, y \rangle + 1 \right)^d .$$
- Hyperbolic tangent kernel:
$$k(x, y) = \tanh\left(\alpha \langle x, y \rangle + c \right) \quad \alpha > 0,\ c < 0.$$

Using definition 2, we can now define an important tool: the Reproducing Kernel Hilbert Space.

Definition 3 (Reproducing kernel, RKHS),[9]
Let $S \neq \emptyset$ and H a $\mathbb{K}$-Hilbert function space over X, i.e *a $\mathbb{K}$-Hilbert space that consists of function from X to $\mathbb{K}$.*
A function $k : S \times S \to \mathbb{K}$ is called a reproducing kernel of H if:

$$\begin{aligned} &i. \quad \forall x \in S,\ k(\cdot, x) \in H. \\ &ii. \quad \forall f \in H,\ \forall x \in S,\ \langle f, k(\cdot, x) \rangle_H = f(x). \end{aligned}$$

The space H is called a reproducing kernel Hilbert space over S if the Dirac functional $\delta_x : H \to \mathbb{K}$ (see the definition below) is continous.

$$\delta_x(f) = f(x), \quad f \in H.$$

[11] $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$.

Having these definitions in mind, we can define the kernel matrix. In fact it is simply the Gram matrix of $\Phi_n = (\phi(x_1), \ldots, \phi(x_n))$, the image of the data points by the feature map ϕ. The kernel matrix is very important because it is one of the few quantities of the feature space that can be computed in practice (see section 2.1.1).

Definition 4 (Kernel matrix)
Given a training set $X_n = (x_1, \ldots, x_n)$, the kernel matrix $K \in \mathbb{R}^{n \times n}$ is defined by:

$$K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle_H = k(x_i, x_j).$$

K can be expressed as the Gram matrix of[12] $\Phi_n = (\phi(x_1, \ldots, \phi(x_n)) \in \mathbb{R}^{d' \times n}$ *:*

$$K = \Phi_n^\top \Phi_n.$$

The kernel matrix inherits from the properties of the generating kernel (symmetry and positiveness).

Proposition 5 *K is symmetric positive definite.*

Proof
The symmetry comes from the symmetry of k.
Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$. We have:

$$\alpha^\top K \alpha = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i k(x_i, x_j) \alpha_j \underset{\text{Proposition 4}}{\geq} 0,$$

and the proposition follows. ∎

2.1.1 Practical consequences of the use of kernels

In practice, the feature map ϕ is not known, that is, we can only use the kernel k to compute distances and, more generally, to compute inner products in the feature space. The algorithms that we will discuss in the next sections rely heavily on this fact as they require distances to be computed at each step.
This fact is at the basis of some differences between non-kernel and kernel versions of k-means and k-flats. For instance the outputs of k-means and kernel k-means are different: k-means produces a set of point whereas kernel k-means cannot. Indeed, in theory, we can write:

$$m_j = \frac{\sum_{i \in \mathcal{C}_j} \phi(x_i)}{n_j},$$

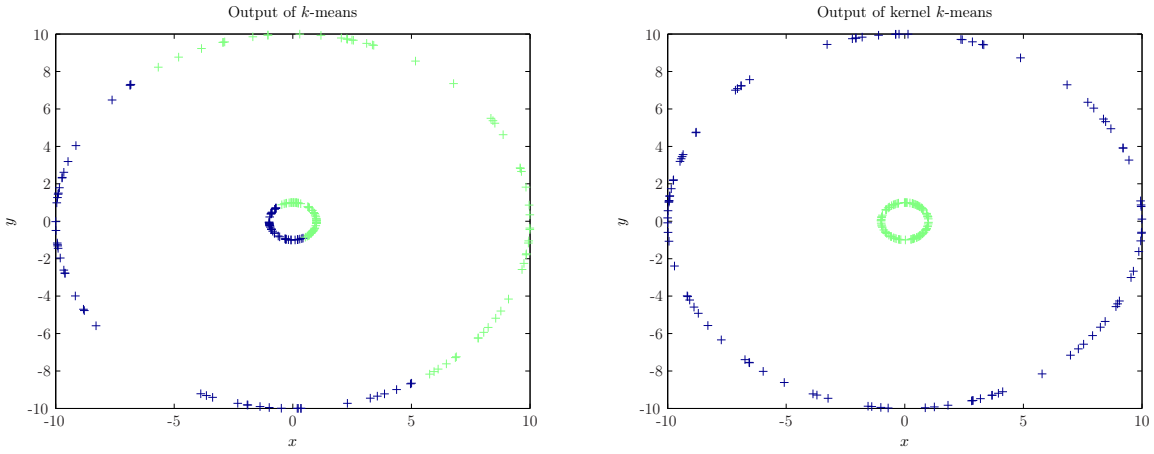
but in practice it is impossible to compute such a quantity as ϕ is not known. However, we show in section 2.2 that knowing the assignment vector $C = (C(i))_{1 \leq i \leq n}$ is sufficient to compute distances of the form $d_H(\phi(x_i), m_j)$.

We must also mention that the expression of the reconstruction error (1) is a bit modified to take into account the fact that we map points to a feature space H:

Definition 5 (Empirical reconstruction error in the kernel case)
For a closed set $S \subseteq H$, the empirical reconstruction error is:

$$\hat{\mathcal{E}}^{(f)}(S) = \frac{1}{n} \sum_{i=1}^{n} d_H(\phi(x_i), S)^2 = \frac{1}{n} \sum_{i=1}^{n} \left\| \phi(x_i) - \pi_S(\phi(x_i)) \right\|_H .$$

[12] d' can be infinite.



(a) k-means (b) Kernel k-means

Figure 4: Comparison of the outputs of k-means and kernel k-means for $k = 2$ on a dataset composed of two concentric circles of radiuses 1 and 10.

It differs from the original reconstruction error by the fact that distances are measured using the norm $\|\cdot\|_H$ of the feature space H. As a consequence, it will highly depend on the kernel chosen[13].

2.2 Kernel k-means

The necessity of creating a kernel version of k-means, kernel k-means [19, 32], has stemmed from the need to separate clusters that are nonlinearly separable. Indeed, as we have seen in section 1.1, k-means produces linear boundaries between the clusters. By mapping the data points to a feature space and running k-means in this space, kernel k-means produces nonlinear boundaries (see figure 4) in the input space (although the boundaries in the feature space are linear). Points in the feature space cannot be computed and thus the computation of algorithm 4 and Lloyd's algorithm are different: algorithm 4 outputs the assignment vector C such that

$$C(i) = j \Leftrightarrow \|\phi(x_i) - m_j\|_H \leq \|\phi(x_i) - m_l\|_H \quad \forall l \in [\![1, k]\!],$$

and requires a kernel $k(\cdot, \cdot)$ in addition to the input of k-means. We show in section 2.2.2 that the assignment vector is sufficient to compute distances from any data point to any mean using the kernel matrix.
Kernel k-means can be seen as minimizing error (5) over the following hypothesis space:

$$\mathcal{H} = \mathcal{P}_k^{(f)} \text{ where } \mathcal{P}_k^{(f)} \text{ denotes the sets of } k \text{ points from the feature space.}$$

2.2.1 Algorithm

Kernel k-means and k-means are identical in principle but there are some differences in their computations. In this section, we present a variant of Lloyd's algorithm adapted to kernels.

Algorithm 4
Input[14]*: a data set X_n, an integer k (number of means), and a kernel $k(\cdot, \cdot)$.*
Output*: assignment vector C.*

[13] And also of the parameters of the kernels, such as the value of σ for a gaussian kernel.
[14] An alternative would be to provide the kernel matrix K and an integer k (number of means).

1. *Initialize the assignment vector C with values between 1 and k.*

2. *Compute the distances from the data points to the means (see equation (11) for details), that is compute:*
$$d_H(\phi(x_i), m_j), \quad \forall i \in [\![1, n]\!], \ \forall j \in [\![1, k]\!].$$

3. *Assign data points to their closest mean:*
$$\forall i \in [\![1, n]\!], \ C(i) = \arg \min_{1 \leq j \leq k} \|\phi(x_i) - m_j\|_H .$$

4. *Repeat steps 2. and 3. until convergence.*

2.2.2 How are distances computed in the feature space ?

Equation (11) provides the mathematical details for step 2. of algorithm 4. In this equation $\|\cdot\|_H$ is the norm of the feature space H:

$$\begin{aligned} \|\phi(x_i) - m_j\|_H^2 &= \left\| \phi(x_i) - \frac{1}{n_j} \sum_{l \in \mathcal{C}_j} \phi(x_l) \right\|_H^2 \\ &= \|\phi(x_i)\|_H^2 + \frac{1}{n_j^2} \left\| \sum_{l \in \mathcal{C}_j} \phi(x_l) \right\|_H^2 - \frac{2}{n_j} \sum_{l \in \mathcal{C}_j} \langle \phi(x_i), \phi(x_l) \rangle_H \\ &= K_{ii} + \frac{1}{n_j^2} \sum_{l, l' \in \mathcal{C}_j} K_{ll'} - \frac{2}{n_j} \sum_{l \in \mathcal{C}_j} K_{il}. \end{aligned} \tag{11}$$

where K denotes the kernel matrix. As expected, distances between means and data points do not require the knowledge of the feature map ϕ to be computed: everything can be done using the kernel matrix K.

2.2.3 Computational analysis

The computational analysis for kernel k-means is exactly the same as for k-means except that everything takes place in the feature space H instead of the input space $\mathcal{X}$. Therefore, the properties of k-means, in particular its correctness, easily transpose to kernel k-means.

Proposition 6
Algorithm 4 converges in a finite number of steps.

2.3 Kernel k-flats

In the previous section, we have seen that the use of a kernel in kernel k-means allowed us to get nonlinear boundaries between the clusters. The same idea transposes to k-flats: we have seen (cf section 1.2) that k-flats produced a linear approximation of the samples by using k affine spaces. By using a kernel, a nonlinearity is introduced and has for consequence that the approximating surfaces in the input space are nonlinear. The hope is that by allowing the approximating surface to be nonlinear, the representation of data sets that are poorly approximated by linear surfaces will be improved.
Kernel k-flats takes as input a data set, two integers k and m respectively representing the

number of flats and the dimension of each flat and ouputs the assignment vector C (and possibly the coefficients of the directions of the flats in the basis $\Phi_n = (\phi(x_1), \dots, \phi(x_n))$). Given the values of k and m, kernel k-flats can be seen as minimizing the error of equation (5) on the following hypothesis space:

$\mathcal{H} = \mathcal{F}_{k,m}^{(f)}$ where $\mathcal{F}_{k,m}^{(f)}$ is the class of sets of k vector spaces of dimension m in the feature space.

2.3.1 Algorithm

In this section, we propose an algorithm that mixes the ideas of kernel k-means and k-flats. The principle is the same as Lloyd's algorithm, that is, an alternation between optimization over the assignments and optimization over the flats.

Algorithm 5
Input[15]*: a data set X_n, two integers k (number of means) and m (number of flats), a kernel $k(\cdot, \cdot)$.*
Output*: assignment vector C and optionally the coefficients $(a_i^{(j)})_{1 \leq i \leq n, 1 \leq j \leq m}$ (as defined in equation (12)).*

1. *Initialize the assignment vector C with values between 1 and k.*
2. *$\forall j \in [\![1, k]\!]$, compute the distances from the data points to the flats, that is compute:*

$$d_H(\phi(x_i), F_j)^2, \quad \forall i \in [\![1, n]\!], \ \forall j \in [\![1, k]\!].$$

3. *Assignment: assign each data point to the closest (in the feature space) flat:*

$$\forall i \in [\![1, n]\!], \ C(i) = \arg \min_{1 \leq j \leq k} d_H(\phi(x_i), F_j).$$

4. *Repeat steps 2. and 3. until convergence.*

2.3.2 How are flats computed ?

In this section, we detail step 2. of algorithm 5. We justify this result in proposition 9.

Given the assignment vector C of the data, we can compute each flat independently, that is: for $j \in [\![1, k]\!]$, we consider the kernel matrix restricted to the j-th group of points:

$$K_j = K_{|\mathcal{C}_j} = (K_{uv})_{u \in \mathcal{C}_j, v \in \mathcal{C}_j}.$$

As in k-flats, we basically run PCA in the feature space (this is called *kernel PCA* [33, 34]) on each group of points. To be able to do this, each one of the $(K_j)_{1 \leq j \leq k}$ needs to be recentered, which can be done by the following transformation (see proposition 7 for the proof of this result):

$$\tilde{K_j} = K_j - 1_{n_j} K_j - K_j 1_{n_j} + 1_{n_j} K_j 1_{n_j},$$
$$\text{with } (1_{n_j})_{uv} = \frac{1}{n_j} \quad \forall u, v \in [\![1, n_j]\!].$$

[15] An alternative would be to provide the kernel matrix K and two integers k (number of means) and m (dimension of flat).

The j-th flat F_j is then given by:

$$F_j = \frac{\sum_{i\in\mathcal{C}_j}\phi(x_i)}{n_j} + \text{span}(\tilde{\Phi}_n.a_1^{(j)},\dots,\tilde{\Phi}_n.a_m^{(j)}). \tag{12}$$

where the $(a_l^{(j)})_{1\leq l\leq n_j}$ are the eigenvectors of $\tilde{K}_j$ with the associated eigenvalues $\lambda_1^{(j)} \geq \cdots \geq \lambda_d^{(j)}$. Notice that the $(a_l^{(j)})_{1\leq l\leq n_j}$ form an orthogonal basis, and we choose to normalize them (with respect to the feature space norm $\|\cdot\|_H$), that is:

$$a_l^{(j)}\tilde{K}_j a_l^{(j)} = 1 \quad \forall l \in [\![1, n_j]\!].$$

Rigorously, $a_l^{(j)} \in \mathbb{R}^{n_j}$, but for the sake of simplicity, $a_l^{(j)}$ will either denote a vector of $\mathbb{R}^{n_j}$ as suggested by the definition above or a vector of $\mathbb{R}^n$ defined as follows (for a vector v , $(v)_p$ denotes its p-th component):

$$(a_l^{(j)})_p = \begin{cases} 0 & \text{if } p \notin \mathcal{C}_j, \\ (a_l^{(j)})_p \text{ (the } a_l^{(j)} \text{ previously obtained)} & \text{otherwise.} \end{cases}$$

The main idea for the computation of the flats is that for a given group of points the flat goes through the center of mass of the image of data points of the group, and is directed by the m principal components.

2.3.3 How are distances to the flats computed ?

In this section, we detail step 3. of algorithm 5.

Given the assignment vector C of the data and for each $j \in [\![1, k]\!]$ the eigenvectors $(a_l^{(j)})_{1\leq l\leq n_j}$ of $\tilde{K}_j$, we can compute distances to the flats using only the kernel matrix (see appendix B.3 for the proof of this result).

$$\begin{aligned} d_H(\phi(x_i), F_j)^2 \;=\; & -\sum_{l=1}^{m}\sum_{p,p'\in\mathcal{C}_j}(a_l^{(j)})_{p'}(a_l^{(j)})_p\left(K_{ip'} - \frac{1}{n_j}\sum_{q\in\mathcal{C}_j}\left(K_{qp'}+K_{iq}\right) + \frac{1}{n_j^2}\sum_{q,q'\in\mathcal{C}_j}K_{qq'}\right) \\ & \left(K_{ip} - \frac{1}{n_j}\sum_{q\in\mathcal{C}_j}(K_{qp}+K_{iq}) + \frac{1}{n_j^2}\sum_{q,q'\in\mathcal{C}_j}K_{qq'}\right) + K_{ii} + \frac{1}{n_j^2}\sum_{q,q'\in\mathcal{C}_j}K_{qq'} - \frac{2}{n_j}\sum_{q\in\mathcal{C}_j}K_{iq}. \end{aligned}$$

2.3.4 Computational analysis

At each iteration, the computation of the k flats in algorithm 5 can be done by splitting the data set into k groups (according to the assignment vector) and compute one flat on each group of points (the computation of the flat of a given group does not require data points that do not belong to the group). For the sake of simplicity in the notation, the proofs of this section (most of them are relegated in appendix B.4) take advantage of this remark and are written as if there was only one group of points.
We begin by justifying the recentering step.

Proposition 7 (Recentering in the feature space),[33]
If $\tilde{K}$ is defined by:

$$\tilde{K}_{ij} = \left\langle \phi(x_i) - \frac{\sum_{l=1}^n\phi(x_l)}{n}, \phi(x_j) - \frac{\sum_{l=1}^n\phi(x_l)}{n}\right\rangle_H \quad \forall i,j \in [\![1,n]\!].$$

Then $\tilde{K}$ can be computed from K using the following transformation:

$$\tilde{K} = K - 1_n K - K 1_n - 1_n K 1_n,$$
$$\text{where } (1_n)_{ij} = \frac{1}{n} \quad \forall i, j \in [\![1, n]\!].$$

Proof

$$\begin{aligned}
\tilde{K}_{ij} &= \left\langle \phi(x_i) - \frac{\sum_{l=1}^n \phi(x_l)}{n}, \phi(x_j) - \frac{\sum_{l=1}^n \phi(x_l)}{n} \right\rangle_H \\
&= \underbrace{\langle \phi(x_i), \phi(x_j) \rangle_H}_{K_{ij}} - \underbrace{\left\langle \frac{\sum_{l=1}^n \phi(x_l)}{n}, \phi(x_i) \right\rangle_H}_{(K1_n)_{ij}} - \underbrace{\left\langle \frac{\sum_{l=1}^n \phi(x_l)}{n}, \phi(x_j) \right\rangle_H}_{(1_n K)_{ij}} \\
&\quad + \underbrace{\left\langle \frac{\sum_{l=1}^n \phi(x_l)}{n}, \frac{\sum_{l'=1}^n \phi(x_{l'})}{n} \right\rangle_H}_{(1_n K 1_n)_{ij}}. \quad \blacksquare
\end{aligned}$$

As algorithm 5 follows the same principle as Lloyd's algorithm, it is easy to prove the following result.

Proposition 8
Algorithm 5 converges in a finite number of steps.

We now show the main result of this section, which justifies the process of the computation of the flats.

Proposition 9 (Computation of the flats)
Let $X_n = (x_1, \ldots, x_n)$ be a data set, $\Phi_n = (\phi(x_1), \ldots, \phi(x_n))$ the image of the data set ($\Phi_n \in \mathbb{R}^{d' \times n}$), $\tilde{\Phi}_n$ the recentered image of the data set $\tilde{\Phi}_n = (\phi(x_1) - \overline{\Phi}_n, \ldots, \phi(x_n) - \overline{\Phi}_n)$, with $\overline{\Phi}_n = \frac{1}{n}\sum_{i=1}^n \phi(x_i)$, m an integer between 1 and $\min(d', n)$. Let also Ω_m be the set of m-dimensional vector subspaces of H and $\mathcal{L}$ the loss function:

$$\begin{aligned}
\mathcal{L}: \quad H \times \Omega_m &\to \mathbb{R} \\
(y, F) &\mapsto \sum_{i=1}^n \|\phi(x_i) - \pi_{y+F}(\phi(x_i))\|_H^2 = \sum_{i=1}^n \|\phi(x_i) - y - \pi_F(\phi(x_i) - y)\|_H^2
\end{aligned}$$

Let $(a_l)_{1 \le l \le n}$ be the family of eigenvectors of $\tilde{\Phi}_n^\top \tilde{\Phi}_n$ ranked in the order of decreasing eigenvalues. Then the solution to

$$(y^*, F^*) = \arg\min_{y,F} \mathcal{L}(y, F),$$

is

$$(y^*, F^*) = \left(\frac{\sum_{i=1}^n \phi(x_i)}{n}, \text{span } (\tilde{\Phi}_n.a_1, \ldots, \tilde{\Phi}_n.a_m) \right).$$

We show in proposition 10 that we can use the image of the data Φ_n (instead of the recentered image $\tilde{\Phi}_n$) to compute the flat that is:

$$\overline{\Phi}_n + \text{span } (\tilde{\Phi}_n.a_1, \ldots, \tilde{\Phi}_n.a_m) = \overline{\Phi}_n + \text{span } (\Phi_n.a_1, \ldots, \Phi_n.a_m).$$

Proposition 10

$$\forall l \in [\![1, n]\!], \lambda_l \neq 0 \Rightarrow \Phi_n a_l = \tilde{\Phi}_n a_l.$$

2.3.5 Relation between kernel k-flats and k-flats

In this section, we prove that for the choice

$$\begin{array}{rccc} \phi: & \mathcal{X} & \to & \mathcal{X} \\ & x & \mapsto & x, \end{array}$$

(that is $k(x, x') = \langle x, x' \rangle$ and $H = \mathcal{X}$), algorithm 3 and 5 are equivalent. Indeed, in this case the kernel matrix is the Gram matrix of X_n: $K = G = X_n^\top X_n$
From the SVD theorem (appendix A), we know that G and $Z = X_n X_n^\top$ have the same nonzero eigenvalues and that their eigenvectors are linked. More precisely, if we call $A = (a_1, \dots, a_m) \in \mathbb{R}^{n \times m}$ the set of m eigenvectors of G associated with the largest eigenvalues $\lambda_1 \geq \cdots \geq \lambda_m$ and $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_m) \in \mathbb{R}^{m \times m}$, the columns $(u_l)_{1 \leq l \leq m}$ of $U = XA\Lambda^{-1} \in \mathbb{R}^{d \times m}$ are eigenvectors of Z with eigenvalues $\lambda_1, \dots, \lambda_m$, which are the largest eigenvalues of Z and this is precisely what we are looking for when running algorithm 3.
We used this link to check that the practical implementations of algorithms 3 and 5 were coherent.

This remark can also be useful from a computational point of view: indeed, it can be more efficient to compute flats using one or the other formulation.
Algorithm 3 computes at each step k d-by-d matrices whereas algorithm 5 computes k matrices with dimensions n_l-by-n_l for $l \in [\![1, k]\!]$. Thus:

$$\begin{aligned} \text{cost}_{\text{Algorithm 3}} &= kd^2 \\ \text{cost}_{\text{Algorithm 5}} &= \sum_{i=1}^{k} n_l^2 \leq kn^2 \end{aligned}$$

where cost denotes the cost in terms of memory at a given iteration of the algorithm.
For instance, if $n \leq d$, it is more efficient, in terms of memory, to use algorithm 5 with the inner product of $\mathcal{X}$ as a kernel.

3 Framework and relation to existing problems

The computational aspects of k-means and k-flats being recalled, we now present the framework as in [18, 17] for the statistical study of the properties of these algorithms. We will in particular focus on three points:

- relation of k-means to existing problems (sections 3.2.1 and 3.2.2).
- the asymptotic behavior of the reconstruction error with respect to the number of means (section 4.1).
- propose an adaptive choice of k based on complexity regularization (section 4.2).

3.1 Formal setting

Assume given a separable Hilbert space $\mathcal{X}$, with inner product denoted by $\langle \cdot, \cdot \rangle$, endowed with a Borel probability measure p that is supported over a compact, smooth d-manifold $\mathcal{M}$ with metric of class $\mathcal{C}^1$. Indeed, it is a common assumption in statistical learning to suppose that even if the data is embedded in a high-dimensional space, there exists a low-dimensional structure on which the data lies [35].

The probability measure p is assumed to be absolutely continuous with respect to the volume measure on $\mathcal{M}$, with density ρ. Further, assume that the samples $X_n = (x_1, \ldots, x_n)$ are drawn identically and independently with respect to p (the training set.) We give a few examples to motivate this setting.

Examples

- Euclidean spaces: the space $\mathcal{X}$ is a finite dimensional Euclidean space $\mathcal{X} = \mathcal{M} = \mathbb{R}^d$. This is the setting we used in section 1.
- Signals can be seen as functions in $\mathcal{X} = L^2(\mathbb{R})$ with dot product $\langle f, g \rangle = \int_{\mathbb{R}} f(x)g(x)dx$. In practice, signals are discretized and their study can then be done in an Euclidean space; but it is important to note that the framework we use is general enough to allow the theoretical study of time-continuous signals.
- Feature space: when the data set is mapped to a high-dimensional (possibly infinite dimensional) Hilbert space H called the feature space (see section 2.1), $\mathcal{X}$ is replaced by H and $\langle \cdot, \cdot \rangle$ is replaced by the kernel function $k(\cdot, \cdot)$.

We have defined in section 1 an empirical measure of risk based on the training set. We now give a continuous version of this measure called the *reconstruction error*:

Definition 6 (Reconstruction error)
Given a closed set S, the reconstruction error of S for a source distributed according to the probability measure p is defined by:

$$\mathcal{E}(S) = \int_{\mathcal{X}} d^2_{\mathcal{X}}(x, S)dp(x), \tag{13}$$

where the distance between a point x and a set S is:

$$d_{\mathcal{X}}(x, S) = \min_{s \in S} d_{\mathcal{X}}(x, s).$$

The definition of the reconstruction error is consistent with the definition of the empirical reconstruction error: indeed, in the case of a training set drawn identically independently with respect to p, the empirical reconstruction error is an unbiased estimator of the reconstruction error:

$$\mathbb{E}\left(\hat{\mathcal{E}}(S)\right) = \mathcal{E}(S),$$

and

$$\mathbb{P}\left(\lim_{n \to \infty} \hat{\mathcal{E}}(S) = \mathcal{E}(S)\right) = 1.$$

Notice that the risk of equation (13) is minimized (with null risk) by setting S to be the support $\text{supp}(p) = \mathcal{M}$ of p, which is the set that perfectly estimates $\mathcal{M}$ itself, and in this sense, equation (13) is consistent with our problem definition. However, equation (13) is also minimized by any set S' that contains $\mathcal{M}$ and, in particular, is zero when simply setting $S = \mathcal{X}$, independently of p.

In the remainder of this report, we are going to consider the problem of minimizing the reconstruction error on a *hypothesis space* $\mathcal{H}$. In this view, an unsupervised learning algorithm is a function $\mathcal{A}$ that computes a set in the hypothesis space using the training set:

$$\begin{array}{rccl} \mathcal{A}: & \chi^n & \to \mathcal{H} \subset \mathcal{P}(\mathcal{X}) \\ & X_n & \mapsto \hat{S} \end{array}$$

where $\mathcal{P}(\mathcal{X})$ is the set of subsets of $\mathcal{X}$.
For a given algorithm (and its corresponding hypothesis space), the objective can be defined as follows:

$$\min_{S \in \mathcal{H}} \mathcal{E}(S).$$

As the distribution p is not known a priori, $\mathcal{E}$ cannot be computed. Instead the algorithms will attempt to minimize the empirical error on the hypothesis space[16], that is find the set $\hat{S} \in \mathcal{H}$ as defined in equation (2). However, we must keep in mind that the quantities in which we are interested measure the perfomance of the set $\hat{S}$ on the true distribution and they are:

$$\mathcal{E}(\hat{S}) \text{ and } \mathbb{P}\left(\mathcal{E}(\hat{S}) > \epsilon\right) \tag{14}$$

Indeed, as we will see in section 5, the empirical reconstruction error and the reconstruction error may behave differently for some parameters. For instance, the empirical reconstruction error is always a decreasing function of the number k of means (in k-means) whereas, depending on the distributon of the data, the reconstruction error may decrease until a value k strictly between 1 and n and then increase or, in some cases, even be monotically increasing. Even if the causes of this phenomenon are not well understood, this phenomenon can be compared to *overfitting* in a supervised framework and we will see that it is possible to adapt methods from the supervised learning framework such as *complexity regularization* (see section 4.2) to propose a choice of k and prove it is relevant (in the sense that it is optimal up to a certain error).

3.2 Relation to other known problems

In this section, we show the relation between the problem of studying $\mathbb{P}\left(\mathcal{E}(\hat{S})\right)$ (equation (14)) and existing, well-studied problems, namely the problems of optimal quantization of probability distributions [36, 37], and clustering [31]. In particular, the minimization of the *expected* risk of equation (13), over the class of finite sets $\mathcal{H}$ of a prescribed size k, is exactly the problem of optimal quantization of probability distributions, while the minimization of the *empirical* risk of equation (1) can be seen to be related the problem of clustering. Although throughout this section we will assume that the minimization is over finite sets, as in k-means, the tools introduced will turn out to be useful even in more general cases (e.g. when the hypothesis space consists of collections of affine spaces, as in section 1.2).

3.2.1 Clustering

Given a set of samples X_n, the problem of k-clustering, is that or grouping the n samples into k subsets $(x_i)_{i \in \mathcal{C}_j}$ where the $(\mathcal{C}_j)_{1 \leq j \leq k}$ (see equation (5) for the definition of $(\mathcal{C}_j)_{1 \leq j \leq k}$) form a partition of $[\![1, n]\!]$, such that the sum of intra-cluster distances

$$\frac{1}{2} \sum_{j=1}^{k} \frac{1}{n_j} \left(\sum_{i \in \mathcal{C}_j} \sum_{l \in \mathcal{C}_j} \|x_i - x_l\|^2 \right),$$

is minimized.

[16] This process is justified by the fact that if the size of the training set goes to infinity, the reconstruction error and its empirical version coincide.

By letting $\bar{x}_j = \frac{1}{n}\sum_{i\in\mathcal{C}_j} x_i$ be the center of mass of cluster $\mathcal{C}_j$, it is easy to see that the above sum can be written as

$$\sum_{j=1}^{k}\sum_{i\in\mathcal{C}_j}\|x_i-\bar{x}_j\|^2.$$

Indeed,

$$\begin{aligned}\sum_{j=1}^{k}\left(\frac{1}{2n_j}\sum_{i\in\mathcal{C}_j}\sum_{l\in\mathcal{C}_j}\|x_i-x_l\|^2\right) &= \sum_{j=1}^{k}\left(\frac{1}{2n_j}\sum_{i\in\mathcal{C}_j}\sum_{l\in\mathcal{C}_j}\left(\|x_i\|^2+\|x_l\|^2-2\langle x_i,x_l\rangle\right)\right)\\ &= \sum_{j=1}^{k}\left(\sum_{i\in\mathcal{C}_j}\|x_i\|^2-\frac{1}{n_j}\sum_{i\in\mathcal{C}_j}\sum_{l\in\mathcal{C}_j}\langle x_i,x_l\rangle\right)\end{aligned}$$

on the other hand,

$$\begin{aligned}\sum_{i\in\mathcal{C}_j}\|x_i-\bar{x}_j\|^2 &= \sum_{i\in\mathcal{C}_j}\left(\|x_i\|^2-2\frac{1}{n_j}\left\langle x_i,\sum_{l\in\mathcal{C}_j}x_l\right\rangle+\frac{1}{n_j^2}\left\langle\sum_{l\in\mathcal{C}_j}x_l,\sum_{l'\in\mathcal{C}_j}x_{l'}\right\rangle\right)\\ &= \sum_{i\in\mathcal{C}_j}\|x_i\|^2-\frac{1}{n_j}\left\langle\sum_{l\in\mathcal{C}_j}x_l,\sum_{l\in\mathcal{C}_j}x_l\right\rangle\end{aligned}$$

and therefore,

$$\sum_{j=1}^{k}\left(\frac{1}{2n_j}\sum_{i\in\mathcal{C}_j}\sum_{l\in\mathcal{C}_j}\|x_i-x_l\|^2\right)=\sum_{j=1}^{k}\sum_{i\in\mathcal{C}_j}\|x_i-\bar{x}_j\|^2. \tag{15}$$

In order to minimize the above cost, we may first relax the minimization to be over the larger space of all possible sets $\{m_1,\dots,m_k\}$ of size k (instead of minimizing only over centers of mass of clusters). Minimizing

$$\sum_{i=1}^{n}\min_{1\le j\le k}\|x_i-m_j\|^2,$$

over sets $\{m_1,\dots,m_k\}$, is the k-means problem, for which there are practical algorithms, such as Lloyd's algorithm (see section 1.1).

To prove the equivalence between the k-means problem and the clustering problem, we need to prove the following points (some proofs are relegated in appendix B.1):

- the center of mass is the point that strictly minimizes the sum of intra-cluster distances (which has been proved in lemma 1).
- a training point is never halfway between two means (proposition 12).
- if we denote by $\mathcal{B}_j$ the set of points that are strictly closer to the j-th mean (equation (16)), the $(\mathcal{B}_j)_{1\le j\le k}$ form a partition of $[\![1,n]\!]$ (Corollary 1).

where $\mathcal{B}_j$ is defined by:

$$\mathcal{B}_j=\{i\in[\![1,n]\!],\|x_i-m_j\|<\|x_i-m_l\|\ \ \forall l\in[\![1,k]\!]\setminus\{j\}\} \tag{16}$$

We also define $\mathcal{R}$ to be the set of indices of points that do not belong to one of the $\mathcal{B}_j$. Therefore

$$\left(\bigcup_{j=1}^{k}\mathcal{B}_j\right)\bigcup\mathcal{R}=[\![1,n]\!]. \tag{17}$$

We begin by proving that in an optimal configuration, two means can never be equal.

Proposition 11 *If $k \leq n$, and if $\{m_1, \ldots, m_k\}$ denotes the optimal set of means:*

$$j \neq j' \Rightarrow m_j \neq m_{j'}.$$

We can now prove that no point lies halfway between two means using the same idea as in [38].

Proposition 12 *If $\{m_1, \ldots, m_k\}$ denotes the optimal set of means, then no point can be exactly between two means* i.e.*:*

$$\{i \in [\![1, n]\!], \exists u, v \in [\![1, k]\!], u \neq v \text{ and } \|x_i - m_u\| = \|x_i - m_v\|\} = \emptyset,$$

and therefore $\mathcal{R} = \emptyset$.

The following corollary is the landmark of the equivalence between the clustering problem and k-means.

Corollary 1 *The the $(\mathcal{B}_j)_{1 \leq j \leq k}$ form a partition of $[\![1, n]\!]$* i.e.*:*

$$i. \quad \bigcup_{j=1}^{k} \mathcal{B}_j = [\![1, n]\!].$$
$$ii. \quad \forall j, j' \in [\![1, k]\!], j \neq j' \Rightarrow \mathcal{B}_{j'} \cap \mathcal{B}_j = \emptyset.$$

Proof
Proposition 12 allows us to say that:

$$\mathcal{R} = \emptyset,$$

and thus by equation (17) we get i.. ii. trivially holds by the definition of $\mathcal{B}_j$. ∎

With all these technical points in mind, we can prove the expected equivalence.

$$\begin{aligned}
\min_{\{m_1,\ldots,m_k\} \in \chi^k} \left(\sum_{i=1}^{n} \min_{1 \leq j \leq k} \|x_i - m_j\|^2 \right) &\underset{\text{Proposition 12}}{=} \min_{\{m_1,\ldots,m_k\} \in \chi^k} \left(\sum_{j=1}^{k} \sum_{i \in \mathcal{B}_j} \|x_i - m_j\|^2 \right) \\
&\underset{\text{Lemma 1 and corollary 1}}{=} \min_{(\mathcal{C}_1,\ldots,\mathcal{C}_k) \in \mathrm{Pa}([\![1,n]\!])} \left(\sum_{j=1}^{k} \sum_{i \in \mathcal{C}_j} \|x_i - \bar{x}_j\|^2 \right) \\
\min_{\{m_1,\ldots,m_k\} \in \chi^k} \left(\sum_{i=1}^{n} \min_{1 \leq j \leq k} \|x_i - m_j\|^2 \right) &\underset{\text{Equation (15)}}{=} \min_{(\mathcal{C}_1,\ldots,\mathcal{C}_k) \in \mathrm{Pa}([\![1,n]\!])} \sum_{j=1}^{k} \left(\frac{1}{2n_j} \sum_{i \in \mathcal{C}_j} \sum_{l \in \mathcal{C}_j} \|x_i - x_l\|^2 \right) \quad (18)
\end{aligned}$$

where $\mathrm{Pa}([\![1, n]\!])$ denotes the set of partitions of $[\![1, n]\!]$. Equation (18) proves that the problem of clustering and k-means are identical.

3.2.2 Optimal quantization

The problem of optimal quantization consists in finding the best appproximation (in the sense that it minimizes some measure of distortion) of a d-dimensional distribution by a set of k points (where k is a given integer) and is therefore very similar to k-means. Let us begin by recalling the definitions of vector quantizers and Voronoi regions.

Definition 7 (Vector quantizer, Voronoi regions)
A k-point nearest neighbor quantizer $Q : \mathcal{X} \to \mathcal{X}$ is a Borel function that maps vectors from the ambient space $\mathcal{X}$ into a finite set of k vectors $\{m_1, \dots, m_k\}$. Given $\{m_1, \dots, m_k\}$, a Borel partition $R_1, \dots, R_k$ of the space satisfying

$$R_j \subset \left\{ x \in \mathcal{X}, \|x - m_j\| = \min_{1 \leq j' \leq k} \|x - m_{j'}\| \right\},$$

defines a quantizer by:

$$Q(x) = \sum_{j=1}^{k} m_j \mathbb{1}_{R_j}(x).$$

$m_1, \dots, m_k$ are called code points and the collection of all code points is the codebook. In this case, the Voronoi region V_j associated to m_j is defined by:

$$V_j = \left\{ x \in \mathcal{X},\ d_{\mathcal{X}}(x, m_j) \leq d_{\mathcal{X}}(x, m_{j'})\ \forall j' \in [\![1, k]\!] \right\}.$$

The output of k-means is a precisely a nearest neighbor k-point quantizer. Such a quantizer splits the space into k Voronoi regions, each of them representing the set of points that are mapped to a given mean. In this sense a quantizer naturally induces clusters: more precisely, let $m_1, \dots, m_k$ be code points associated with a quantizer Q and define $(R_j)_{1 \leq j \leq k}$, such that:

$$R_j = Q^{-1}(m_j) = \{ x \in \mathcal{X},\ Q(x) = m_j \}.$$

One trivially sees that $\bigcup_{1 \leq j \leq k} R_j = \mathcal{X}$.

To measure how well the quantizer approximates the data, we need to define a measure of risk that, in the context of optimal quantization, is called *distortion*.

Definition 8 (Distortion of a quantizer)
Let $\mathcal{L}$ be a loss function and p a probability measure on $\mathcal{X}$. The distortion of a quantizer Q with respect to the probability measure p is:

$$D(Q) = \int_{\mathcal{X}} \mathcal{L}(x, Q(x)) dp(x).$$

A common case is ($\mathcal{X}$ needs to be a normed space) $\mathcal{L}(x, Q(x)) = \|x - Q(x)\|^2$. In this case, if S denotes the set of code points of Q, we have $D(Q) = \mathcal{E}(S)$.

As we already pointed out in section 3.1, we are given only a training set X_n (we do not know the distribution p) and thus we need to define an empirical measure of risk, the *empirical distortion*:

Definition 9 (Empirical distortion of a quantizer)
Under the assumptions of definition 8, the empirical distortion of a quantizer Q is:

$$\hat{D}(Q) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(x_i, Q(x_i)).$$

If $\mathcal{L}(x, Q(x)) = \|x - Q(x)\|^2$ and S denotes the set of code points of Q, we have $\hat{D}(Q) = \hat{\mathcal{E}}(S)$.

Given these definitions of a k-point quantizer, the problem of optimal quantization for a given k can be written:

$$\text{Find } Q_k \text{ such that } Q_k = \arg\min_{Q\in\mathcal{Q}_k} D(Q) = \arg\min_{Q \text{ Borel function, } \mathrm{card}(\mathrm{Im}(Q))\leq k} D(Q),$$

where $\mathcal{Q}_k$ is the set of k-point quantizers.

We now give essential results on optimal quantization as they will be of great use to prove bounds for k-means. In the remainder of this section, unless otherwise specified, we use the squared Euclidean distance as the loss function $\mathcal{L}$ because most of the results we will use from [36, 37, 21] hold only in this case. As our goal is to use the results for k-means and k-flats, we use the notation introduced in the framework, namely $\mathcal{E}(S)$ (respectively $\hat{\mathcal{E}}(S)$) for the distortion (respectively the empirical distortion).

Proposition 13 [36]
For any probability distribution and any $k \in \mathbb{N}$, there exists at least an optimal quantizer.

Now that we are ensured of the existence of the optimal quantizer, we give some results on the k-th quantization error of order r for a distribution p, $V_{k,r}(p)$, defined as follows [36][17]:

$$V_{k,r}(p) = \inf_{Q\in\mathcal{Q}_k} \mathbb{E}_{X\sim p}\left(\|X - Q(X)\|^r\right). \tag{19}$$

Proposition 14 [36]
Let r be an integer, and X be a random variable in $\mathbb{R}^d$ drawn according to p where p is a probability measure with absolutely continuous part ρ_a such that $\exists\delta > 0,\ \mathbb{E}\left(\|X\|^{r+\delta}\right) < \infty$. Then

$$\lim_{k\to\infty} k^{\frac{r}{d}} V_{k,r}(p) = C_{r,d} \|\rho_a\|_{\frac{d}{d+r}},$$

where $C_{r,d}$ is a constant depending on d and r only and $\|.\|_p$ denotes the L^p norm.

In particular we will often use this result for $r = 2$, that is:

$$\lim_{k\to\infty} k^{\frac{2}{d}} \mathcal{E}_k = C_d \|\rho_a\|_{\frac{d}{d+2}}, \tag{20}$$

where $\mathcal{E}_k$ is the best reconstruction error obtained for a set of points of size k *i.e.*:

$$\mathcal{E}_k = \inf_{S\in\mathcal{P}_k} \mathcal{E}(S).$$

However, the above result holds in the case of a random variable in $\mathbb{R}^d$ for a certain integer d whereas in our framework we suppose that the data lies on a d-dimensional manifold. Gruber [37] provides a generalization of the result to manifolds that will allow us to upper bound the approximation error of k-means under the manifold assumption.

Proposition 15 [37]
Let $\mathcal{P}_k^{(\mathcal{M})}$ be the set of sets of k points constrained to be on the manifold $\mathcal{M}$, d_I the geodesic distance on the manifold $\mathcal{M}$, and $\mathcal{E}_{\mathcal{M},k}$ the minimum reconstruction error over $\mathcal{P}_k^{(\mathcal{M})}$, that is

$$\mathcal{E}_{\mathcal{M},k} = \inf_{S\in\mathcal{P}_k^{(\mathcal{M})}} \int_{\mathcal{M}} d_I(x,S)\rho_a(x)d\mu_I(x).$$

Then the following asymptotic result holds:

$$\lim_{k\to\infty} k^{\frac{2}{d}}.\mathcal{E}_{\mathcal{M},k} = C_d \left(\int_{\mathcal{M}} \rho_a(x)^{\frac{d}{d+2}} d\mu_I(x)\right)^{\frac{d+2}{d}},$$

[17] The quantization problem of order r uses the loss function $\mathcal{L}(x) = x^r$. We choose to give the result of proposition 14 for a generic r because we will use it not only for $r = 2$ but also for $r = 4$ in the analysis of k-flats.

where μ_I is the measure of volume on the manifold $\mathcal{M}$.
This result will be of particular use to study the behavior of the approximation error of k-means (section 4.1).

3.3 Piece-wise estimation

In this section, we use definitions and results of the previous sections to explain in what sense k-means and k-flats can be seen as realizing piece-wise approximations of the input space $\mathcal{X}$. In particular, we detail the way points from $\mathcal{X}$ are encoded to generate the approximation of the space. The proofs of this section are relegated in appendix B.5.

3.3.1 k-means as a constant piece-wise approximation

The output of k-means is a set of k means. But according to the strong links between code points and clusters (see section 3.2.1), k-means can be seen as splitting the space into k regions (the Voronoi regions, see section 3.2.2). Points from a given region are encoded by the mean belonging to this region, and thus, k-means can be seen as realizing a constant piece-wise approximation of the space. More precisely, let us denote by $\mathcal{S}_j$ the part of the space that is strictly closer to m_j than any other mean, that is:

$$\mathcal{S}_j = \{x \in \mathcal{X},\ \|x - m_j\| < \|x - m_l\|\ \ \forall l \in [\![1, k]\!] \setminus \{j\}\},$$

and the corresponding cells augmented to include the boundaries of Voronoi regions:

$$\bar{\mathcal{S}}_j = \{x \in \mathcal{X},\ \|x - m_j\| \leq \|x - m_l\|\ \ \forall l \in [\![1, k]\!] \setminus \{j\}\}.$$

Let $(T_j)_{1 \leq j \leq k}$ be a partition of $\mathcal{X}$ such that $\mathcal{S}_j \subseteq T_j \subseteq \bar{\mathcal{S}}_j$.
Then a point x of the input space $\mathcal{X}$ is encoded by:

$$\sum_{j=1}^{k} m_j \mathbb{1}_{T_j}(x),$$

which is piece-wise constant. With this definition, each point of a region $\mathcal{S}_j$ is encoded by its closest mean m_j, whereas the points that lie at the intersection of (at least) two Voronoi regions (points that belong to a set $\bar{\mathcal{S}}_j \cap \bar{\mathcal{S}}_{j'}$ for $j \neq j'$) are encoded by one of the closest means. However, this indetermination is not very important as the set of points I that lie at the intersection of (at least) two Voronoi regions has zero measure (with respect to the Lebesgue measure of $\mathcal{X}$). Indeed,

$$I \subseteq \underbrace{\bigcup_{1 \leq j \leq k} \bigcup_{1 \leq j' < j} \bar{\mathcal{S}}_j \cap \bar{\mathcal{S}}_{j'}}_{J},$$

and J is composed of a countable union of zero-measure sets (see proposition 16). Thus it has zero measure and so has I.

Proposition 16 *The intersection of two extended Voronoi regions has zero measure, that is:*

$$\mu_{\mathcal{X}}(\bar{\mathcal{S}}_j \cap \bar{\mathcal{S}}_{j'}) = 0 \quad \forall j \neq j' \in [\![1, k]\!],$$

where $\mu_{\mathcal{X}}$ is the Lebesgue measure on $\mathcal{X}$.

3.3.2 k-flats as a linear piece-wise approximation

We have seen that k-means could be seen as a constant piece-wise approximation of the space (section 3.3.1). Similarly, k-flats can be seen as realizing a piece-wise linear approximation of the space[18]: instead of mapping entire regions to a point, each region is projected on an affine space, leading to a linear approximation of $\mathcal{X}$. More precisely, let us denote by $\mathcal{S}_j^{(f)}$ the part of the space that is stricly closer to F_j than any other flat, that is:

$$\mathcal{S}_j^{(f)} = \{x \in \mathcal{X},\ d_{\mathcal{X}}(x, F_j) < d_{\mathcal{X}}(x, F_l)\ \forall l \in [\![1, k]\!] \setminus \{j\}\},$$

and the corresponding augmented sets:

$$\bar{\mathcal{S}}_j{}^{(f)} = \{x \in \mathcal{X},\ d_{\mathcal{X}}(x, F_j) \leq d_{\mathcal{X}}(x, F_l)\ \forall l \in [\![1, k]\!] \setminus \{j\}\}.$$

and let $(T_j^{(f)})_{1 \leq j \leq k}$ be a partition of $\mathcal{X}$ such that $\mathcal{S}_j^{(f)} \subseteq T_j^{(f)} \subseteq \bar{\mathcal{S}}_j^{(f)}$.
Then a point x of the input space $\mathcal{X}$ is encoded by:

$$\sum_{j=1}^{k} \pi_{F_j}(x) \mathbb{1}_{T_j^{(f)}}(x),$$

which is piece-wise linear.
With this definition, each point of a region $\mathcal{S}_j^{(f)}$ is encoded by its projection on the closest flat F_j, whereas the points that lie at the intersection of (at least) two extended Voronoi regions (points that belong to a set $\bar{\mathcal{S}}_j^{(f)} \cap \bar{\mathcal{S}}_{j'}^{(f)}$ for $j \neq j'$) are projected on one of the closest flats. However, this indetermination is not very important as the set of points $I^{(f)}$ that lie at the intersection of (at least) two Voronoi regions has zero measure (with respect to the Lebesgue measure of $\mathcal{X}$). Indeed,

$$I^{(f)} \subseteq \underbrace{\bigcup_{1 \leq j \leq k} \bigcup_{1 \leq j' < j} \bar{\mathcal{S}}_j^{(f)} \cap \bar{\mathcal{S}}_{j'}^{(f)}}_{J^{(f)}},$$

and as $J^{(f)}$ has zero measure (see proposition 17), so has $I^{(f)}$.

Proposition 17
The reunion of intersections of extended Voronoi regions has zero measure, that is:

$$\mu_{\mathcal{X}}\left(\bigcup_{1 \leq j \leq k} \bigcup_{1 \leq j' < j} \bar{\mathcal{S}}_j^{(f)} \cap \bar{\mathcal{S}}_{j'}^{(f)}\right) = 0.$$

4 Adaptive choice of k for k-means

In this section, we extend the study of Canas and Rosasco [17], in which they study the regularization properties of k-means and k-flats, compute oracle values of the parameter k (to optimize an upper bound of the error) and get convergence rates. We first recall existing bounds on the statistical and approximation errors and the results of [17]. We then provide a novel method based on complexity regularization in which we adapt proofs of [39, 22].

Usually, the error of equation (14) is decomposed in two parts that are studied independently:

[18] In this section, we suppose that $\mathcal{X}$ is a Euclidean space, because Rademacher's theorem, that we will use to prove results, applies only to open subsets of Euclidean spaces.

Proposition 18 (Decomposition of the error)
The error $\mathcal{E}(\hat{S})$ can be decomposed in two parts:

$$\mathcal{E}(\hat{S}) \leq 2 \underbrace{\sup_{S\in\mathcal{H}} |\mathcal{E}(S) - \hat{\mathcal{E}}(S)|}_{\textit{Statistical error}} + \underbrace{\mathcal{E}(S^*)}_{\textit{Approximation error}},$$

where S^* is the optimal set (that cannot be computed in practice)

$$S^* = \arg\min_{S\in\mathcal{H}} \mathcal{E}(S).$$

Proof

$$\begin{aligned}
\mathcal{E}(\hat{S}) &= \underbrace{\hat{\mathcal{E}}(S^*) - \mathcal{E}(S^*)}_{\leq \sup_{S\in\mathcal{H}} |\mathcal{E}(S)-\hat{\mathcal{E}}(S)|} + \underbrace{\mathcal{E}(\hat{S}) - \hat{\mathcal{E}}(\hat{S})}_{\leq \sup_{S\in\mathcal{H}} |\mathcal{E}(S)-\hat{\mathcal{E}}(S)|} + \underbrace{\hat{\mathcal{E}}(\hat{S}) - \hat{\mathcal{E}}(S^*)}_{\leq 0} + \mathcal{E}(S^*) \\
&\leq 2\sup_{S\in\mathcal{H}} |\mathcal{E}(S) - \hat{\mathcal{E}}(S)| + \mathcal{E}(S^*). \quad \blacksquare
\end{aligned}$$

Using this decomposition, we will try to give a theoretical explanation to the trade-off (see figure 7-(d)) we observe on some distributions between the statistical and the approximation error. The presence of this trade-off in absence of noise is somewhat surprising as we would expect the error of the estimator to always go down as the number of means increase. To draw a parallel with supervised learning, the approximation error can be compared to the bias as it (asymptotically) decreases with the number of means whereas the statistical error would correspond to the variance.

4.1 Study of the performance of k-means

We now study independently the approximation and the statistical error of k-means. The approximation error will make use of results we mentioned in section 3.2.2 about optimal quantization, whereas the study of the statistical error require techniques based on Gaussian and Rademacher complexities.

Approximation error

Let $\mathcal{E}_k$ be the best error achievable over sets of size k *i.e.*:

$$\mathcal{E}_k = \inf_{S\in\mathcal{P}_k} \mathcal{E}(S).$$

We use proposition 15 to upper-bound $\mathcal{E}_k$. We have (see proposition 15 for the definition of $\mathcal{E}_{\mathcal{M},k}$):

$$\begin{aligned}
\mathcal{E}_{\mathcal{M},k} &= \inf_{S\in\mathcal{P}_k^{(\mathcal{M})}} \int_{\mathcal{M}} d_I(x,S)\rho(x)d\mu_I(x) \\
&\underset{d_I(x,y)\geq d_\chi(x,y)}{\geq} \inf_{S\in\mathcal{P}_k^{(\mathcal{M})}} \int_{\mathcal{M}} d_\chi(x,S)\rho(x)d\mu_I(x) \\
&\underset{\mathcal{P}_k^{(\mathcal{M})}\subset\mathcal{P}_k}{\geq} \inf_{S\in\mathcal{P}_k} \int_{\mathcal{M}} d_\chi(x,S)\rho(x)d\mu_I(x) = \mathcal{E}_k.
\end{aligned}$$

In our setting, we suppose that the probability measure p is absolutely continuous with respect to the volume measure μ_I on the manifold (with a density ρ), so we can apply the result of proposition 15 with $\rho_a = \rho$:

$$\lim_{k\to\infty} \mathcal{E}_k.k^{\frac{2}{d}} \leq C_d \left(\int_{\mathcal{M}} \rho(x)^{\frac{d}{d+2}} d\mu_I(x) \right)^{\frac{d+2}{d}}, \tag{21}$$

and thus the asymptotic behavior of the appproximation error with respect to the number of means k is $\mathcal{E}_k = \mathcal{O}(k^{-\frac{2}{d}})$, and in particular, the approximation error is a asymptotically decreasing with the number of means.

However, for some distributions (especially on high-dimensional data), we observe a trade-off (see figure 7-(d)) between the statistical and the approximation error. This indicates that for some distributions, the statistical error must asymptotically be an increasing function of k. We study the statistical error in the next section.

Statistical error

Proposition 19 [18]
Let ρ be such that supp$(\rho) \subseteq \mathcal{B}(0,1)^d$. Then:

$$\mathbb{P}\left(\sup_{S\in\mathcal{P}_k} |\mathcal{E}(S) - \hat{\mathcal{E}}(S)| \leq \frac{k\sqrt{18\pi}}{\sqrt{n}} + \sqrt{\frac{8\log 1/\delta}{n}} \right) \geq 1-\delta.$$

By putting together these two results, we get the following result:

Proposition 20 [17]
If $\hat{S}_k$ is the minimizer of the empirical error (1) over sets of points of size k then, for $\delta \leq \frac{1}{e}$, there are constants C_d and γ_d dependent only on d, and a sufficiently large N such that, by setting

$$k_n = n^{\frac{d}{2(d+2)}}.\left(\frac{C_d}{24\sqrt{\pi}}\right)^{\frac{d}{d+2}}.\int_{\mathcal{M}} \rho(x)^{\frac{d}{d+2}} d\mu_I(x), \tag{22}$$

and $\hat{S}^* = \hat{S}_{k_n}$, it is

$$\forall n \geq N, \quad \mathbb{P}\left(\mathcal{E}(\hat{S}^*) \leq \gamma_d.\int_{\mathcal{M}} \rho(x)^{\frac{d}{d+2}} d\mu_I(x) \sqrt{\ln\left(\frac{1}{\delta}\right)} n^{-\frac{1}{d+2}} \right) \geq 1-\delta,$$

where $C_d \sim d$ and γ_d grows sublinearly with d.

Remark 2 *The choice of k provided in proposition 20 is obtained by optimizing the sum of the bound on the statistical error (proposition 19) and the bound on the approximation error (equation (21)).*

4.2 Complexity regularization

As mentioned before, the error $\mathcal{E}(\hat{S}_k)$ may not be a decreasing function of the parameter k and in this case, the value of k^* minimizing the $\mathcal{E}(\hat{S}_k)$ will be between 1 and n. In equation (22), we discussed a choice of the number of means k_n and computed the associated bound on the reconstruction error. However, such parameter choice depends on quantities that we

do not know in practice such as the dimension of the manifold $\mathcal{M}$ and more importantly, the unknown distribution ρ. To address this problem, we study a method based on complexity regularization [39] and draw ideas from [22]. Complexity regularization is based on the idea of penalizing the empirical reconstruction error (equation (1)). The penalty increases as the model complexity (in our case the size of the approximating set) increases. The hope is that the penalized reconstruction error and the reconstruction error will have the same behavior with respect to the parameter k.
In the propositions and their proofs (which are relegated in appendix B.6), we use the following notations:

$$\begin{aligned}\hat{S}_k &= \arg\min_{S\in\mathcal{P}_k} \hat{\mathcal{E}}(S),\\ S_k &= \arg\min_{S\in\mathcal{P}_k} \mathcal{E}(S),\\ S^* &= \arg\min_{S\in\mathcal{H}} \mathcal{E}(S).\end{aligned}$$

We introduce a penalty $p(k,n)$ that is likely to have the same variations as the statistical error with respect to the number of means k. Based on the bound we have in proposition 19, we choose:

$$p(k,n) = \sqrt{18\pi}\frac{k}{\sqrt{n}} + 4\sqrt{\frac{\ln k}{n}},$$

and the associated penalized error $\tilde{\mathcal{E}}$ is:

$$\tilde{\mathcal{E}}(\hat{S}_k) = \hat{\mathcal{E}}(\hat{S}_k) + p(k,n). \tag{23}$$

Having the $\tilde{\mathcal{E}}(\hat{S}_k)$ for $1 \leq k \leq n$, we compute the value $\hat{k}$ that minimizes the penalized risk, that is:

$$\hat{k} = \arg\min_{1\leq k\leq n} \tilde{\mathcal{E}}(\hat{S}_k),$$

and the set we obtain is denoted by $\tilde{S} = \hat{S}_{\hat{k}}$. The following propositions study the closeness between $\mathcal{E}(S^*)$ and $\mathcal{E}(\tilde{S})$. The first result we prove studies the closeness between the penalized reconstruction error and the reconstruction error on the set $\tilde{S}$ obtained by minimization of the penalized error.

Proposition 21

$$\forall \epsilon > 0,\ \mathbb{P}\left(\mathcal{E}(\tilde{S}) - \tilde{\mathcal{E}}(\tilde{S}) > \epsilon\right) \leq \frac{\pi^2}{6} e^{-\frac{\epsilon^2 n}{8}}.$$

However, the quantity in which we are interested is the difference between the reconstruction error of $\tilde{S}$ and the reconstruction error of the optimal set S^*. The following proposition studies the expected value of this difference.

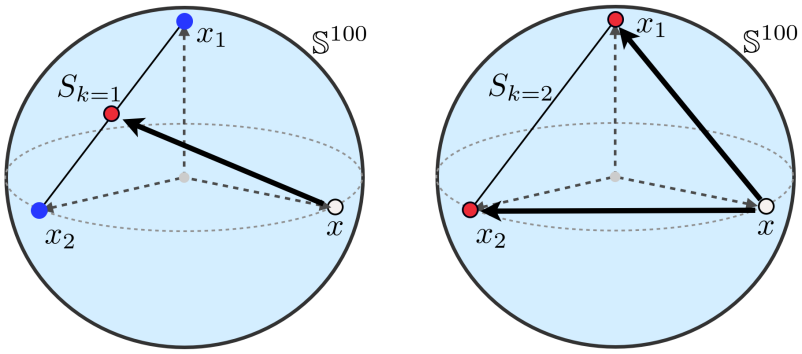
Proposition 22

$$\mathbb{E}\left(\mathcal{E}(\tilde{S})\right) - \mathcal{E}(S^*) \leq \min_{1\leq k\leq n}\left(p(k,n) + \inf_{S\in\mathcal{P}_k}\left(\mathcal{E}(S) - \mathcal{E}(S^*)\right)\right) + \sqrt{\ln\left(\frac{e\pi^2}{6}\right)\frac{8}{n}}.$$

Proposition 23 provides a concentration inequality for the expected risk of set $\tilde{S}$. The result of this proposition shows that up to the penalty $p(k,n)$ plus a factor $\sqrt{\frac{\ln k}{n}}$ the expected risk of $\tilde{S}$ is close to the best reconstruction error achievable with exponential probability. This result shows that the choice $\hat{k}$ of the parameter k provided by the minimization of the penalized reconstruction error is interesting.

Proposition 23

$$\forall \epsilon > 0,\ \mathbb{P}\left(\mathcal{E}(\tilde{S}) > \min_{1\leq k\leq n}\left(\mathcal{E}(S_k) + p(k,n) + 4\sqrt{\frac{\ln k}{n}}\right) + \epsilon\right) \leq \frac{\pi^2}{3}e^{-\frac{n\epsilon^2}{32}}.$$



(a) $\mathcal{E}(S_{k=1}) \simeq 3/2$ (b) $\mathcal{E}(S_{k=2}) \simeq 2$

Figure 5: The solutions of k-means (in red) on a data set composed of $n = 2$ samples (in blue) drawn on a 100-dimensional unit sphere for $k = 1$ (a), and $k = 2$ (b).

5 Experiments

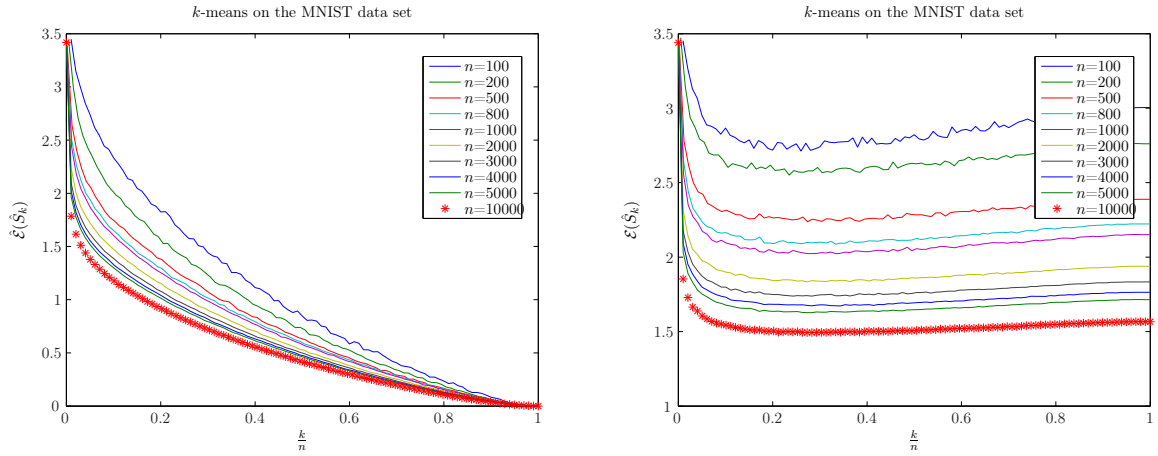
5.1 k-means

The first experiment (figure 7) we make is on data sets composed by unit spheres of different dimensions. The purpose of this experiment is to show that there exists different regimes when running k-means. By different regime, we mean that the variation of the testing error with respect to k is different.
Indeed, the training error (figures 7-(a),(c),(e)) is always a decreasing function of the number of means k whatever the dimension d of the sphere and the size n of the training set. However, the testing error can have different behaviors depending on the dimension d of the sphere with respect to the size n of the training set. From our experiments, we observe that:

- when the training set is big enough compared to the dimension of the sphere ($n \geq 10d$), the testing error is monotically decreasing (figure 7-(b)).
- when the training set is too small compared to the dimension of the sphere (typically $n < d$), the testing error is monotically increasing (figure 7-(f)).
- there exists an intermediary regime in between ($n \sim d$): the testing error decreases until a certain value of k and then increases (figure 7-(d)). This is what we call a *trade-off*.

The causes of the existence of these three regimes, and in particular of a trade-off is not well understood. An intuition is that when the number of samples is too small compared to the dimension, it is better to group several means together to average on several points. Thus the rule "the more means, the better" does not, in general, hold (see example below and figure 5-(d)).

Example [17]
Consider a setup in which $n = 2$ samples x_1, x_2 are drawn from a uniform distribution on the 100-dimensional unit sphere ($d = 100$). Because $d \gg n$, with high probability, the samples are nearly orthogonal: $\langle x_1, x_2 \rangle \simeq 0$, while a third sample x drawn uniformly on $\mathcal{S}(0,1)^{100}$ will also very likely be nearly orthogonal to both x_1, x_2 [40]. The k-means solution on this dataset is clearly $S_{k=1} = \{(x_1 + x_2)/2\}$ (figure 5-(a)), and $S_{k=2} = \{x_1, x_2\}$ (figure 5-(b)), and therefore $\mathcal{E}(S_{k=1}) \simeq 3/2 < 2 \simeq \mathcal{E}(S_{k=2})$ with very high probability. In this case, it is better to place a single mean closer to the origin (with $\mathcal{E}(\{0\}) = 1$), than to place two means at the sample locations.



(a) Training error

(b) Testing error

Figure 6: Training and testing error for Lloyd's algorithm as a function of the number of means $\frac{k}{n}$ (n is the size of the training set) for different values of n. The testing set contains ten times more points than the training set.

We perform a second round of experiments on a real-world data set (figure 6), MNIST, that is composed of images of handwritten digits [41]. We plot the training error (figure 6-(a)) and the testing error (figure 6-(b)) for different sizes of the training set. We observe that:

- the training error is monotically decreasing for all sizes n of the training set.
- the testing error shows a trade-off and is minimal for a value of k between $0.05n$ and $0.2n$.
- given a value of $\frac{k}{n}$, the training and the testing errors are decreasing functions of the training set size n.

This experiment confirms the existence of a trade-off and proves it does not only happen for toy data sets.

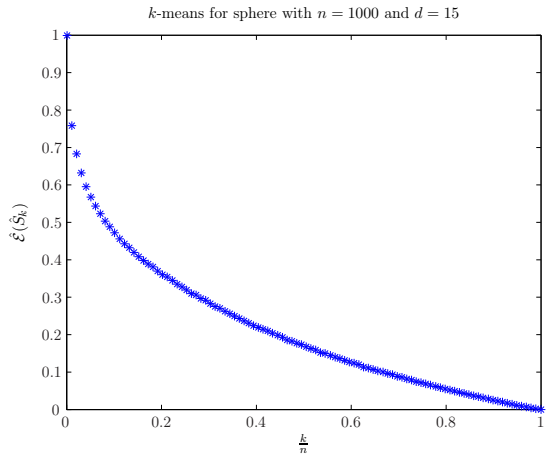
5.2 k-flats

In this section, we discuss the influence of the free parameters k and m of k-flats on the training and the testing error (figure 8). We observe the following trends:

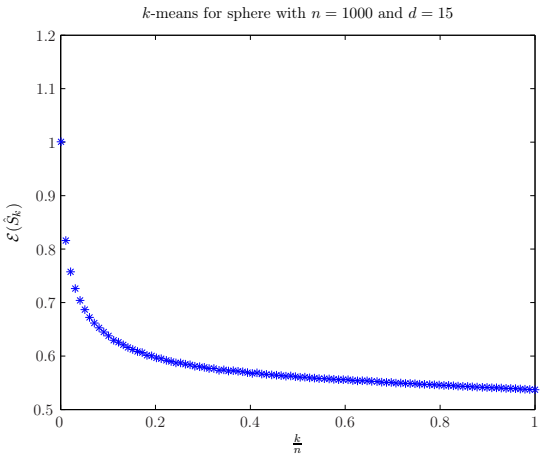
- if k (respectively m) is fixed, the training error is monotically decreasing with m (respectively k). This fact seems intuitive as we expect the training error to decrease both when the number of flats increases, and when the dimension of the flats increases. In fact, when $k(m+1) \geq n$ the training error is zero. This is perfectly normal as a flat of dimension m can contain at least $m+1$ points and thus k flats, if not equal, can contain $k(m+1)$ points. If the k flats contain the n samples, the training error is zero as $d_{\mathcal{X}}(x_i, S) = 0 \ \forall i \in [\![1, n]\!]$.
- the testing error decreases with the dimension of the flats m. This can be justified by the fact that for a given group of points $Y_n = (y_1, \cdots, y_n) \in \mathbb{R}^{d \times n}$, if F_l denotes the flat computed by algorithm 5 (with $k = 1$), it holds: $F_l \subset F_{l+1} \ \forall l \in [\![1, d]\!]$. Indeed, if $(f_1, \cdots, f_d)$ are the eigenvectors of $Y_n Y_n^\top$ associated with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$, we have:

$$F_l = \bar{y} + \text{span}(f_1, \cdots, f_l) \subset \bar{y} + \text{span}(f_1, \cdots, f_{l+1}) = F_{l+1}.$$

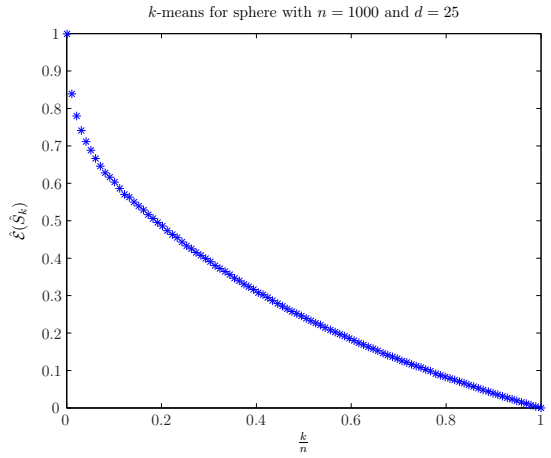
- for a fixed m, we observe a small trade-off in the variation of the testing error with k (the testing error is minimal when k is about $0.9n$), and the amplitude of this trade-off slightly decreases as m increases.



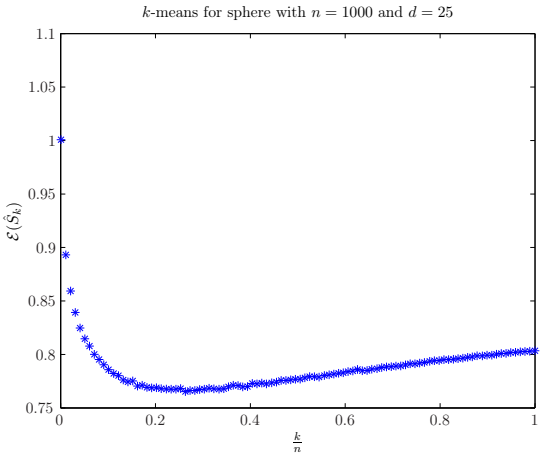
(a) d=15, training



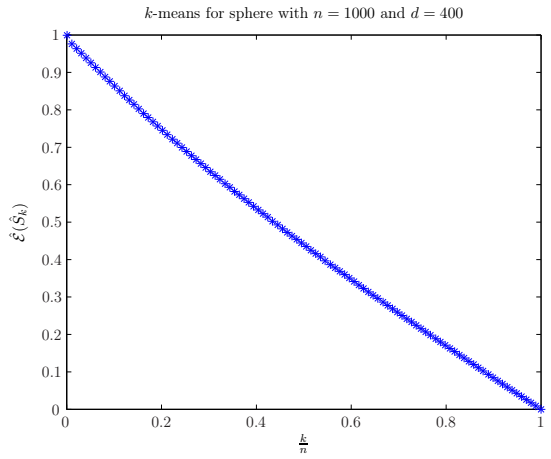
(b) d=15, testing



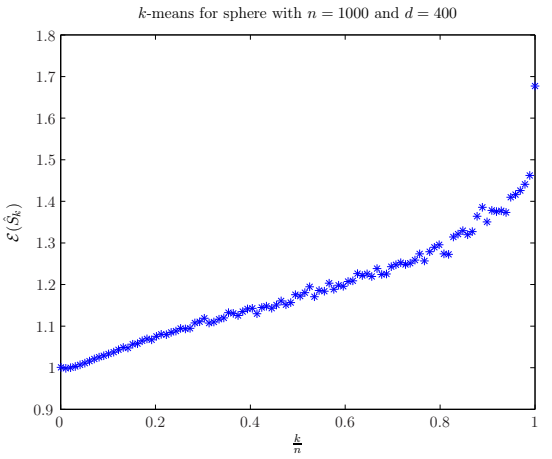
(c) d=25, training



(d) d=25, testing

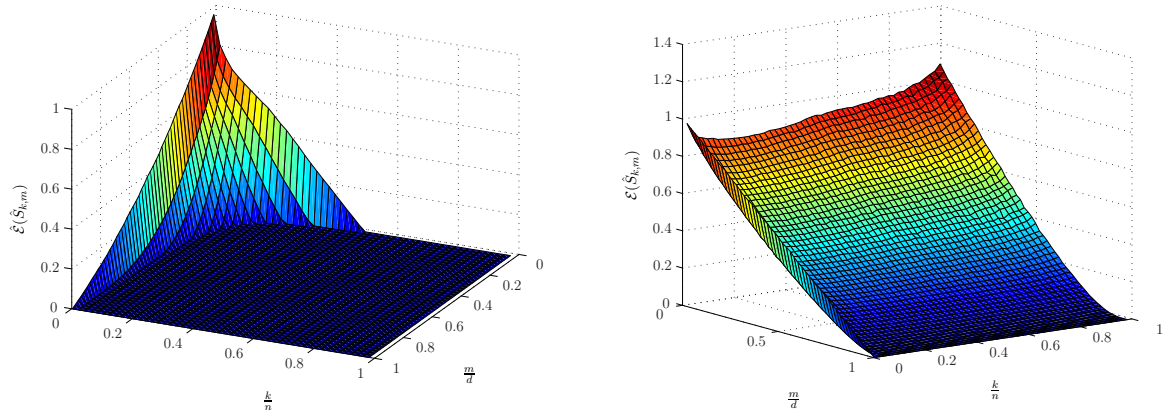


(e) d=400, training



(f) d=400, testing

Figure 7: Training and testing error for Lloyd's algorithm as a function of the number of means $\frac{k}{n}$. The data set is a composed of $n = 1000$ points belonging to a d-dimensional ($d \in \{15, 25, 400\}$) unit sphere. The testing set is composed of 10000 points.



(a) Training error

(b) Testing error

Figure 8: Training and testing error for algorithm 3 as a function of the number of flats $\frac{k}{n}$ ($n = 500$) and the dimension of each flat $\frac{m}{d}$ ($d = 50$). The data set is a composed of $n = 500$ points belonging to a 50-dimensional unit sphere. The testing set is composed of 1500 points.

5.3 Kernel k-means

For the experiment of figure 9, we used the same data set as in the experiment of figure 7. The choice of the value of the parameter σ is explained below.
We observe several things:

- the training error is monotically decreasing with k.
- we observe a slight trade-off on figures 9-(b),(d), but the testing error is monotically increasing for $k > 0.05n$. This would suggest that the optimal choice of k (with a view to minimize the testing error) for kernel k-means on this particular data set would be much smaller than for the usual k-means.

5.4 Kernel k-flats

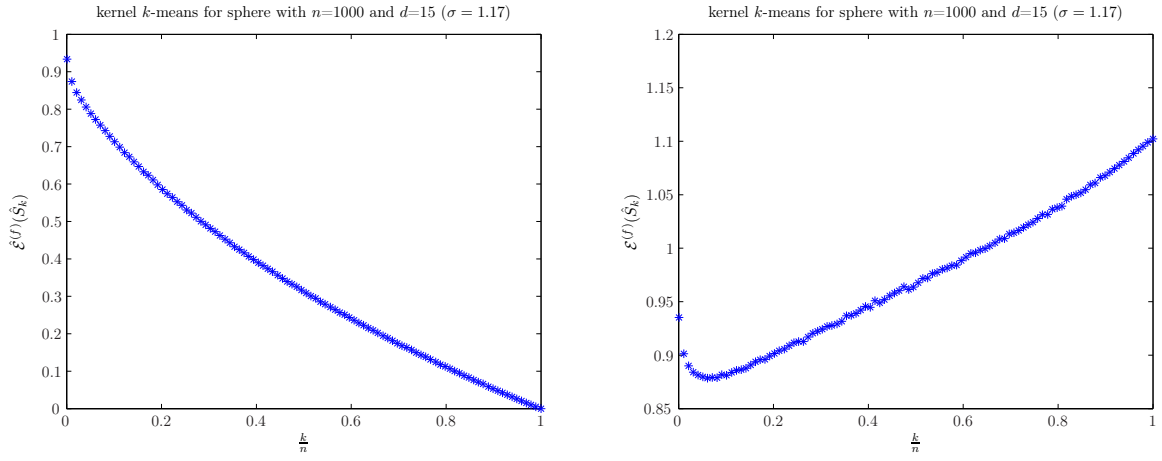
To test kernel k-kfats, we used a data set composed of a 50-dimensional sphere with $n = 1000$ points in the training set and 2000 points in the testing set. We computed flats of dimension $m = 2$ (figures 10-(a),(b)) and dimension $m = 10$ (figures 10-(c),(d)). We limited the number of flats to values between 1 and $0.2n$ for $m = 2$ and $0.08n$ for $m = 10$ (see section 5.6.2 for a practical reason to this limitation). We observe the following facts:

- the training error goes down with the number of flats both for $m = 2$ and $m = 10$.
- we observe a trade-off in the testing error when $m = 2$. The optimal value of k is around $0.1n$. For $m = 10$, we do not observe such a trade-off, but the range of values that k can take is more than two times smaller than for $m = 2$.

On the choice of the kernel and its parameter

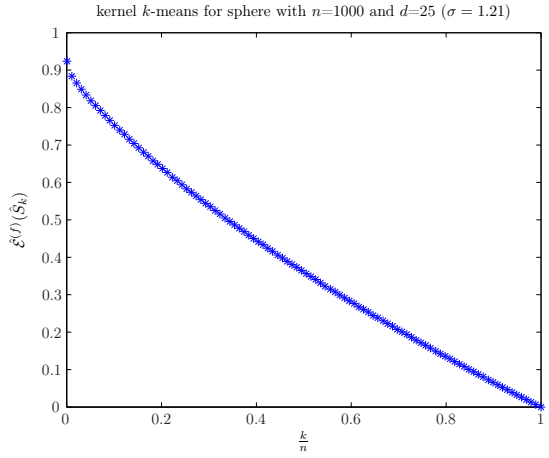
We used a gaussian kernel with parameter σ in our experiments on kernel k-means and kernel k-flats.
For the experiments of figures 9 and 10, the value of σ has been determined by taking the median of the distribution of radiuses such that a ball centered on a data point contains $n/10$ points. More precisely,

$$\sigma = \text{median}\left\{r_i,\ i \in [\![1, n]\!]\right\},\ \text{where } r_i \text{ is such that } \mathcal{B}(x_i, r_i) \text{ contains } \frac{n}{10} \text{ points.}$$

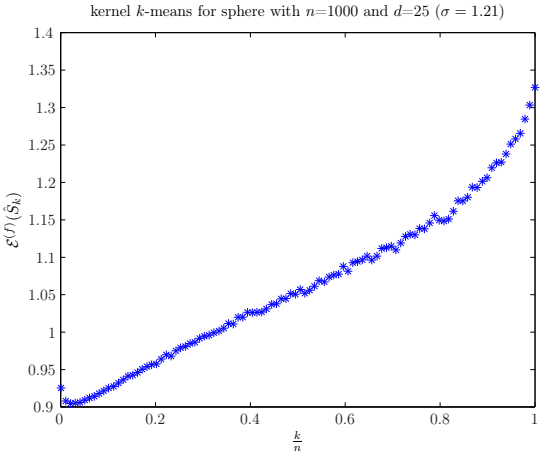


(a) d=15, training

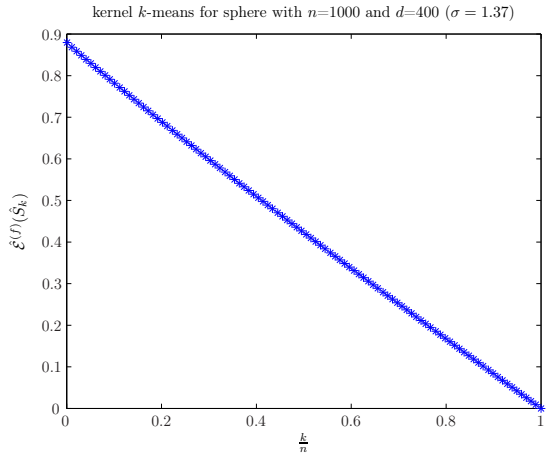
(b) d=15, testing



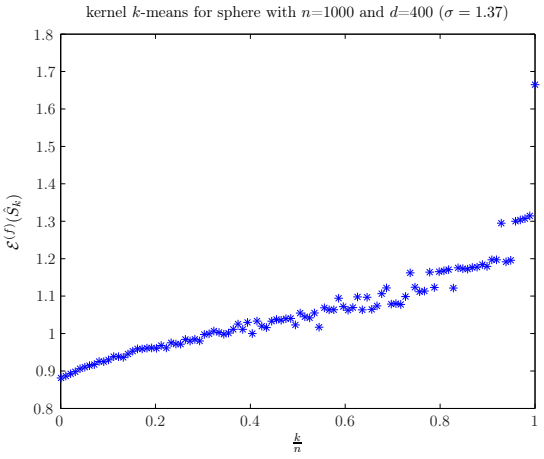
(c) d=25, training



(d) d=15, testing

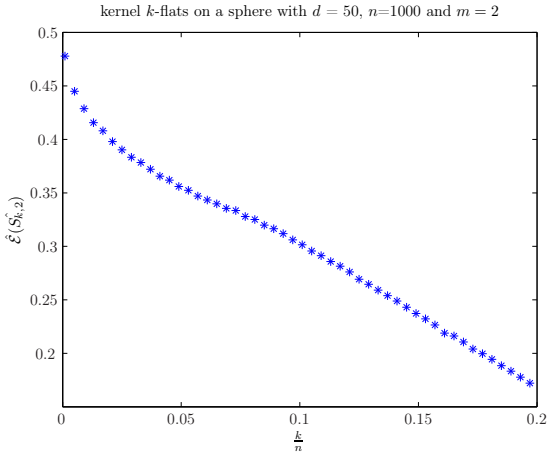


(e) d=400, training

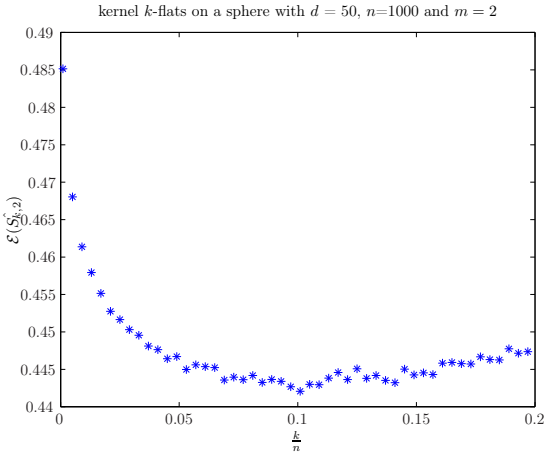


(f) d=400, testing

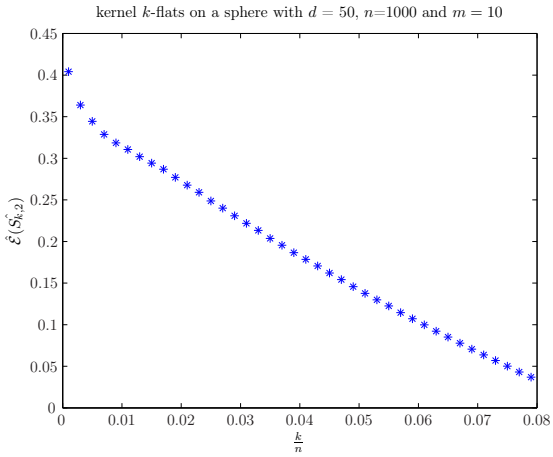
Figure 9: Training and testing error for algorithm 4 as a function of the number of means $\frac{k}{n}$. The data set is a composed of $n = 1000$ points belonging to a d-dimensional ($d \in \{15, 25, 400\}$) unit sphere. The testing set is composed of 10000 points. The kernel used is a gaussian kernel with parameters $\sigma = \{1.17, 1.21, 1.37\}$ respectively for $d = \{15, 25, 400\}$.



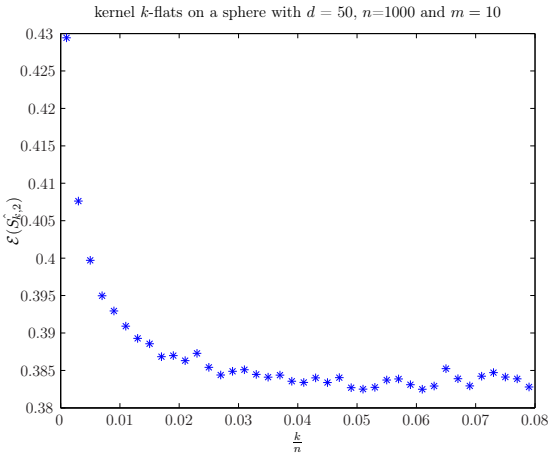
(a) Training error, $m = 2$



(b) Testing error, $m = 2$



(c) Training error, $m = 10$



(d) Testing error, $m = 10$

Figure 10: Training and testing error for algorithm 5 as a function of the number of flats $\frac{k}{n}$ ($n = 1000$) for flats of dimension $m = 2$ ((a) and (b)) and $m = 10$ ((c) and (d)). The data set is a composed of $n = 1000$ points belonging to a 50-dimensional unit sphere. The testing set is composed of 2000 points. The value of the parameter σ used for the gaussian kernel is $\sigma = 1.2$.

5.5 About k-means' initialization

One of Lloyd's algorithm drawback is that the configuration reached at the convergence highly depends on the initialization. Several initializations have been studied and compared [42]. In our simulations we used the following ones:

- Random: pick the means $(m_j)_{1\leq j\leq k}$ among the data points with uniform probability.
- k-means++: pick the first mean among data points with uniform probability. Pick the other means among data points with a probability proportional to the squared distance to the closest existing mean (see section 1.1.3 for more details).
- ϵ-net: pick the first mean among data points with uniform probability. Pick the following means to be the farthest point from existing means *i.e.*:

$$m_j = x_{i^*} \text{ with } i^* = \arg\max_{1\leq i\leq n} D(x_i) \text{ and } D \text{ denotes the distance to the closest mean.}$$

When not specified, the initialization we used for k-means is k-means++.

For all these initializations, the output is not deterministic. To address this problem, an algorithm called *global k-means* has been proposed in [43]. Its principle is the following:

- for k=1, choose the mean $m_1^{(j)}$ to be the center of mass.
- for a given $2 \leq j \leq k$, let $(m_1^{(j-1)}, \ldots, m_{j-1}^{(j-1)})$ be the configuration obtained after having run Lloyd's algorithm with $j-1$ means. Run Lloyd's algorithm for j means with n different initializations $(m_1^{(j-1)}, \ldots, m_{j-1}^{(j-1)}, x_i)$ (for $1 \leq i \leq n$) and pick the ouput that minimizes the objective function (3).

In a nutshell, global k-means runs n times Lloyd's algorithm at each step and solves j-means for every j between 1 and k using the output of $j-1$-means.

5.6 Practical problems

In this section, we briefly mention the main practical problems encountered during the implementation of the algorithms.

5.6.1 Number of means

In the implementation of the various algorithms mentioned in this report, the most important quantity is the assignement vector C because every quantity can be computed from it. When we want to obtain k means (or k flats depending on the algorithm), C must take values between 1 and k. If during the process, C does not take one of these values *i.e.*

$$\exists j \in [\![1, k]\!],\ \forall i \in [\![1, n]\!],\ C(i) \neq j,$$

one of the cluster has become empty and it will stay empty until convergence is reached. When this kind of behavior happens, we just start from scratch with a new seeding.

5.6.2 Dimension of the flats in kernel k-flats

As we have seen in section 2.3.2, the computation of flats in kernel k-flats requires the computation of k submatrices ($K_j \in \mathbb{R}^{n_j \times n_j}$) of the kernel matrix. On each of these matrices, we need to find m eigenvectors, that is the condition $m \leq n_j \ \forall j \in [\![1, k]\!]$ must hold. This condition is somewhat restrictive in practice as it imposes the training set to be of a sufficient size (at least $n \geq km$) for the algorithm not to stop prematurely (we did not have this kind of condition with the usual k-flats as each of the submatrices $(Z_j)_{1 \leq j \leq k}$ (see section 1.2.2) belonged to $\mathbb{R}^{d \times d}$ and by assumption $m \leq d$).

Conclusion

In this report, we studied theoretical and numerical properties of piece-wise estimators (k-means and k-flats). In particular, we discussed their kernel extensions, provided an adaptive parameter choice, and discussed connections of our framework to related ones such as optimal quantiztion.

Our study points to several further questions. Among other, we note that it would be interesting to provide lower bounds for kernel k-means and k-flats. More generally, it would be interesting to further study the reconstruction properties of these methods when the reconstruction is measured with respect to the metric of the data.

A Singular Value Decomposition

See for instance [31], pages 487-488, and [44].

Proposition 24 (Singular value decomposition)
Let $M \in \mathbb{R}^{d\times n}$. Then:

$$\exists U \in \mathbb{R}^{d\times d},\ V \in \mathbb{R}^{n\times n},\ \Sigma \in D_{dn}, \begin{cases} M = U\Sigma V^\top \\ U^\top U = I_d \\ V^\top V = I_n \end{cases}$$

The d columns of U are called the left singular vectors and the n columns of V are called the right singular vectors.

Proposition 25 (Eigenvalues of $MM^\top$ and $M^\top M$)
Let $M \in \mathbb{R}^{d\times n}$. Then $MM^\top$ and $M^\top M$ have exactly the same nonzero eigenvalues with the same multiplicity. Their nonzero eigenvalues are the square of the nonzero numbers found on the diagonal of Σ. Moreover, the d (respectively n) columns of U (respectively V) are the eigenvectors of $MM^\top$ (respectively $M^\top M$) and they are related by:

$$\begin{aligned} MV &= U\Sigma \\ M^\top U &= V\Sigma^\top \end{aligned}$$

B Proofs

B.1 Section 3.2.1

Proof of proposition 11
We use a proof by contradiction: suppose that $\exists l, l' \in [\![1,k]\!], l \neq l'$ and $m_l = m_{l'}$.
Let O be the cost that we want to minimize:

$$O = \sum_{j\in[\![1,k]\!]\setminus\{l,l'\}} \sum_{i\in\mathcal{B}_j} \|x_i - m_j\|^2 + \sum_{i\in\mathcal{B}_l \bigcup \mathcal{B}_{l'}} \|x_i - m_l\|^2 .$$

$$k \leq n \Rightarrow \underbrace{\exists j \in [\![1,k]\!] \setminus \left\{l, l'\right\}, |\mathcal{B}_j| \geq 2}_{i.} \text{ or } \underbrace{|\mathcal{B}_l \bigcup \mathcal{B}_{l'}| \geq 2|}_{ii.} .$$

Case i.:
Let w be such that $|\mathcal{B}_w| \geq 2$: we know that $\exists u \in \mathcal{B}_w, \|x_u - m_w\| > 0$ because the points are different and thus m_w cannot be equal to all the points in $\mathcal{B}_w$.
Setting $m_{l'} = x_u$ and keeping everything else constant, we have:

$$\begin{aligned} O &= \sum_{j\in[\![1,k]\!]\setminus\{l,l',w\}} \sum_{i\in\mathcal{B}_j} \|x_i - m_j\|^2 + \sum_{i\in\mathcal{B}_l \bigcup \mathcal{B}_{l'}} \|x_i - m_l\|^2 + \sum_{i\in\mathcal{B}_w\setminus\{u\}} \|x_i - m_w\|^2 + \underbrace{\|x_u - m_w\|^2}_{>\|x_u - m_{l'}\|^2 = 0} \\ &> \sum_{j\in[\![1,k]\!]\setminus\{l,l',w\}} \sum_{i\in\mathcal{B}_j} \|x_i - m_j\|^2 + \sum_{i\in\mathcal{B}_l \bigcup \mathcal{B}_{l'}} \|x_i - m_l\|^2 + \sum_{i\in\mathcal{B}_w\setminus\{u\}} \|x_i - m_w\|^2 + \|x_u - m_{l'}\|^2 \end{aligned}$$

which contredicts the optimality of $\{m_1, \ldots, m_k\}$.

Case ii.:
By the same reasoning as before, $\exists u \in \mathcal{B}_l \bigcup \mathcal{B}_{l'}, \|x_u - m_l\| > 0$. Setting $m_{l'} = x_u$ and keeping everything else constant, we have:

$$\begin{aligned} O &= \sum_{j \in [\![1,k]\!] \setminus \{l,l'\}} \sum_{i \in \mathcal{B}_j} \|x_i - m_j\|^2 + \sum_{i \in \mathcal{B}_l \bigcup \mathcal{B}_{l'} \setminus \{u\}} \|x_i - m_l\|^2 + \underbrace{\|x_u - m_l\|^2}_{> \|x_u - m_{l'}\|^2 = 0} \\ &> \sum_{j \in [\![1,k]\!] \setminus \{l,l'\}} \sum_{i \in \mathcal{B}_j} \|x_i - m_j\|^2 + \sum_{i \in \mathcal{B}_l \bigcup \mathcal{B}_{l'} \setminus \{u\}} \|x_i - m_l\|^2 + \|x_u - m_{l'}\|^2 \end{aligned}$$

which contredicts the optimality of $\{m_1, \ldots, m_k\}$. ∎

Proof of proposition 12
We use a proof by contradiction: we suppose that $\{m_1, \ldots, m_k\}$ is an optimal set and that there exists at least one point exactly halfway between two means. To obtain the contradiction, we show that we can strictly decrease the objective O.
Let us write the objective as follows:

$$O = \sum_{j=1}^{k} \sum_{i \in \mathcal{B}_j} \|x_i - m_j\|^2 + \sum_{i \in \mathcal{R}} \left\|x_i - m_{C(i)}\right\|^2,$$

where $\mathcal{R}$ is defined in equation (17).

We distinguish two cases:

- Case i.: $\exists w \in [\![1,k]\!], m_w \neq \frac{\sum_{i \in \mathcal{B}_w} x_i}{n_w}$
- Case ii.: $\forall w \in [\![1,k]\!], m_w = \frac{\sum_{i \in \mathcal{B}_w} x_i}{n_w}$

Case i.:
Let w be such that $m_w \neq \frac{\sum_{i \in \mathcal{B}_w} x_i}{n_w}$. We show that we can decrease the cost by setting $m_w = \bar{x}_w$.

$$\begin{aligned} O &= \sum_{j \in [\![1,k]\!] \setminus \{w\}} \sum_{i \in \mathcal{B}_j} \|x_i - m_j\|^2 + \underbrace{\sum_{i \in \mathcal{B}_w} \|x_i - m_w\|^2}_{> \sum_{i \in \mathcal{B}_w} \|x_j - \bar{x}_w\|^2} + \sum_{i \in \mathcal{R}} \left\|x_i - m_{C(i)}\right\|^2 \\ &> \sum_{j \in [\![1,k]\!] \setminus \{w\}} \sum_{i \in \mathcal{B}_j} \|x_i - m_j\|^2 + \sum_{i \in \mathcal{B}_w} \|x_i - \bar{x}_w\|^2 + \sum_{i \in \mathcal{R}} \left\|x_i - m_{C(i)}\right\|^2 \end{aligned}$$

which contredicts the optimality of $\{m_1, \ldots, m_k\}$.
Case ii.:
By hypothesis, $\exists u, v \in [\![1,k]\!], u \neq v$ and $\exists w \in [\![1,n]\!], \|x_w - m_u\| = \|x_w - m_v\|$. Then:

$$\begin{aligned} O &= \sum_{j \in [\![1,k]\!] \setminus \{u,v\}} \sum_{i \in \mathcal{B}_j} \|x_i - m_j\|^2 + \sum_{i \in \mathcal{R}} \left\|x_i - m_{C(i)}\right\|^2 + \sum_{i \in \mathcal{B}_u} \|x_i - \bar{x}_u\|^2 - \|x_w - \bar{x}_v\|^2 \\ &+ \underbrace{\sum_{i \in \mathcal{B}_v} \|x_i - \bar{x}_v\|^2 + \|x_w - \bar{x}_v\|^2}_{\Delta} \end{aligned}$$

Let us define $\bar{x}'_v = \frac{\sum_{i \in \mathcal{B}_v} x_i + x_w}{n_v + 1}$

$$\Delta \underset{(*)}{\geq} \sum_{i \in \mathcal{B}_v} \left\|x_i - \bar{x}'_v\right\|^2 + \left\|x_w - \bar{x}'_v\right\|^2$$

and the above inequality is strict as soon as $\bar{x}'_v \neq \bar{x}_v \Leftrightarrow x_w \neq \bar{x}_v$.
Suppose that (*) is an equality *i.e.* $x_w = \bar{x}_v$. As $\|x_w - m_u\| = \|x_w - m_v\|$, this implies that $x_w = \bar{x}_v = \bar{x}_u$. But as we have seen in proposition 11, this is not possible (two means cannot coincide) and thus the inequality (*) is strict and we get the expected contradiction on the optimality of $\{m_1, \ldots, m_k\}$. ∎

B.2 Section 1.2.3

Proof of lemma 3
Let $f : \mathcal{X} \to \mathbb{R}$ be the function that we want to minimize:

$$f(y) = \sum_{i=1}^{n} \left(\|x_i - y\|^2 - \sum_{l=1}^{m} \langle x_i - y, g_l \rangle^2 \right)$$

We compute its derivative with respect to y

$$\nabla_y f(y) = 2 \sum_{i=1}^{n} (y - x_i) - \sum_{i=1}^{n} \sum_{l=1}^{m} 2\langle y - x_i, g_l \rangle g_l$$

We look for solution(s) to $\nabla_y f(y) = 0$:

$$\begin{aligned}
\nabla_y f(y) = 0 \quad &\Leftrightarrow \quad 0 = y - \frac{\sum_{i=1}^n x_i}{n} - \sum_{l=1}^{m} \langle y - \frac{\sum_{i=1}^n x_i}{n}, g_l \rangle g_l \\
&\Leftrightarrow \quad y - \frac{\sum_{i=1}^n x_i}{n} = \pi_{\mathcal{G}}(y - \frac{\sum_{i=1}^n x_i}{n}) \text{ where } \mathcal{G} = \text{span}(g_1, \cdots, g_m) \\
&\Leftrightarrow \quad y - \frac{\sum_{i=1}^n x_i}{n} \in \mathcal{G}. \quad (*)
\end{aligned}$$

We now show that f is convex by computing the Hessian matrix:

$$\nabla^2_{yy} f(y) = \underbrace{I_d - GG^\top}_{H},$$

where G is the matrix with the $(g_l)_{1 \leq l \leq m}$ as columns. The Hessian of f has no dependency on space and we will note it H. It is clear that H is symmetric, so we can prove that it is non negative by studying its minimal eigenvalue. We will conclude that f is convex thanks to the following property:

$$f \text{ convex } \Leftrightarrow H \geq 0.$$

As $G^\top G = I_m$, the Singular Value Decomposition theorem (see proposition 25) allows us to say that ($\lambda_{\max}$ and $\lambda_{\min}$ denote respectively the maximum and minimum eigenvalues):

$$\lambda_{\max}(GG^\top) = \lambda_{\max}(G^\top G) = 1,$$

and thus $\lambda_{\min}(H) = 0$ and f is convex.
(*) implies that f reaches its minimum on the space $\overline{x} + \mathcal{G}$. A particular solution is therefore $y^* = \overline{x}$. ∎

Proof of lemma 4
Z is a d-by-d symmetric matrix and thus

$$\exists\ (e_1,\dots,e_d)\in(\mathbb{R}^d)^d, (\lambda_1,\dots\lambda_d)\in\mathbb{R}^d, \begin{cases}\lambda_1\geq\cdots\geq\lambda_d,\\ \langle e_l,e_{l'}\rangle=\delta_{ll'},\\ Ze_l=\lambda_l e_l.\end{cases}$$

$$\begin{aligned}\sum_{i=1}^{n}\sum_{l=1}^{m}\langle x_i', f_l\rangle^2 &= \sum_{l=1}^{m} f_l^\top \tilde{X}_n\tilde{X}_n^\top f_l\\ &= \sum_{l=1}^{m} f_l^\top Z f_l\\ &= \sum_{l=1}^{m}\langle f_l, Zf_l\rangle\end{aligned}$$

We now show by induction that the optimum is reached by choosing $f_l=e_l$.

- Initialization: we want to maximize $\langle x,Zx\rangle$ with $\|x\|=1$. We can decompose x on $(e_1,\dots,e_d)$, $x=\sum_{l=1}^m a_l e_l$:

$$\begin{aligned}\langle x,Zx\rangle &= \langle\sum_{l=1}^{d} a_l e_l, \sum_{l'=1}^{d} a_{l'}\lambda_{l'}e_{l'}\rangle\\ &= \sum_{l=1}^{d} a_l^2\lambda_l\leq\lambda_1,\end{aligned}$$

  and the inequality becomes an equality by choosing $a_1=1,\ a_l=0\ \forall l\geq 2$ *i.e.* $x=e_1$.

- Inductive step: we suppose that $(f_1,\dots,f_{m-1})=(e_1,\dots,e_{m-1})$ for $2\leq m\leq d$ and we want to show that $(f_1,\dots,f_m)=(e_1,\dots,e_m)$.

$$\begin{aligned}\sum_{l=1}^{m}\langle f_l,Zf_l\rangle &= \sum_{l=1}^{m-1}\langle f_l,Zf_l\rangle+\langle f_m,Zf_m\rangle\\ &= \sum_{l=1}^{m-1}\lambda_l+\langle f_m,Zf_m\rangle.\end{aligned}$$

  Let E_{m-1} be the space generated by the $(e_l)_{1\leq l\leq m-1}$. It holds:

$$\lambda_m=\max_{x\in B_{m-1}^{\perp},\|x\|=1}\langle x,Zx\rangle,$$

  and thus $\langle f_m,Zf_m\rangle\leq\lambda_m$ and the maximum λ_m is reached for $f_m=e_m$. ∎

Proof of proposition 3
According to lemma 3,

$$\mathcal{L}\left(\frac{\sum_{i=1}^n x_i}{n},F\right)\leq\mathcal{L}(y,F)\ \ \forall(y,F)\in(\mathcal{X},\Omega_m).$$

Let $\overline{x}$ denote $\frac{\sum_{i=1}^n x_i}{n}$ and consider the recentered data $\tilde{X}_n$, *i.e*:

$$\forall i \in [\![1, n]\!],\ x'_i = x_i - \overline{x}.$$

Then we can define $\tilde{\mathcal{L}} : \Omega_m \to \mathbb{R}$ by

$$\tilde{\mathcal{L}}(F) = \mathcal{L}(\overline{x}, F),$$

and rewrite the new loss function $\tilde{\mathcal{L}}$:

$$\begin{aligned}
\tilde{\mathcal{L}}(F) &= \sum_{i=1}^n \|x_i - \overline{x} - \pi_F(x_i - \overline{x})\|^2 \\
&= \sum_{i=1}^n \left\|x'_i - \pi_F(x'_i)\right\|^2 \\
&= \sum_{i=1}^n \left\|x'_i\right\|^2 - \sum_{i=1}^n \left\|\pi_F(x'_i)\right\|^2
\end{aligned}$$

and thus minimizing $\tilde{\mathcal{L}}$ over Ω_m is equivalent to maximizing $\tilde{\mathcal{J}} : \Omega_m \to \mathbb{R}$ over Ω_m such that:

$$\tilde{\mathcal{J}}(F) = \sum_{i=1}^n \left\|\pi_F(x'_i)\right\|^2.$$

Using lemma 4, we get that $F^* = \text{span}(e_1, \ldots, e_m)$. ∎

B.3 Computation of the distances to flats in the feature space

$$\begin{aligned}
d_H(\phi(x_i), F_j)^2 &= \left\|\phi(x_i) - m_j - \sum_{l=1}^m \langle \phi(x_i) - m_j, \tilde{\Phi}_n.a_l^{(j)}\rangle \tilde{\Phi}_n.a_l^{(j)}\right\|_H^2 \\
&= \underbrace{\left\|\phi(x_i) - \frac{1}{n_j}\sum_{i\in\mathcal{C}_j}\phi(x_i)\right\|_H^2}_{A_1} - \underbrace{\left\|\sum_{l=1}^m \langle \tilde{\Phi}_n.a_l^{(j)}, \phi(x_i) - \frac{1}{n_j}\sum_{q\in\mathcal{C}_j}\phi(x_q)\rangle \tilde{\Phi}_n.a_j^{(l)}\right\|_H^2}_{A_2} \\
A_1 &= K_{ii} + \frac{1}{n_j^2}\sum_{q,q'\in\mathcal{C}_j} K_{qq'} - \frac{2}{n_j}\sum_{q\in\mathcal{C}_j} K_{iq} \\
A_2 &= \sum_{l=1}^m \langle \tilde{\Phi}_n.a_l^{(j)}, \phi(x_i) - \frac{1}{n_j}\sum_{q\in\mathcal{C}_j}\phi(x_q)\rangle^2 \\
&= \sum_{l=1}^m \left(\sum_{p\in\mathcal{C}_j} (a_l^{(j)})_p \langle \phi(x_i) - \frac{1}{n_j}\sum_{q\in\mathcal{C}_j}\phi(x_q), \phi(x_p) - \frac{1}{n_j}\sum_{q'\in\mathcal{C}_j}\phi(x'_q)\rangle\right)^2 \\
&= \sum_{l=1}^m \left(\sum_{p\in\mathcal{C}_j} (a_l^{(j)})_p \left(K_{ip} - \frac{1}{n_j}\sum_{q\in\mathcal{C}_j}(K_{qp} + K_{iq}) + \frac{1}{n_j^2}\sum_{q,q'\in\mathcal{C}_j} K_{qq'}\right)\right)^2 \\
&= \sum_{l=1}^m \sum_{p,p'\in\mathcal{C}_j} (a_l^{(j)})_{p'}(a_l^{(j)})_p \left(K_{ip'} - \frac{1}{n_j}\sum_{q\in\mathcal{C}_j}\left(K_{qp'} + K_{iq}\right) + \frac{1}{n_j^2}\sum_{q,q'\in\mathcal{C}_j} K_{qq'}\right) \\
&\quad \left(K_{ip} - \frac{1}{n_j}\sum_{q\in\mathcal{C}_j}(K_{qp} + K_{iq}) + \frac{1}{n_j^2}\sum_{q,q'\in\mathcal{C}_j} K_{qq'}\right)
\end{aligned}$$

B.4 Section 2.3.4

Proof of proposition 9
By noticing that kernel k-flats is k-flats in the feature space, we can apply proposition 3:

$$(y^*, F^*) = \left(\frac{\sum_{i=1}^n \phi(x_i)}{n}, \text{span } (e_1, \ldots, e_m) \right),$$

where $(e_1, \ldots, e_m)$ are the eigenvectors associated to the m largest eigenvalues $\lambda_1 \geq \cdots \geq \lambda_m$ of $\tilde{\Phi}_n \tilde{\Phi}_n^\top$.
By the SVD theorem (see appendix A), we know that (with the $(a_l)_{1 \leq l \leq n}$ defined in the proposition):

$$\tilde{\Phi}_n a_l = \lambda_l e_l.$$

And thus, as long as $\lambda_m \neq 0$, we have[19]:

$$\text{span } (e_1, \ldots, e_m) = \text{span } (\tilde{\Phi}_n.a_1, \ldots, \tilde{\Phi}_n.a_m). \quad \blacksquare$$

Proof of proposition 10
Let $l \in [\![1, n]\!]$, $\lambda_l \neq 0$. We begin by proving that:

$$e^\top a_l = \sum_{p=1}^n (a_l)_p = 0 \text{ where } e = \underbrace{(1, \ldots, 1)}_{n \text{ times}}.$$

$$\begin{aligned}
e^\top a_l &= \frac{1}{\lambda_l} e^\top \tilde{K} a_l \\
(e^\top \tilde{K})_p &= \sum_{q=1}^n (\tilde{K})_{qp} \\
&= \sum_{q=1}^n \left\langle \phi(x_q) - \overline{\Phi}, \phi(x_p) - \overline{\Phi} \right\rangle_H \\
&= \Bigg\langle \underbrace{\sum_{q=1}^n \left(\phi(x_q) - \overline{\Phi} \right)}_{0 \text{ by definition of } \overline{\Phi}}, \phi(x_p) - \overline{\Phi} \Bigg\rangle_H = 0.
\end{aligned}$$

And thus:

$$\begin{aligned}
e^\top a_l &= \frac{1}{\lambda_l} \sum_{p=1}^n (e^\top \tilde{K})_p (a_l)_p \\
&= 0. \qquad (24)
\end{aligned}$$

[19] It is useful to mention that if $\lambda_m = 0$, the user can decrease the value of m without increasing the objective function, that is

$$\mathcal{L}\left(y^*, \text{span } (e_1, \ldots, e_m)\right) = \mathcal{L}\left(y^*, \text{span } (e_1, \ldots, e_{m-1})\right).$$

$$\begin{aligned}
\tilde{\Phi}_n a_l &= \sum_{p=1}^{n} (a_l)_p \phi(x_p) \\
&= \sum_{p=1}^{n} (a_l)_p \left(\phi(x_p) - \overline{\Phi}\right) \\
&= \sum_{p=1}^{n} (a_l)_p \phi(x_p) - \underbrace{\left(\sum_{p=1}^{n} (a_l)_p\right)}_{0 \text{ by equation 24}} \overline{\Phi} \\
&= \Phi_n a_l. \quad \blacksquare
\end{aligned}$$

B.5 Sections 3.3.1 and 3.3.2

Proof of proposition 16
Let $f_{ij} : \mathcal{X} \to \mathbb{R}$ be the function defined for $i, j \in [\![1, k]\!]$ by:

$$f_{ij}(x) = \|x - m_i\|^2 - \|x - m_j\|^2 .$$

By definition, $f_{ij}(x) = 0 \Leftrightarrow x \in \bar{\mathcal{S}}_i \cap \bar{\mathcal{S}}_j$. We now show, that $f_{ij}(x) = 0 \Leftrightarrow x \in m_{ij} + \text{span}(\Delta_{ij})^\perp$ where $\Delta_{ij} = m_i - m_j$ and $m_{ij} = \frac{m_i + m_j}{2}$.
As $\text{span}(\Delta_{ij})$ is closed, we can write: $\mathcal{X} = \text{span}(\Delta_{ij}) \oplus T_{ij}$, where $T_{ij} = (\text{span}(\Delta_{ij}))^\perp$.
Let $x \in \mathcal{X}$. x can be written $x = t + \lambda\Delta_{ij},\ t \in T_{ij},\ \lambda \in \mathbb{R}$. Similarly, m_{ij} can be written $m_{ij} = \mu\Delta_{ij} + t',\ t' \in T_{ij}$. We have:

$$\begin{aligned}
f(x) &= \left\|m_{ij} + (\lambda - \mu)\Delta_{ij} + t - t' - m_i\right\|^2 - \left\|m_{ij} + (\lambda - \mu)\Delta_{ij} + t - t' - m_j\right\|^2 \\
&= \left\|(\lambda - \mu - \frac{1}{2})\Delta_{ij} + t - t'\right\|^2 - \left\|(\lambda - \mu + \frac{1}{2})\Delta_{ij} + t - t'\right\|^2 \\
&\underset{\text{Pythagorean theorem}}{=} (|\lambda - \mu - \frac{1}{2}| - |\lambda - \mu + \frac{1}{2}|) \|\Delta_{ij}\|^2
\end{aligned}$$

and thus $f(x) = 0 \Leftrightarrow \lambda = \mu \Leftrightarrow x \in m_{ij} + T_{ij}$.
T_{ij} is a hyperplane so $m_{ij} + T_{ij} = \bar{\mathcal{S}}_i \cap \bar{\mathcal{S}}_j$ is an affine hyperplane, which has zero measure with respect to the Lebesgue measure of $\mathcal{X}$. $\blacksquare$

To prove the same result for k-flats (proposition 17), we need the following lemmas:

Lemma 5
If S is composed of $k \in \mathbb{N}$ affine spaces, $d_{\mathcal{X}}(\cdot, S)$ is 2-Lipschitz.

Proof
Let us write $S = \bigcup_{1 \le j \le k} F_j$ where the $(F_j)_{1 \le j \le k}$ are affine spaces. Let $x, y \in \mathcal{X}$.

$$\exists p, q \in [\![1, k]\!],\ |d_{\mathcal{X}}(x, S) - d_{\mathcal{X}}(y, S)| = |d_{\mathcal{X}}(x, F_p) - d_{\mathcal{X}}(y, F_q)|.$$

Case i.: $p = q$
We write $F = m + G = F_p = F_q$, where G is the vector space associated with F. Then, for any point $x \in \mathcal{X}$, we have (lemma 2)

$$\pi_F(x) = m + \pi_G(x - m),$$

and

$$\begin{aligned} |d_{\mathcal{X}}(x,S) - d_{\mathcal{X}}(y,S)| &= \|x - y - (\pi_F(x) - \pi_F(y))\| = \|x - y - \pi_G(x-y)\| \\ &\leq \|x-y\| + \|\pi_G(x-y)\| \underset{\|\pi_G(x)\| \leq \|x\|}{\leq} 2\|x-y\|. \end{aligned}$$

Case ii.: $p \neq q$

$$\begin{aligned} |d_{\mathcal{X}}(x,S) - d_{\mathcal{X}}(y,S)| &= |d_{\mathcal{X}}(x,F_p) - d_{\mathcal{X}}(y,F_q)| \leq \max(|d_{\mathcal{X}}(x,F_q) - d_{\mathcal{X}}(y,F_q)|, |d_{\mathcal{X}}(x,F_q) - d_{\mathcal{X}}(y,F_q)|) \\ &\underset{\text{Case i.}}{\leq} 2\|x-y\|. \quad \blacksquare \end{aligned}$$

Lemma 6
Let $F = m + G$ be an affine space of direction G and f the function such that:

$$\begin{aligned} f: \ \chi &\to \mathbb{R} \\ x &\mapsto d_{\mathcal{X}}(x,F) = \|x - \pi_F(x)\|. \end{aligned}$$

Then:

$$D_x f(y) = \frac{\langle x - \pi_F(x), y - \pi_G(y)\rangle}{\|x - \pi_F(x)\|},$$

where $D_x f(y)$ denotes the derivative of f taken at point $x \in \mathcal{X}$ and applied to $y \in \mathcal{X}$.

Proof
Let us write $f = g \circ h$ with:

$$\begin{aligned} g: \ \chi &\to \mathbb{R} \\ x &\mapsto \|x\| \\ h: \ \chi &\to F^{\perp} \\ x &\mapsto x - \pi_F(x) \end{aligned}$$

We have:

$$D_x f(y) = \left(D_{g(x)} \circ D_x h\right)(y).$$

We compute each of the derivative $D_x g$ and $D_x h$. The derivative of the composite function f is given by:

$$D_x g(y) = \frac{\langle x, y\rangle}{\|x\|}.$$

$$\begin{aligned} &h(x) = x - \pi_F(x) \underset{\text{Lemma 2}}{=} x - (m + \pi_G(x-m)) = x - m + \pi_G(m) - \pi_G(x), \\ &\text{and thus } D_x h(y) = y - \pi_G(y). \end{aligned}$$

By combining the two derivatives, it holds:

$$D_x f(y) = \frac{\langle x - \pi_F(x), y - \pi_G(y)\rangle}{\|x - \pi_F(x)\|}. \quad \blacksquare$$

The main argument of the proof of proposition 17 relies on Rademacher's theorem that we recall for the sake of completeness.

Proposition 26 (Rademacher's theorem),[45]
Let U be an open subset of $\mathbb{R}^d$ for $d \in \mathbb{N}$ and $f: U \to \mathbb{R}$ a Lipschitz function. Then f is almost everywhere differentiable.

Proof of proposition 17
By lemma 5, we know that $d_{\mathcal{X}}(\cdot, S)$ is 2-Lipschitz whenever S is composed of k affine spaces ($S = \bigcup_{1\leq j\leq k} F_j$). In this case, by Rademacher's theorem, we know that $d_{\mathcal{X}}(\cdot, S)$ is almost everywhere differentiable. By lemma 6, we see that $d_{\mathcal{X}}(\cdot, S)$ is not differentiable on the $(F_j)_{1\leq j\leq k}$ and on the intersections of extended Voronoi regions $\bar{\mathcal{S}}_j^{(f)} \cap \bar{\mathcal{S}}_{j'}^{(f)}$ for $j \neq j' \in [\![1, k]\!]$. Therefore the set J such that

$$J = \left(\bigcup_{1\leq j\leq k} F_j\right) \bigcup \left(\bigcup_{1\leq j\leq k} \bigcup_{1\leq j'<j} \bar{\mathcal{S}}_j^{(f)} \cap \bar{\mathcal{S}}_{j'}^{(f)}\right),$$

has zero measure and so has $\bigcup_{1\leq j\leq k} \bigcup_{1\leq j'<j} \bar{\mathcal{S}}_j^{(f)} \cap \bar{\mathcal{S}}_{j'}^{(f)}$. ∎

B.6 Section 4.2

In all the proofs we use the following inequality that is a direct consequence of proposition 19.

$$\mathbb{P}\left(\mathcal{E}(\hat{S}_k) > \hat{\mathcal{E}}(\hat{S}_k) + \sqrt{18\pi}\frac{k}{\sqrt{n}} + \epsilon\right) \leq \mathbb{P}\left(\sup_{S\in\mathcal{P}_k} |\mathcal{E}(S) - \hat{\mathcal{E}}(S)| > \sqrt{18\pi}\frac{k}{\sqrt{n}} + \epsilon\right) \leq e^{-\frac{\epsilon^2 n}{8}}. \quad (25)$$

Proof of proposition 21

$$\begin{aligned}
\mathbb{P}\left(\mathcal{E}(\tilde{S}) - \tilde{\mathcal{E}}(\tilde{S}) > \epsilon\right) &\leq \mathbb{P}\left(\max_{1\leq k\leq n}\left(\mathcal{E}(\hat{S}_k) - \tilde{\mathcal{E}}(\hat{S}_k)\right) > \epsilon\right) \\
&\underset{\text{Union bound}}{\leq} \sum_{k=1}^{n} \mathbb{P}\left(\mathcal{E}(\hat{S}_k) - \tilde{\mathcal{E}}(\hat{S}_k) > \epsilon\right) \\
&= \sum_{k=1}^{n} \mathbb{P}\left(\mathcal{E}(\hat{S}_k) - \hat{\mathcal{E}}(\hat{S}_k) > \epsilon + 4\sqrt{\frac{\ln k}{n}} + \sqrt{18\pi}\frac{k}{\sqrt{n}}\right) \\
&\underset{\text{Equation (25)}}{\leq} \sum_{k=1}^{n} e^{-\frac{n}{8}\left(\epsilon + 4\sqrt{\frac{\ln k}{n}}\right)^2} \\
&\leq \sum_{k=1}^{n} e^{-\frac{n}{8}(\epsilon^2 + 16\frac{\ln k}{n})} \\
&\leq e^{-\frac{n\epsilon^2}{8}} \sum_{k=1}^{n} \frac{1}{k^2} \\
&\leq e^{-\frac{n\epsilon^2}{8}} \sum_{k=1}^{\infty} \frac{1}{k^2} \\
&= \frac{\pi^2}{6} e^{-\frac{\epsilon^2 n}{8}}. \quad ∎
\end{aligned}$$

Proof of proposition 22

$$\mathbb{E}\left(\mathcal{E}(\tilde{S}) - \mathcal{E}(S_k)\right) = \underbrace{\mathbb{E}\left(\mathcal{E}(\tilde{S}) - \tilde{\mathcal{E}}(\tilde{S})\right)}_{\Gamma_1} + \underbrace{\mathbb{E}\left(\tilde{\mathcal{E}}(\tilde{S}) - \mathcal{E}(S_k)\right)}_{\Gamma_2}$$

$$
\begin{array}{rcl}
\Gamma_1^2 & \underset{\text{Cauchy Schwarz}}{\leq} & \mathbb{E}\left(\left(\mathcal{E}(\tilde{S})-\tilde{\mathcal{E}}(\tilde{S})\right)^2\right) \\
& = & \displaystyle\int_0^\infty \mathbb{P}\left(\left(\mathcal{E}(\tilde{S})-\tilde{\mathcal{E}}(\tilde{S})\right)^2 > t\right) dt \\
& \underset{\mathcal{E}(\tilde{S})-\tilde{\mathcal{E}}(\tilde{S})\geq 0}{=} & \displaystyle\int_0^\infty \mathbb{P}\left(\mathcal{E}(\tilde{S})-\tilde{\mathcal{E}}(\tilde{S}) > \sqrt{t}\right) dt \\
& \leq & \displaystyle\int_0^u \mathbb{P}\left(\mathcal{E}(\tilde{S})-\tilde{\mathcal{E}}(\tilde{S}) > \sqrt{t}\right) dt + \int_u^\infty \mathbb{P}\left(\mathcal{E}(\tilde{S})-\tilde{\mathcal{E}}(\tilde{S}) > \sqrt{t}\right) dt \\
& \underset{\text{Proposition 21}}{\leq} & \displaystyle u + \int_u^\infty e^{-\frac{nt}{8}} \\
& \leq & \displaystyle u + \frac{8\pi^2}{6n} e^{-\frac{nu}{8}} \\
\Gamma_1^2 & \leq & \displaystyle \ln\left(\frac{e\pi^2}{6}\right)\frac{8}{n} \\
\Gamma_2 & \underset{\tilde{\mathcal{E}}(\tilde{S})\leq\tilde{\mathcal{E}}(\hat{S}_k)}{\leq} & \mathbb{E}\left(\tilde{\mathcal{E}}(\hat{S}_k)-\mathcal{E}(S_k)\right) \\
& = & \mathbb{E}\left(\hat{\mathcal{E}}(\hat{S}_k)+p(k,n)-\mathcal{E}(S_k)\right) \\
& \underset{\hat{\mathcal{E}}(\hat{S}_k)\leq\hat{\mathcal{E}}(S_k)}{\leq} & \mathbb{E}\left(\hat{\mathcal{E}}(S_k)+p(k,n)\right)-\mathcal{E}(S_k) \\
\Gamma_2 & \underset{\mathbb{E}\left(\hat{\mathcal{E}}(S_k)\right)=\mathcal{E}(S_k)}{=} & \mathbb{E}\left(p(k,n)\right)
\end{array}
$$

All in all we get:

$$
\begin{array}{rcl}
\mathbb{E}\left(\mathcal{E}(\tilde{S})-\mathcal{E}(S^*)\right) & = & \mathbb{E}\left(\mathcal{E}(\tilde{S})-\mathcal{E}(S_k)\right)+\mathcal{E}(S_k)-\mathcal{E}(S^*) \\
& \leq & \mathcal{E}(S_k)-\mathcal{E}(S^*)+\mathbb{E}\left(p(k,n)\right)+\sqrt{\ln\left(\frac{e\pi^2}{6}\right)\frac{8}{n}} \quad \forall k\in[\![1,n]\!] \\
& \leq & \min_{1\leq k\leq n}\left(p(k,n)+\inf_{S\in\mathcal{P}_k}\left(\mathcal{E}(S)-\mathcal{E}(S^*)\right)\right)+\sqrt{\ln\left(\frac{e\pi^2}{6}\right)\frac{8}{n}}. \quad \blacksquare
\end{array}
$$

Proof of propostion 23

$$
\begin{array}{rcl}
\mathbb{P}\left(\mathcal{E}(\tilde{S}) > \min_{1\leq k\leq n}\left(\mathcal{E}(S_k)+p(k,n)+4\sqrt{\frac{\ln k}{n}}\right)+\epsilon\right) & \leq & \underbrace{\mathbb{P}\left(\mathcal{E}(\tilde{S}) > \tilde{\mathcal{E}}(\tilde{S})+\frac{\epsilon}{2}\right)}_{\Delta_1} \\
& + & \underbrace{\mathbb{P}\left(\tilde{\mathcal{E}}(\tilde{S}) > \min_{1\leq k\leq n}\left(\mathcal{E}(\tilde{S})+p(k,n)+4\sqrt{\frac{\ln k}{n}}\right)+\frac{\epsilon}{2}\right)}_{\Delta_2}.
\end{array}
$$

$$
\begin{aligned}
\Delta_1 &\underset{\text{Proposition 21}}{\leq} \frac{\pi^2}{6} e^{-\frac{n}{8}(\frac{\epsilon}{2})^2} = \frac{\pi^2}{6} e^{-\frac{n\epsilon^2}{32}}. \\
\Delta_2 &\leq \mathbb{P}\left(\max_{1\leq k\leq n} \left(\tilde{\mathcal{E}}(\hat{S}_k) - \mathcal{E}(S_k) - p(k,n) - 4\sqrt{\frac{\ln k}{n}} \right) > \frac{\epsilon}{2} \right) \\
&\underset{\text{Union bound}}{\leq} \sum_{k=1}^{n} \mathbb{P}\left(\tilde{\mathcal{E}}(\hat{S}_k) - \mathcal{E}(S_k) - p(k,n) - 4\sqrt{\frac{\ln k}{n}} > \frac{\epsilon}{2} \right) \\
&\leq \sum_{k=1}^{n} \mathbb{P}\left(\hat{\mathcal{E}}(\hat{S}_k) - \mathcal{E}(S_k) > \frac{\epsilon}{2} + 4\sqrt{\frac{\ln k}{n}} \right) \\
&\leq \sum_{k=1}^{n} \mathbb{P}\left(\hat{\mathcal{E}}(S_k) - \mathcal{E}(S_k) > \frac{\epsilon}{2} + 4\sqrt{\frac{\ln k}{n}} \right).
\end{aligned}
$$

We have:

$$
\begin{aligned}
\mathbb{E}\left(\hat{\mathcal{E}}(S_k)\right) &= \mathcal{E}(S_k). \\
\forall i \in [\![1,n]\!],\ \mathbb{P}\left(d_{\mathcal{X}}(x_i, S_k) \leq 2\right) &= 1.
\end{aligned}
$$

From the second equation, we deduce, that $\mathbb{P}\left(0 \leq \hat{\mathcal{E}}(S_k) \leq 4\right) = 1$ and thus we can apply Hoeffding's inequality:

$$
\mathbb{P}\left(\hat{\mathcal{E}}(S_k) - \mathcal{E}(S_k) > \frac{\epsilon}{2} + 4\sqrt{\frac{\ln k}{n}} \right) \leq e^{-\frac{2n\left(\frac{\epsilon}{2} + 4\sqrt{\frac{\ln k}{n}}\right)^2}{16}}.
$$

$$
\begin{aligned}
\Delta_2 &\underset{\text{Hoeffding's inequality}}{\leq} \sum_{k=1}^{n} e^{-\frac{2n\left(\frac{\epsilon}{2} + 4\sqrt{\frac{\ln k}{n}}\right)^2}{16}} \\
&\leq e^{-\frac{n\epsilon^2}{32}} \sum_{k=1}^{n} \frac{1}{k^2} \\
&\leq \frac{\pi^2}{6} e^{-\frac{n\epsilon^2}{32}}.
\end{aligned}
$$

All in all, we get that:

$$
\begin{aligned}
\mathbb{P}\left(\mathcal{E}(\tilde{S}) > \min_{1\leq k\leq n} \left(\mathcal{E}(S_k) + p(k,n) + 4\sqrt{\frac{\ln k}{n}} \right) + \epsilon \right) &\leq \frac{\pi^2}{6}\left(e^{-\frac{n\epsilon^2}{32}} + e^{-\frac{n\epsilon^2}{8}} \right) \\
&\leq \frac{\pi^2}{3} e^{-\frac{n\epsilon^2}{32}}. \quad \blacksquare
\end{aligned}
$$

C Code

C.1 Equation of the trefoil

The trefoil is a $3d$ curve (see figure 11). A point x on the trefoil is defined as follows:

$$
\begin{aligned}
&t \sim 2\pi\, \mathcal{U}([0,1]), \quad \omega(t) = 2 + \cos(3t), \\
&x = \omega(t)\cos(2t), \\
&y = \omega(t)\sin(2t), \\
&z = \sin(3t).
\end{aligned}
$$

where $\mathcal{U}([0,1])$ is a random variable drawn uniformly on $[0,1]$.

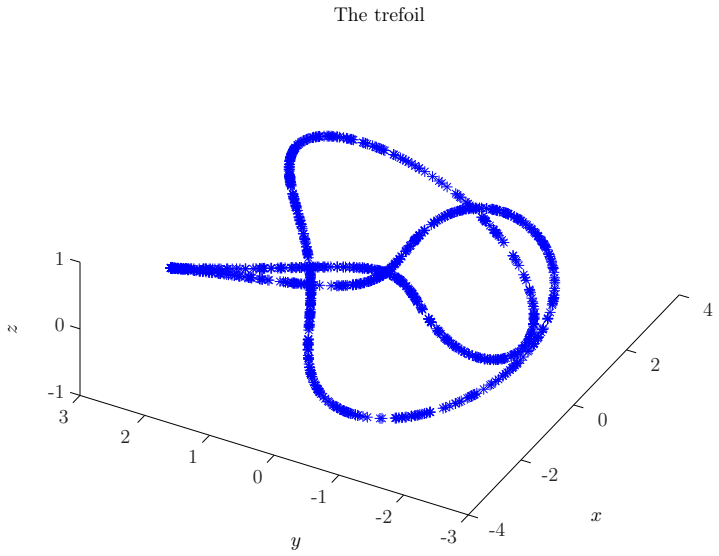
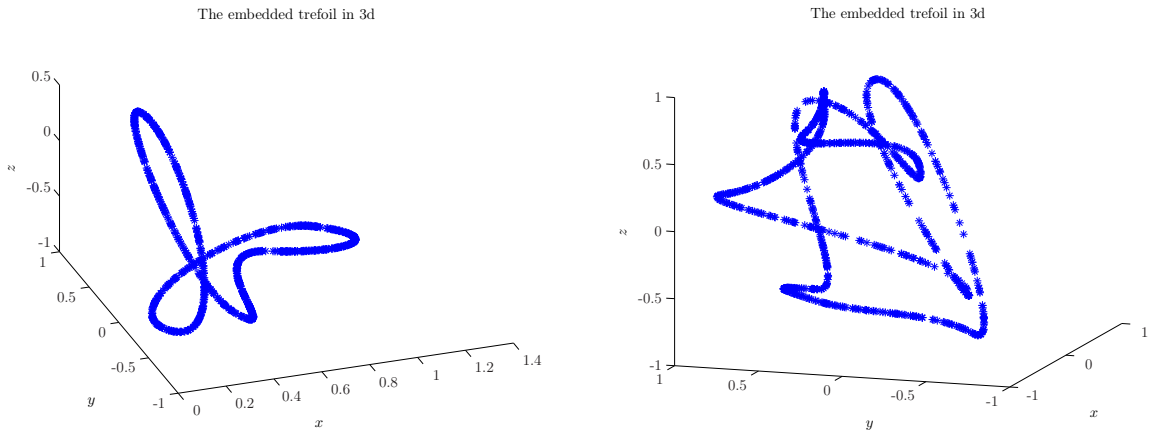


Figure 11: The trefoil.

C.2 Equation of the embedded trefoil

Given two integers d and N, a point x on the embedded trefoil (see figure 12) is defined as:

$$
\begin{aligned}
&s \sim \mathcal{U}([\![1, N]\!]), \quad s' = s \bmod(N) + 1, \quad t \sim \mathcal{N}(0,1),\\
&p_1, \dots, p_N \in \mathcal{S}(0,1)^d,\\
&v_1, \dots, v_N \in \mathcal{S}(0,1)^d,\\
&x_1 = 3(\frac{1}{3}v_s + p_s)(1-t)^2 t,\\
&x_2 = 3(-\frac{1}{3}v_{s'} + p_{s'})(1-t)t^2,\\
&x_0 = p_s(1-t)^3,\\
&x_3 = p_{s'}t^3,\\
&x = x_1 + x_2 + x_3 + x_4.
\end{aligned}
$$



(a) N=5 (b) N=10

Figure 12: The embedded trefoil for N=5 and N=10.

Notation

- $\mathcal{X}$: separable Hilbert space with inner product $\langle ., . \rangle$ and norm $\|.\|$
- H: feature space (d'-dimensional Hilbert space) with inner product $\langle ., . \rangle_H$ and norm $\|.\|_H$ ($d' \in [\![1, \infty]\!]$) (see definition 2)
- $\mathcal{M}$: manifold on which the data lies
- $\mathcal{H}$: hypothesis space (see page 19)
- X_n: training set
- $\mathcal{C}_j$: indices of points in the training set belonging to cluster j (see equation (6))
- C: assignment vector (from a data point to its closest mean or flat, depending on the algorithm) (see equation (5))
- n_j: size of the j-th cluster (see equation (7))
- $\overline{x}$: center of mass of the training set $\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$
- $\overline{x}_j$: center of mass of the j-th group of points $\overline{x}_j = \frac{1}{n_j} \sum_{i \in \mathcal{C}_j} x_i$
- Φ_n: image of the training set, *i.e.* $\Phi_n = (\phi(x_1), \ldots, \phi(x_n))$
- $\overline{\Phi}_n$: center of mass of the image of the training set $\overline{\Phi}_n = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i)$
- $\overline{\Phi}_j$: center of mass of the j-th group of points in the feature space $\overline{\Phi}_j = \frac{1}{n_j} \sum_{i \in \mathcal{C}_j} \phi(x_i)$
- $\hat{S}$: set obtained by minimizing the empirical reconstruction error on $\mathcal{H}$ (see equation (2))
- π_F: projection on a closed set F (see equation (4))
- $\mathbb{R}^{p \times q}$: matrices with p lines and q columns
- D_{pq}: diagonal matrices with p lines and q columns
- O_{pq}: orthogonal matrices with p lines and q columns
- $M^\top$: (conjugate) transpose matrix of M
- $X \sim \mu$: X is distributed according to μ
- $\mathcal{E}$: reconstruction error (see equation (13))
- $\hat{\mathcal{E}}$: empirical reconstruction error (see equation (1))
- $\tilde{\mathcal{E}}$: penalized reconstruction error (see equation (23))
- $\lambda_{\max}(G)$: maximal eigenvalue of G
- $\lambda_{\min}(G)$: minimal eigenvalue of G
- $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$
- K: kernel matrix
- $Z = X_n X_n^\top$ and Z_j the same matrix restricted to the j-th cluster (see section 1.2.2)
- $G = X_n^\top X_n$: Gram matrix of X_n
- $V_{k,r}(p)$: k-th quantization error of order r (see equation (19))

- μ_I: measure of volume on the manifold (see page 25)
- $\mathcal{S}(0,1)^d$: unit sphere in $\mathbb{R}^d$
- $\mathcal{B}(0,1)^d$: unit ball in $\mathbb{R}^d$
- $(a)_p$: p-th component of the vector a
- $\mu_{\mathcal{X}}$: Lebesgue measure on $\mathcal{X}$
- For a point y and a vector space F, $F_y = y + F = \{y + f, f \in F\}$
- $\mathcal{Q}_k$: set of k-point quantizers (see page 24)

References

[1] H B Barlow. Unsupervised learning: introduction. In Geoffrey E. Hinton and Terrence Joseph Sejnowski, editors, *Unsupervised Learning: Foundations of Neural Computation*, pages 1–17. Bradford Company Scituate, MA, USA, 1999.

[2] Aarti Singh, Robert Nowak, and Xiaojin Zhu. Unlabeled data: Now it helps, now it doesn't. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1513–1520. 2009.

[3] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 161–168, New York, NY, USA, 2006. ACM.

[4] Rich Caruana, Nikos Karampatziakis, and Ainur Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 96–103, New York, NY, USA, 2008. ACM.

[5] M Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

[6] Xi Zhou, Kai Yu, Tong Zhang, and Thomas S. Huang. Image classification using supervector coding of local image descriptors. In *Proceedings of the 11th European conference on Computer vision: Part V*, ECCV'10, pages 141–154, Berlin, Heidelberg, 2010. Springer-Verlag.

[7] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 609–616, New York, NY, USA, 2009. ACM.

[8] Sue Inn Ch', ñg, Kah Phooi Seng, and Li-Minn Ang. Block-based deep belief networks for face recognition. *Int. J. Biometrics*, 4(2):130–143, April 2012.

[9] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[10] Jerome H. Friedman. On bias, variance, 0/1loss, and the curse-of-dimensionality. *Data Min. Knowl. Discov.*, 1(1):55–77, January 1997.

[11] M. Balasubramanian, E. L. Shwartz, J. B. Tenenbaum, V. de Silva, and J. C. Langford. The isomap algorithm and topological stability. *Science*, 2002.

[12] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, December 2000.

[13] David L. Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences of the United States of America*, 100(10):5591–5596, May 2003.

[14] K.Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-04)*, volume 2, pages 988–995, Washington D.C., 2004.

[15] Matthew Brand. Charting a manifold. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *NIPS*, pages 961–968. MIT Press, 2002.

[16] Ronald R. Coifman, Ioannis G. Kevrekidis, Stéphane Lafon, Mauro Maggioni, and Boaz Nadler. Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems. *Multiscale Modeling & Simulation*, 7(2):842–864, 2008.

[17] Guillermo D. Canas and Lorenzo Rosasco. Learning manifolds with K-means and K-flats. Preprint, 2012.

[18] Andreas Maurer and Massimiliano Pontil. K-dimensional coding schemes in hilbert spaces. *IEEE Trans. Inf. Theor.*, 56(11):5839–5846, November 2010.

[19] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 551–556, New York, NY, USA, 2004. ACM.

[20] Gerard Biau, Luc Devroye, and Gábor Lugosi. On the performance of clustering in hilbert spaces. *IEEE Transactions on Information Theory*, 54(2):781–790, 2008.

[21] P Bartlett, T Linder, and G Lugosi. *The minimax distortion redundancy in empirical quantizer design*, volume 44, pages 1802–1813. 1997.

[22] Peter L. Bartlett, Stéphane Boucheron, and Gábor Lugosi. Model selection and error estimation. *Mach. Learn.*, 48(1-3):85–113, September 2002.

[23] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar k-means problem is np-hard. In *Proceedings of the 3rd International Workshop on Algorithms and Computation*, WALCOM '09, pages 274–285, Berlin, Heidelberg, 2009. Springer-Verlag.

[24] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[25] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. pages 225–232. ACM Press, 2004.

[26] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129 – 137, mar 1982.

[27] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. NP-hardness of euclidean sum-of-squares clustering. *Machine Learning*, 75:245–248, 2009. 10.1007/s10994-009-5103-0.

[28] Charles Elkan. Using the triangle inequality to accelerate k-means. In *ICML'03*, pages 147–153, 2003.

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

[30] P. Tseng. Nearest q-flat to mpoints. *J. Optim. Theory Appl.*, 105(1):249–252, April 2000.

[31] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[32] Jaehwan Kim, Kwang hyun Shim, and Seungjin Choi. Soft geodesic kernel k-means. In *In IEEE International Conference on Acoustics, Speech and Signal*, pages 429–432, 2007.

[33] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, July 1998.

[34] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Advances in kernel methods - Support Vector Learning*, pages 327–352. MIT Press, 1999.

[35] Matthias Hein and Markus Maier. Manifold denoising. In *Advances in Neural Information Processing Systems (NIPS) 19*. MIT Press. 5, 2006.

[36] Siegfried Graf and Harald Luschgy. *Foundations of quantization for probability distributions*. Lecture notes in mathematics. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2000.

[37] Peter M. Gruber and Peter M. Gruber. Optimum quantization and its applications. *Adv. Math*, 186:2004, 2002.

[38] Allen Gersho and Robert M. Gray. *Vector quantization and signal compression*. Kluwer Academic Publishers, Norwell, MA, USA, 1991.

[39] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[40] M. Ledoux. *The Concentration of Measure Phenomenon*. Mathematical Surveys and Monographs. American Mathematical Society, 2001.

[41] Comparison of learning algorithms for handwritten digit recognition. In F. Fogelman and P. Gallinari, editors, *International Conference on Artificial Neural Networks*, pages 53–60, 1995.

[42] J. M. Peña, J. A. Lozano, and P. Larrañaga. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recogn. Lett.*, 20(10):1027–1040, October 1999.

[43] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):881–892, July 2002.

[44] Michael Syskind Pedersen Kaare Brandt Petersen. *The Matrix Cookbook*. 2008.

[45] Martin Muñoz. Rademacher's theorem.