Report on "Compressive K-means"

Gaetano Agazzotti
gaetano.agazzotti@ens-paris-saclay.fr

Victor Jesequel
victor.jesequel@student-cs.fr

Cyrielle Théobald
cyrielle.theobald@student-cs.fr

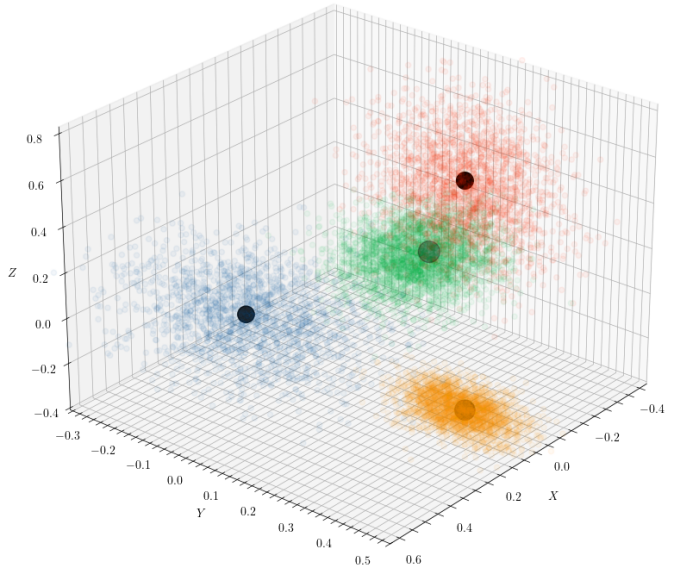

Figure 1: Compressive K-means (3D clustering illustration of our implementation)

ABSTRACT

In this report, we present the compressive K-means method introduced by Keriven *et al.*. This method enables clustering to be carried out in a very short time, overcoming the limitations of Lloyd's algorithm when the number of data items or their dimensions become significant. In addition to its speed, this method has other advantages which we present in this document. However, this method was published without theoretical guarantees regarding its convergence. Using recent literature on compressive learning, we state the conditions under which this method is effective. Finally, we test numerically using our Python implementation the conditions of the convergence theorems and show that some are tight and others could be improved.

1 INTRODUCTION

In machine learning, clustering is one the main challenge on which an extensive literature has been produced and is still at stake today. Among the plethoric number of clustering procedure, we can mention the Support Vector Machine introduced in [1], the dendrogram method developed in [7] and the DBSCAN procedure exposed in [3]. But the most famous one is undeniably the K-means algorithm of [8] that has become a standard in this field. It consists in finding barycenters (also called centroids) that minimize the distances between the data points and their nearest centroid (barycenter). More formally, we consider:

- a dataset $\mathcal{X} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\} \in \mathbb{R}^{d\times n}$ of $d \in \mathbb{N}^*$ dimensional $n \in \mathbb{N}^*$ observations,
- an integer K denoting the number of clusters in which we want to classify our data.

The aim of the K-means algorithm is to find K centroids $\boldsymbol{C} = \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_K\} \in \mathbb{R}^{d\times K}$ that minimize the cost:

$$\mathrm{SSE}(\mathcal{X}, \boldsymbol{C}) = \sum_{i=1}^{n} \min_{k\in[\![1,K]\!]} \|\boldsymbol{x}^{(i)} - \boldsymbol{c}_k\|_2^2. \tag{1}$$

An iteration of the K-means algorithm has a time complexity of $O(ndK)$ and therefore, as soon as one factor becomes too large, the procedure becomes intractable from a computational time perspective.

In order to overcome this major issue, the authors reduce the dimension of the minimization problem (1) by performing a sketching procedure $\mathrm{Sk}(\cdot,\cdot)$ (detailed in section 2) on the dataset $\mathcal{X}$. It yields studying the problem:

$$\begin{aligned} &\underset{\boldsymbol{C},\boldsymbol{\alpha}}{\operatorname{argmin}} \ \|\mathrm{Sk}(\mathcal{X}, \mathbf{1}_n/n) - \mathrm{Sk}(\boldsymbol{C}, \boldsymbol{\alpha})\|_2^2 \\ &\text{s.t. } \boldsymbol{\alpha} \in \mathbb{S}_K \end{aligned} \tag{2}$$

instead of solving (1) to get the K centroids $\boldsymbol{C} = \{\boldsymbol{c}^{(1)}, ..., \boldsymbol{c}^{(K)}\}$ where $\mathbf{1}_n$ is the unit vector $(1, ..., 1)$ of length n and $\mathbb{S}_K$ is the K-dimensional probability simplex, *i.e.* $\mathbb{S}_K := \{\alpha \in \mathbb{R}_+^K | \ \langle \boldsymbol{\alpha}, \mathbf{1}_K\rangle = 1\}$.

In this report, we first focus on a theoretical point of view. More particularly, we focus our attention on the sketching procedure as we believe that it provides very interesting insights into subtile approaches to machine learning problems (the algorithm deals more with optimization). What is the sense of problem 2 ? Why is

this sketching procedure natural ? Can we get some convergence guarantees, and under which constraints ? We will see that five years after the compressive K-means methods was introduced, some guarantees were exposed in [6]. Answering these questions is the aim of section 2. In section 3, we test and numerically stress the assumptions under which these guarantees are provided. We will see that some of them are "tight" and other can be sharpened.

2 DEEP DIVE INTO THE THEORETICAL BACKSTAGE OF COMPRESSIVE LEARNING

Important machine and deep learning research is driven by the following question: how can we estimate the underlying distribution from which the observations have been drawn ? The huge amount of data now available allows us to address this question. However, the memory capacity required to handle these datasets is often a practical limitation since it is now common to have datasets in the Tb range. Of course, this makes running algorithms on a "reasonable" computer almost impossible. To this end, significant research works have been dedicated to reduce the dimensionality of datasets before applying any learning step. More precisely, there are essentially three ways of achieving such a task: *subsampling*, *dimensionality reduction* and *sketching*.

2.1 Alternative methods for size reduction: *subsampling* and *dimensionality reduction*

Subsampling simply consists in taking as learning sets a subset of the initial dataset $\mathcal{X}$, *i.e.*, finding $\mathcal{X}_{\text{sub}} \subset \mathcal{X}$ such that $\#\mathcal{X}_{\text{sub}} =: n_{\text{sub}} \ll n$. Clever ways of finding such datasets are for example exposed in [4] and [11]. However, if the dimension d is very large, the *subsampling* step does not effectively tackle the issue.

Dimensionality reduction, on the other hand, reduces the dimension of the observations. More precisely, it consists in finding a set $\mathcal{X}_{\text{red}} := \{A\boldsymbol{x} | \ \boldsymbol{x} \in \mathcal{X}\}$ where $A : \mathbb{R}^d \to \mathbb{R}^{d_{\text{red}}}$ with $d_{\text{red}} \ll d$. This dimension reduction aims to maintain the most important information. This task can be done under random projection since the *Johnson-Lidenstrauss lemma* (see [2] and [10]) guarantees the distance preservation in $\mathbb{R}^{d_{\text{red}}}$ up to some precision depending on d_{red} (A would be a random matrix in this case). Despite its great theoretical interest, the problem of this technique is that it still requires working with n observations.

In figure 2, we illustrate the two methods described above.

2.2 Introduction to sketching: interlude with the one-dimensional moment method

To avoid the problems that might arise with the two previous techniques, the authors of [9], inspired by the work in [5], introduced a sketching step prior to any learning. This step consists in representing the data as a one-dimensional vector of small size (compared to the largest dimension of the problem).

To illustrate this method, let us consider the estimation problem 1.

Problem 1. *Let us assume that we have $\mathcal{X}$ a collection of n i.i.d. observations of an unknown probability π. We want to find an approximation $\pi^{\mathscr{C}}\left(\widehat{\theta}\right)$ of π belonging to some parametric distribution space $\mathscr{C}$ by estimating its parameters $\widehat{\theta}$. How can we determine $\widehat{\theta}$ from the n observations ?*

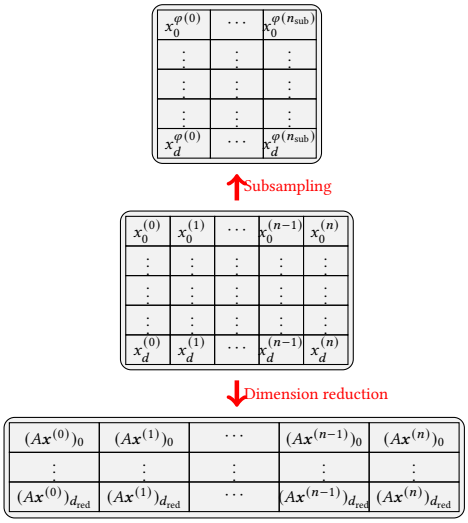

Figure 2: Illustration of subsampling and dimension reduction

Designing a sketching procedure can be seen as feature engineering. The main idea of this approach is to reduce the dimensionality without losing significant information. To simplify the ideas, let us focus on the one-dimensional case, *i.e.*, the data set is composed $\mathcal{X}$ of n real numbers.

To recover the density π from which the data has been drawn, the first information that one would like to compute is the mean of $\mathcal{X}$ denoted $\mathfrak{m}_{\mathcal{X}}^{(1)}$, which only gives a piece of information and clearly cannot fully describe a complex structure. When the model leaves the naive world of normal distributions, a larger number of descriptors is needed. To enrich the description of the dataset, a natural idea is to compute the first m moments where $m \in \mathbb{N}^*$ is chosen to be small compared to n. In other words, one could approximately describe the dataset thanks to the vector $\mathfrak{m}_{\mathcal{X}} = \left(\mathfrak{m}_{\mathcal{X}}^{(k)}\right)_{k \in [\![1,m]\!]}$ where for all $k \geq 0$:

$$\mathfrak{m}_{\mathcal{X}}^{(k)} = \frac{1}{n} \sum_{\ell=1}^{n} \left(\boldsymbol{x}^{(\ell)}\right)^k. \tag{3}$$

Having a great number of observations n and computing enough empirical moments m provides an alternative (but equivalent) representation of the probability distribution. This shows that a sketching procedure is completely justified from a theoretical point of view. Note that, for $m \in \mathbb{N}^*$, if we define $\Phi_{(m)}$ as the function:

$$\forall \boldsymbol{x} \in \mathbb{R}, \quad \Phi_{(m)}(x) = \begin{pmatrix} x^1 \\ \vdots \\ x^m \end{pmatrix}, \tag{4}$$

using this notation, the moment vector can be written as

$$\mathfrak{m}_{\mathcal{X}} = \frac{1}{n} \sum_{i=1}^{n} \Phi_{(m)}\left(\boldsymbol{x}^{(i)}\right) \tag{5}$$

We see that this quantity is simply the empirical expectation

$$\mathfrak{m}_{\mathcal{X}} = \mathbb{E}_{X\sim\widehat{\pi}_n}[\Phi_{(m)}(X)] \tag{6}$$

where $\widehat{\pi}_n$ stands for the empirical discrete distribution, that is:

$$\widehat{\pi}_n = \frac{1}{n}\sum_{i=1}^{n}\delta_{\boldsymbol{x}^{(i)}}. \tag{7}$$

In other words, we have compressed the information of the dataset with an empirical expectation, which is a linear operator w.r.t. the underlying distribution. The minimization problem to solve is then:

$$\underset{\theta}{\operatorname{argmin}} \|\mathbb{E}_{X\sim\pi_{\mathscr{C}}(\theta)}[\Phi_{(m)}(X)] - \mathbb{E}_{X\sim\widehat{\pi}_n}[\Phi_{(m)}(X)]\|_2^2. \tag{8}$$

2.3 Application to K-means

We will see here how this method applies to clustering. If we want to cluster, it seems natural to think that our data are spatially localized. In other words, we want to find a distribution easy to represent that best represents the underlying distribution of the observations π. The natural parametric distribution space to introduce is the K-Dirac mixture space defined as all the discrete measures that can be written as:

$$\sum_{i=1}^{K}\boldsymbol{\alpha}_i\delta_{\boldsymbol{c}_i} \tag{9}$$

where $\boldsymbol{\alpha}\in\mathbb{S}_K$ and $\boldsymbol{c}\in\mathbb{R}^{d\times K}$. The objective of clustering is the following: from the observations $\boldsymbol{x}$, how can we recover the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{c}$?

We have seen in the one-dimensional case that our goal is to find a reduced vector, that contains a sufficient amount of information, and is linear w.r.t. the underlying density of the data. We have previously established that a good candidate for this sketch vector , would be $\mathbb{E}_{X\sim\widehat{\pi}_n}[\Phi(X)]$ and the recovering problem would be:

$$\begin{aligned}\underset{\pi_K}{\operatorname{argmin}}\ &\|\mathbb{E}_{X\sim\widehat{\pi}_n}[\Phi(X)] - \mathbb{E}_{X\sim\pi_K}[\Phi_{(m)}(X)]\|_2^2\\ \text{s.t. }&\pi_K\ K\text{-Dirac measure}\end{aligned}. \tag{10}$$

This problem is then equivalent to its parametrical form:

$$\begin{aligned}\underset{\alpha\in\mathbb{S}_K,\boldsymbol{c}\in\mathbb{R}^{d\times K}}{\operatorname{argmin}}\ &\|\mathbb{E}_{X\sim\widehat{\pi}_n}[\Phi(X)] - \mathbb{E}_{X\sim\pi_K}[\Phi_{(m)}(X)]\|_2^2\\ \text{s.t. }&\pi_K = \sum_{i=1}^{K}\boldsymbol{\alpha}_i\delta_{\boldsymbol{c}_i}\end{aligned}. \tag{11}$$

To address this problem, we first aim at finding a function Φ that brings this problem into a m-dimensional minimization problem. More precisely, we are looking for an operator linear w.r.t. the probability distribution, that transforms our matrix observation in a vector of dimension m. It is well known that the Discrete Fourier Transform (DFT) satisfies these criterion. In this case, the sketch function Φ would then be:

$$\forall\boldsymbol{x}\in\mathbb{R}^n,\quad \Phi(\boldsymbol{x}) = \begin{pmatrix} e^{-\mathrm{i}\langle\omega_1,\boldsymbol{x}\rangle}\\ \vdots\\ e^{-\mathrm{i}\langle\omega_m,\boldsymbol{x}\rangle}\end{pmatrix} \tag{12}$$

where $(w_1,\dots w_m)$ are some frequencies. The sketch vector becomes:

$$\widehat{\boldsymbol{z}} = \begin{pmatrix} \frac{1}{n}\sum_{i=1}^{n} e^{-\mathrm{i}\langle\omega_1,\boldsymbol{x}^{(i)}\rangle}\\ \vdots\\ \frac{1}{n}\sum_{i=1}^{n} e^{-\mathrm{i}\langle\omega_m,\boldsymbol{x}^{(i)}\rangle}\end{pmatrix} = \mathrm{DFT}(\mathcal{X})\left[(\omega_\ell)_{\ell\in[\![1,m]\!]}\right]\in\mathbb{R}^m. \tag{13}$$

Here, all the observations have equal weights since we considered the empirical distribution. This can be generalized by introducing the sketching operator associated with the frequencies $(\omega_1,\dots,\omega_m)$ and weights $\boldsymbol{\alpha}$ (non-negative and summing to 1), defined as:

$$\mathrm{Sk}(\boldsymbol{y},\boldsymbol{\alpha}) := \left(\sum_{i=1}^{\#\boldsymbol{y}}\alpha_i e^{-\mathrm{i}\langle\omega_\ell,\boldsymbol{y}^{(i)}\rangle}\right)_{\ell\in[\![1,m]\!]}. \tag{14}$$

It is now easy to see that $\mathrm{Sk}(\mathcal{X},\mathbf{1}_n/n) = \mathbb{E}_{X\sim\widehat{\pi}_n}[\Phi(X)]$ and it is clear that we are looking for a set of K centroids $\boldsymbol{C}$ and their associated weights $\boldsymbol{\alpha}$ that minimizes the difference between the two "DFT". More explicitly, it means that we reformulate the problem (1) as being:

$$\begin{aligned}\underset{\boldsymbol{C},\boldsymbol{\alpha}}{\operatorname{argmin}}\ &\|\mathrm{Sk}(\mathcal{X},\mathbf{1}_n/n) - \mathrm{Sk}(\boldsymbol{C},\boldsymbol{\alpha})\|_2^2\\ \text{s.t. }&\boldsymbol{\alpha}\in\mathbb{S}_K\end{aligned}. \tag{15}$$

With all this information, we can already highlight an important advantage of this method:

- the independence in the computation of the coordinates of the vector $\widehat{\boldsymbol{z}}$ enables one to leverage state-of-the-art methods in **distributed computation**,
- if one has to cluster stream data (meaning that the observations $\boldsymbol{x}^{(i)}$ arrives at time $t=i$), the update is easily achieved using the recursion relation:

$$\widehat{z}_{t+1} = \frac{1}{t+1}\left(t\widehat{z}_t + \begin{pmatrix} e^{-\mathrm{i}\langle w_1,\boldsymbol{x}^{(t+1)}\rangle}\\ \vdots\\ e^{-\mathrm{i}\langle w_m,\boldsymbol{x}^{(t+1)}\rangle}\end{pmatrix}\right). \tag{16}$$

  showing that the procedure is **streamable**,
- once the sketch vector $\widehat{\boldsymbol{z}}$ is computed, the dataset $\mathcal{X}$ can be discarded. The only information needed is the one contained in the compressed vector. This shows that the procedure is **privacy aware**.

Remark 1. *Is the learning procedure a deep learning one ? First, let us denote $\boldsymbol{\omega} := (\omega_1^\intercal,\dots,\omega_m^\intercal)\in\mathbb{R}_+^{n\times m}$ the matrix composed of the frequency vectors and $\boldsymbol{x} = (\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(n)})\in\mathbb{R}_+^{d\times n}$ the matrix of the observations. The previous computations showed that:*

$$\mathrm{Sk}(\mathcal{X},\mathbf{1}_n/n) = \mathrm{Average}_{\mathrm{cols.}}[\exp(-\mathrm{i}\boldsymbol{\omega}\boldsymbol{x})]\in\mathbb{R}^m \tag{17}$$

where the exponentiation is intended to be componentwize and the average linewise. The sketching procedure is then a composition of a linear and non-linear function succedeed by an average pooling layer. A direct link can be done with a single-layer network.

This procedure published in 2016 is known to produce very good results in practice. However, the choice of the frequencies and the sketch size m remained unclear. In 2020, theoretical guarantees and precise guidance for the choice of m and ω were given.

2.4 Theoretical framework and guarantees

In this subsection, we will prove how minimizing the sketch problem 2 is equivalent to minimizing the traditional K-means problem 1.

Let suppose that π is a probability distribution on $\mathbb{R}^d$ and that we have access to n samples of this distribution, denoted $\mathcal{X} = \{x_1, ..., x_n\}$. As said before, the aim of clustering is to find K centroids c minimizing the expectation of the distance between a draw of π and its nearest centroid. In other words, we want to minimize the risk:

$$\mathcal{R}(\pi, c) := \mathbb{E}_{X\sim\pi}\left[\min_{k\in[\![1,K]\!]} \|X - c_k\|_2^2\right]. \tag{18}$$

More precisely, if we take $c^* \in \operatorname{argmin}_c \mathcal{R}(\pi, c)$ and $\widehat{c}$ an estimate of the optimal centroids, the aim is to control the difference between what would the risk with optimal centroids and the estimated one. This quantity is the *excess risk* defined by:

$$\Delta_{c^*}\mathcal{R}(\pi, \widehat{c}) := \mathcal{R}(\pi, \widehat{c}) - \mathcal{R}(\pi, c^*)\ (\geq 0). \tag{19}$$

Let us now define the analog of the risk function in the reduced space. For this, let us consider a modification of the linear sketching operator considered in 2, defined as:

$$\mathcal{A}\pi := \mathbb{E}_{X\sim\pi}\left[\Phi_{\mathcal{A}}(X)\right] \tag{20}$$

with:

$$\Phi_{\mathcal{A}}(x) := \left(\frac{e^{-\mathrm{i}\langle x,\omega_i\rangle}}{\sqrt{m\left(1 + \frac{s\|\omega_i\|_2^2}{d}\right)}}\right)_{i\in[\![1,m]\!]} \tag{21}$$

where $s > 0$ is a scale parameter, m is the sketch size and $(\omega_i)_{i\in[\![1,m]\!]}$ are the frequencies (drawn from a special distribution as we will see later). Let us now define the sketching-risk:

$$\mathsf{R}_{\text{clust.}}(\widehat{\pi}_n, c) := \min_{\alpha\in\mathbb{S}_K} \|\mathcal{A}\widehat{\pi}_n - \mathcal{A}\pi_\alpha^c\|_2^2 \tag{22}$$

where π_α^c is the discrete distribution with locations c and weights α, *i.e.*, $\pi_\alpha^c = \sum_{i\in[\![1,K]\!]} \alpha_i \delta_{c_i}$ and $\widehat{\pi}_n$ denotes the empirical distribution defined as:

$$\widehat{\pi}_n = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}. \tag{23}$$

As said before, this $\mathsf{R}_{\text{clust.}}(\widehat{\pi}_n, c)$ represents how far the empirical distribution $\widehat{\pi}_n$ is from the (best) K-centroid distribution π_α^c in the $\mathcal{A}$-domain.

To go further, we need some additional objects to be defined. Firstly, for centroids $c \in \mathbb{R}^{d\times K}$, let us define the associated Voronoi cells $(V_i(c))_{i\in[\![1,K]\!]}$ by:

$$\forall i \in [\![1, K]\!],\ V_i(c) := \{x \in \mathbb{R}^d |\ \|x - c_i\|_2^2 = \min_{j\in[\![1,k]\!]} \|x - c_j\|_2^2\} \tag{24}$$

Then, let us take again the centroid minimizing the risk w.r.t. the underlying distribution π, *i.e.* $c^* \in \operatorname{argmin}_{c\in\mathbb{R}^{d\times K}} \mathcal{R}(\pi, c)$ and let us associate the distribution π^* obtained from associating to each draw of π, its nearest centroid of c^* (minimizing the K-means problem for the underlying distribution π). More formally, π^* is defined as:

$$\pi^* = \sum_{i=1}^K \pi(X \in V_i(c^*))\delta_{c_i^*}. \tag{25}$$

Additionally, for $R, \varepsilon \geq 0$ we denote the set of K centroids that are ε-separated and bounded by R by $\mathcal{H}_{K,2\varepsilon,R}$. Formally, we have:

$$\begin{aligned}\mathcal{H}_{K,2\varepsilon,R} := \{c \in \mathbb{R}^{d\times K} |\ &\min_{i\neq j} \|c_i - c_j\|_2 \geq 2\varepsilon \text{ and} \\ &\max_{i\in[\![1,K]\!]} \|c_i\|_2 \leq R\}\end{aligned} \tag{26}$$

We can now show that minimizing the sketching problem 2 is equivalent to controlling the excess risk problem 18 with high probability.

Let us first suppose that we have drawn independently m frequencies $(\omega_i)_{i\in[\![1,m]\!]} \in \mathbb{R}_+^{m\times d}$ according to the distribution Λ_s with density:

$$\forall \omega \in \mathbb{R}^d, \quad \Lambda_s(\omega) := Z_s\left(1 + \frac{s^2\|\omega\|_2^2}{d}\right)e^{-\frac{s^2\|\omega\|_2^2}{2}} \tag{27}$$

where $s > 0$ is a scaling parameter and Z_s the normalizing constant.

Let us finally define:

$$\varepsilon := 4s\sqrt{\log(eK)} \tag{28}$$

and take $R \geq \varepsilon$ (ε will be the lower bound on the distance between two different centroids in our result) .

Let us assume that we have found an estimate $\widehat{c}$ of the centroids up to some non-negative constants $\nu, \nu' \geq 0$ satisfying:

$$\mathsf{R}_{\text{clust.}}(\pi, \widehat{c}) \leq (1 + \nu)\min_{\mathcal{H}_{K,2\varepsilon,R}} \mathsf{R}_{\text{clust.}}(\pi, c) + \nu'. \tag{29}$$

Of course, the smaller the constants ν, ν', the better the estimation $\widehat{c}$.

Theorem 2.1. *There exists $C > 0$ such that, for all $(\zeta, \delta) \in (0, 1)$, if m satisfies the following condition:*

$$\begin{aligned}m \geq C\delta^{-2}(k^2 d(1 + \log kd + \log R/\varepsilon + \log 1/\delta) + \log 1/\zeta) \\ \times \log(ke)\min(\log(ke), d),\end{aligned} \tag{30}$$

with probability $1-\zeta$, then the excess risk $\Delta_{\widehat{c}}\mathcal{R}(\pi, c^*)$ can be controlled as:

$$\begin{aligned}\Delta_{\widehat{c}}\mathcal{R}(\pi, c^*) \leq\ &(2 + \nu)C_{\mathcal{A}}\|\mathcal{A}\pi - \mathcal{A}\widehat{\pi}_n\|_2^2 \\ &+ (2 + \nu)C_{\mathcal{A}}\|\mathcal{A}\pi - \mathcal{A}\pi^*\|_2^2 + d(\pi^*, \mathcal{H}_{K,2\varepsilon,R}) + C_{\mathcal{A}}\nu'\end{aligned} \tag{31}$$

where $C_{\mathcal{A}}$ is a non-negative constant defined as $C_{\mathcal{A}} = 224\sqrt{K/(1-\delta)}R^2$, and $d(\pi^*, \mathcal{H}_{K,2\varepsilon,R})$ measures the distance from c^* to the set $\mathcal{H}_{K,2\varepsilon,R}$.

This theorem is quite consistent and few explications are needed:

- the first step is to chose an $\varepsilon \geq 0$ that will be our minimal centroid separation distance (we only look for centroids separated by at least 2ε),
- we can then set the scale parameter of the frequency distribution (27) according to the relation (28) and chose the sketch size m according to the criteria (30),
- the core of the method is now to minimize the quantity $\mathsf{R}_{\text{clust.}}(\widehat{\pi}_n, c)$ obtained by the sketching procedure to get an estimate of the centroids $\widehat{c}$ and the eq gives the theoretical guarantee,

Let us now analyze the control of the excess risk given in (31):

- the first term $\|\mathcal{A}\pi - \mathcal{A}\widehat{\pi}_n\|_2^2$ measures how far the empirical measure $\widehat{\pi}_n$ is from the underlying distribution π in the sketching space. It has been proven that this quantity behaves as $n^{-1/2}$ as $n \to \infty$. This term is quite natural,

the control on the risk is better when a larger number of samples have been drawn,

- the second term $\|\mathcal{A}\pi - \mathcal{A}\pi^*\|_2^2$ measures the distance between π and its associated centroid version defined in (25). It is important to notice that the centroid of π^* might not satisfy the ε-separation constraint (since the minimization is done on $c \in \mathbb{R}^{d\times K}$ and not in $\mathcal{H}_{K,2\varepsilon,R}$). This quantity gives an idea of the "clusterability" of π. When the quantity is small, the mass of π is concentrated on some locations corresponding to the centroids c^*. If the quantity is large, the mass of π is unlocalized and thus not well "clusterable". It is natural to think that the excess risk is better controlled when the underlying distribution is well localized,
- the third term measures the distance between c^* and $\mathcal{H}_{K,2\varepsilon,R}$. As said before, π^* might not respect the separability condition. Of course, this term vanishes if the previous condition is satisfied.
- finally, a residual constant term is inherited from the quality of the minimization (29). The better the minimization, the smaller the values of ν and ν'. In case of a perfect minimization, both constants vanish.

These results, published five years after the introduction of the compressive K-means method give strong theoretical guarantees. However, some questions still hold. In particular, the results are proven for the distribution Λ_s, but is this distribution the only one giving the result ? Also, the result is valid for m satisfying a condition of the form $m \geq k^2 d$ (we have dropped the log factors). With intuition coming from the compressive sensing film, one would expect to have a linear relation w.r.t. k, *i.e.* to have a condition of the form $m \geq kd$.

3 NUMERICAL EXPERIMENTS

In the previous section we have seen why solving the minimization problem (2) could give an approximate solution to the original problem (1). This led to a very interesting result but there is still the task of minimizing in the sketching space. In the paper, the authors give the algorithm CLOMPR. The numerical results displayed in this report have been produced using our implementation (in `Python`).

3.1 Choice of the frequency distribution

The choice of the frequency distribution used in the sketching step of the compressive KMeans algorithm significantly impacts the quality of the clustering. We tested four different distributions: the theoretical adapted radius, the adapted radius, folded gaussian, and uniform. To evaluate clustering performance, we computed the sum of squared errors (SSE), where a lower SSE indicates better clustering. In addition to that, we computed the mean SSE and its standard deviation by repeating the experiment 50 times to evaluate the stability of the results.

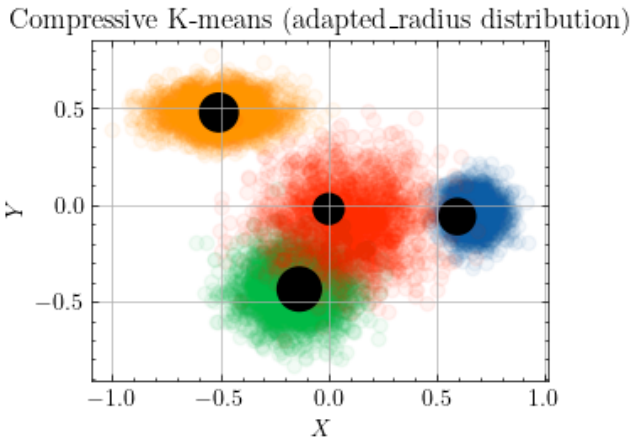

Figure 3: Compressive KMeans with adapted radius distribution

For the theoretical adapted radius distribution, the mean SSE across 50 experiments was 127 with a standard deviation of 61, demonstrating robust clustering performance with relatively low variance. In the experiment displayed in Figure 3, this distribution adjusts the frequencies according to the dataset's structure, leading to consistent and accurate clustering.

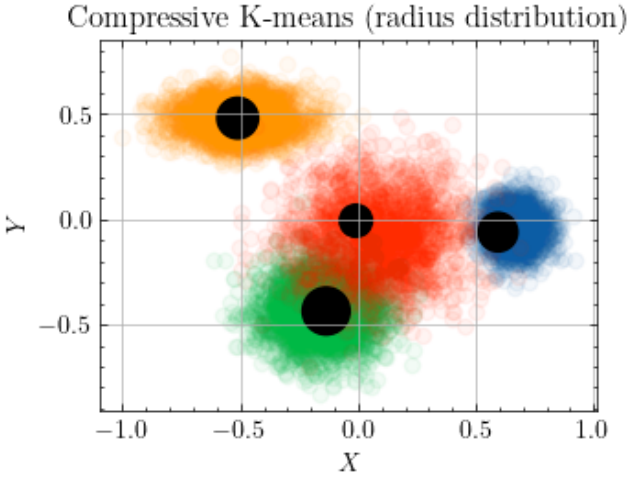

Figure 4: Compressive KMeans with radius distribution for KMeans

For the radius distribution, the mean SSE was 141 with a standard deviation of 66, which is slightly higher than the theoretical radius distribution.

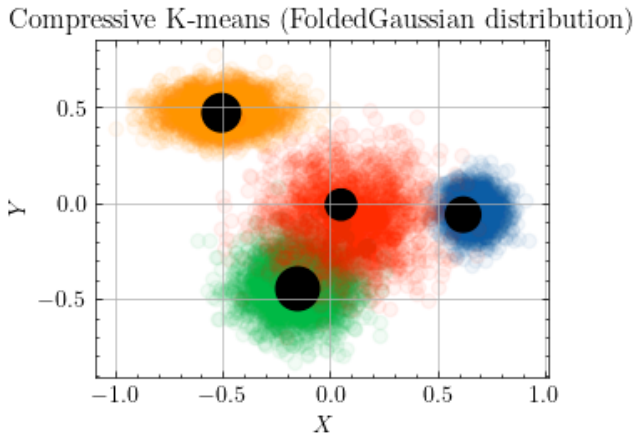

Figure 5: Compressive KMeans with folded Gaussian distribution

The folded Gaussian distribution achieved a mean SSE of 161 with a standard deviation of 85, showing a slightly less effective sketching and clustering compared to the adapted radius distribution

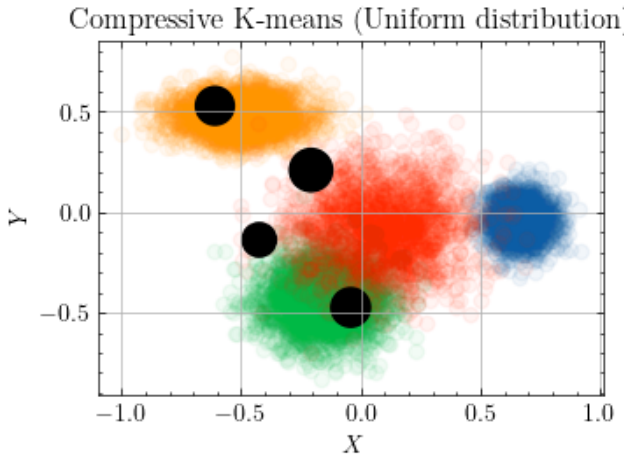

Figure 6: Compressive KMeans with uniform distribution

Lastly, the uniform distribution performed the worst. Its mean SSE was 344, with a standard deviation of 491, indicating both high error and high instability. In the experiment displayed in Figure 6, the SSE was 1762, illustrating that the uniform selection of frequencies does not capture the dataset's structure effectively, leading to poor clustering results.

In summary, the adapted radius distribution outperforms the others, providing the best balance between accuracy and theoretical guarantees, as suggested in the reference [9]. The folded Gaussian distribution also performs well. In contrast, the uniform distribution perform poorly, with much higher SSE values, indicating they are less suitable choices for this task. This highlights the importance of choosing a frequency distribution that aligns with the data's characteristics and the algorithm's theoretical foundations for optimal performance in compressive clustering techniques.

3.2 Choice of m

The plot in Figure 7 illustrates the **evolution of the SSE** as the number of sketching frequencies m increases for compressive K-means using the *adapted_radius* distribution.

- **Rapid decrease:** For $m < 20$, the SSE drops significantly as more frequencies improve clustering. High SSE for $m \leq 5$ reflects insufficient data representation.
- **Stabilization:** After $m \approx 20$, the SSE stabilizes, confirming that precision is achieved with $m = O(Kd)$. This shows there is no need to increase m to $O(K^2 d)$, which is remarkable. For this experiment, $K = 4$ and $d = 2$.

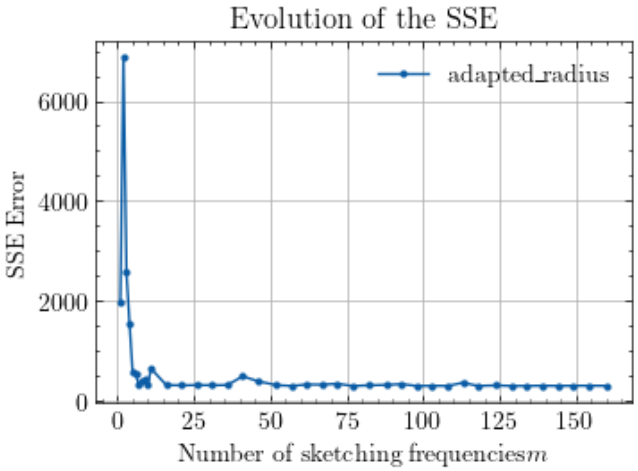

Figure 7: Evolution of the SSE with respect to m

4 CONCLUSION

During this project, we dived into the fascinating world of compressive learning. In particular, we discovered a method to significantly alleviate clustering problems. This compression method has the advantage of being easily adjustable, streamable and privacy aware. However, until 3 years ago, no theoretical guarantee could justify the performance of this algorithm. In this report, we recall the theoretical guarantees recently presented in the literature. Furthermore, thanks to our Python implementation, we highlight that among the two main hypotheses, the hypothesis on m could surely be sharper and we could question which distributions could maintain theoretical convergence.

REFERENCES

[1] Corinna Cortes and Vladimir Naumovich Vapnik. 1995. Support-Vector Networks. *Machine Learning* 20 (1995), 273–297. https://api.semanticscholar.org/CorpusID:52874011

[2] Robert Durrant and Ata Kaban. 2013. Sharp Generalization Error Bounds for Randomly-projected Classifiers. In *Proceedings of the 30th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 28)*, Sanjoy Dasgupta and David McAllester (Eds.). PMLR, Atlanta, Georgia, USA, 693–701. https://proceedings.mlr.press/v28/durrant13.html

[3] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Knowledge Discovery and Data Mining*. https://api.semanticscholar.org/CorpusID:355163

[4] Moran Feldman, Amin Karbasi, and Ehsan Kazemi. 2018. Do Less, Get More: Streaming Submodular Maximization with Subsampling. *ArXiv* abs/1802.07098 (2018). https://api.semanticscholar.org/CorpusID:3444158

[5] Gereon Frahling and Christian Sohler. 2006. A fast k-means implementation using coresets. In *Proceedings of the twenty-second annual symposium on Computational geometry*. 135–143.

[6] Rémi Gribonval, Gilles Blanchard, Nicolas Keriven, and Yann Traonmilin. 2020. Statistical learning guarantees for compressive clustering and compressive mixture modeling. *arXiv preprint arXiv:2004.08085* (2020).

[7] S C Johnson. 1967. Hierarchical clustering schemes. *Psychometrika* 32 (1967), 241–254. https://api.semanticscholar.org/CorpusID:930698

[8] S. Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (1982), 129–137. https://doi.org/10.1109/TIT.1982.1056489

[9] Keriven Nicolas, Tremblay Nicolas, Traonmilin Yann, and Gribonva Rémi. 2017. Compressive K-means. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[10] Hugo Reboredo, Francesco Renna, Robert Calderbank, and Miguel R. D. Rodrigues. 2013. Projections designs for compressive classification. In *2013 IEEE Global Conference on Signal and Information Processing*. 1029–1032. https://doi.org/10.1109/GlobalSIP.2013.6737069

[11] Christopher Williams and Matthias Seeger. 2000. Using the Nyström Method to Speed Up Kernel Machines. In *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp (Eds.), Vol. 13. MIT Press. https://proceedings.neurips.cc/paper_files/paper/2000/file/19de10adbaa1b2ee13f77f679fa1483a-Paper.pdf