Geometric Data Analysis Project
Compressive K-Means

Ben Kabongo B.
ben.kabongo_buzangu@ens-paris-saclay.fr
ENS Paris-Saclay, Master MVA
Paris, France

ABSTRACT

The **Compressive K-Means** paper proposes a compressive version of the K-Means algorithm, adapted to very high-dimensional datasets. The approach consists in building a **sketch**, a size independent representation of the dataset, and retrieving cluster centers from the sketch using the CLOMPR (Compressive Learning Orthogonal Matching Pursuit) algorithm.

In this project report, we'll start with a quick look at clustering with K-Means. We will then present the approach proposed by the Compressive K-Means (CKM) algorithm for clustering. We will then take up the guarantees given and the experiments performed by the authors of the paper, which we will objectively criticize through our own experiments in which we compare the basic K-Means algorithm and the CKM algorithm. Finally, we conclude.

KEYWORDS

K-Means, Clustering, Compressive Sensing, Compressive Learning, Random Fourier Features

1 INTRODUCTION

The aim of clustering is to partition a data set into homogeneous and disjoint subsets, such that the data in each subset share common characteristics, according to proximity criteria defined by introducing distance measures and classes between examples.

Algorithm 1 K-Means Algorithm

```
Input: Data set $X$, number of clusters $k$
Output: Cluster centroids $C_1, C_2, \ldots, C_k$
Initialize cluster centroids $C_1, C_2, \ldots, C_k$ randomly from $X$
repeat
    for each data point $x_i \in X$ do
        Assign $x_i$ to the nearest centroid:
            $c_i = \arg\min_j \|x_i - C_j\|^2$
    end for
    for each cluster $C_j$ do
        Update centroid $C_j$:
            $C_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$
    end for
until Convergence
```

1.1 K-Means

K-Means [5] is a clustering algorithm that calculates cluster centers with the aim of minimizing the sum of squared errors (SSE), which is the sum of the distances from each point to its cluster center.

$$\text{SSE}(X, C) = \sum_{i=1}^{n} \min_{k} \|x_i - c_k\|^2 \tag{1}$$

1.2 Compressive K-Means (CKM)

CKM [4] is an algorithm offering a compressive approach to K-Means. The idea is to build a compressive representation of the data, called a sketch, independent of the number of examples and the size of the data. The sketch is then used to find the cluster centers using a Compressive Learning Orthogonal Matching Pursuit (CLOMPR) algorithm [3], adapted for K-Means.

2 EVALUATION METRICS

Consider the two toy datasets in Figure 1. The first dataset has 3 clusters, correctly separated and far apart in terms of distance from each other. The second has 2 clusters, one nested within the other.

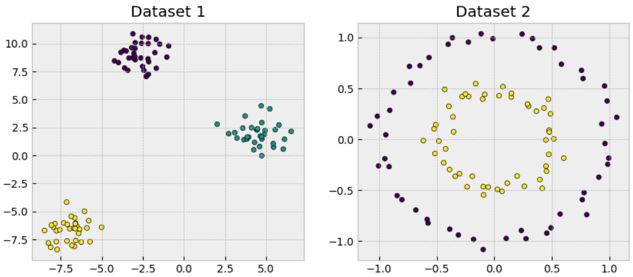

Figure 1: Toy datasets. The dataset on the left has three correctly disjointed clusters, and the one on the right has two with one nested within the other.

We apply the K-Means algorithm to the toy datasets, and plot the K-Means-induced clustering in Figure 2. Each example is then associated with the nearest centroid (cluster center). Clustering algorithms such as K-Means have difficulty separating nested clusters. We note that the clusters in the first dataset were found correctly. However, although they have the same number of elements, the normalized SSE (divided by the number of examples) of the first clustering is greater than that of the second clustering. This raises the question of whether SSE is a sufficient indicator of good clustering.

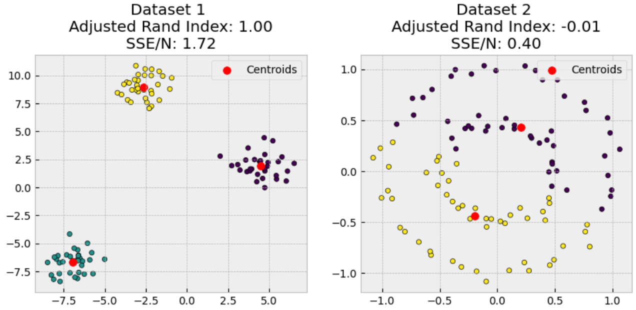

Figure 2: K-Means on toy datasets. The colors of the examples correspond to their K-Means cluster. Cluster centers are shown in red.

In the Compressive K-Means paper, the only clustering metric used for the comparison between CKM and K-Means is the SSE. We thought it would be useful to use other clustering metrics to compare the two algorithms. In this section, we will present these different metrics, in addition to the SSE already introduced.

2.1 ARI (Adjusted Rand Index)

The Adjusted Rand Index measures the similarity between the true labels and the predicted labels, correcting for chance. It considers all pairs of samples and counts the agreements and disagreements between true and predicted clusters.

$$\text{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \frac{\left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]}{\binom{N}{2}}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \frac{\left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]}{\binom{N}{2}}}$$

2.2 AMI (Adjusted Mutual Information)

The Adjusted Mutual Information measures the mutual information between true and predicted labels, adjusted for chance. It quantifies the agreement between the two sets of labels while considering chance.

$$\text{AMI} = \frac{I(Y;Z) - \mathbb{E}[I(Y;Z)]}{\max\{H(Y),H(Z)\} - \mathbb{E}[I(Y;Z)]}$$

2.3 Homogeneity

Homogeneity measures the extent to which each cluster contains only members of a single class. It is high when all clusters are pure.

$$\text{Homogeneity} = 1 - \frac{H(C|K)}{H(C)}$$

2.4 Completeness

Completeness measures the extent to which all members of a given class are assigned to the same cluster. It is high when all members of a class are in the same cluster.

$$\text{Completeness} = 1 - \frac{H(K|C)}{H(K)}$$

2.5 V-Measure

V-Measure is the harmonic mean of homogeneity and completeness, providing a balanced measure.

$$\text{V-Measure} = 2 \times \frac{\text{Homogeneity} \times \text{Completeness}}{\text{Homogeneity} + \text{Completeness}}$$

In Table 1, the various metrics show that the clustering of the first dataset is better than that of the second.

Dataset	**1**	**2**
ARI	1.0	-0.01
AMI	1.0	-0.01
Homogeneity	1.0	0.00
Completeness	1.0	0.00
V-Measure	1.0	0.00
SSE/N	1.72	0.40

Table 1: Clustering Evaluation Metrics for Toy Datasets 1 and 2

SSE remains a highly relevant metric, especially in a completely unsupervised context, where the real distribution of clusters is not available. In the following, we will use the metrics presented to evaluate the different algorithms.

3 COMPRESSIVE K-MEANS (CKM)

3.1 Compressive sensing

The concept of compressive sensing was initially developed and successfully applied to sparse signals of finite dimension. This approach allows the recovery of such signals from a number of measurements significantly smaller than the total dimension of the space. Currently, the scope of compressive sensing has been extended to more general classes of signals, including low-rank matrices and functions.

These generalizations are based on the a priori assumption that the data resides in a much smaller subset than the surrounding space. This makes it possible to non-adaptively compress these data into a reduced representation, while preserving the essential information about them. More precisely, this approach enables data to be reconstructed with controlled accuracy, even after significant compression, while keeping the fundamental features intact.

The compressive representation of the data can then be exploited for machine learning tasks, such as fitting a parametric model to the data.

Consider the dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, with $\mathbf{x}_i \in \mathbb{R}^n$. Let N be the number of examples and n the dimension of the examples.

In traditional compressive sensing, a projection matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$ which projected the data into a reduced space of dimension m, such that $m << n$. The compressed version of the data is then given by $\mathbf{y} \approx \mathbf{Mx}$.

The generalization of compressive sensing applies a linear operator $\mathbf{A}$ to the data distribution p. This gives:

$$\mathbf{z} = \mathbf{A}p \in \mathbb{C}^m \tag{2}$$

Algorithm 2 CKM (CLOMPR for K-Means) Algorithm

```
Input: Sketch $\mathbf{z}$, frequencies $\mathbf{\Omega}$, number of clusters $k$, centroids bounds $\mathbf{l}$, $\mathbf{u}$
Output: Centroids $\mathbf{C}$, weights $\alpha$
$\mathbf{r} = \mathbf{z}$
$\mathbf{C} = \emptyset$
for $t \leftarrow 1$ to $2K$ do
    Step 1 : Find a new centroid
        $\mathbf{c} \leftarrow \text{maximize}_{\mathbf{c}}\left(\text{Re}\langle \frac{\mathbf{A}\delta_{\mathbf{c}}}{\|\mathbf{A}\delta_{\mathbf{c}}\|}, \mathbf{r}\rangle, \mathbf{l}, \mathbf{u}\right)$
    Step 2 : Expand support
        $\mathbf{C} = \mathbf{C} \cup \{\mathbf{c}\}$
    Step 3 : Enforce sparsity by Hard Thresholding if $t > K$
    if $|\mathbf{C}| > K$ then
        $\beta \leftarrow \text{argmin}_{\beta \geq 0} \left\| \mathbf{z} - \sum_{k=1}^{|\mathbf{C}|} \beta_k \frac{\mathbf{A}\delta_{\mathbf{c_k}}}{\|\mathbf{A}\delta_{\mathbf{c_k}}\|} \right\|$
        Select $K$ largest entries $\beta_1, \beta_2 \ldots \beta_K$
        Reduce the support $\mathbf{C} = \{\mathbf{c}_k\}_{k=1}^{K}$
    end if
    Step 4 : Project to find $\alpha$
        $\alpha \leftarrow \text{argmin}_{\alpha \geq 0} \left\| \mathbf{z} - \sum_{k=1}^{|\mathbf{C}|} \alpha_k \mathbf{A}\delta_{\mathbf{c_k}} \right\|$
    Step 5 : Global gradient descent
        $\mathbf{C}, \alpha \leftarrow \text{minimize}_{\mathbf{C},\alpha}\left( \left\| \mathbf{z} - \sum_{k=1}^{|\mathbf{C}|} \alpha_k \mathbf{A}\delta_{\mathbf{c_k}} \right\|, \mathbf{l}, \mathbf{u} \right)$
    Update residual:
        $\mathbf{r} \leftarrow \mathbf{z} - \sum_{k=1}^{|\mathbf{C}|} \alpha_k \mathbf{A}\delta_{\mathbf{c_k}}$
end for
```

3.2 Sketching operator

CKM applies the CLOMPR algorithm to the initial dataset sketch to find the centroids.

Let be :

- $\mathbf{Y} = \{\mathbf{y}_l\}_{l=1}^{L}$, a dataset with $\mathbf{y}_l \in \mathbb{R}^n$
- $\beta = \{\beta_l\}_{l=1}^{L}$: L weights in $\mathbb{R}$
- $\Omega = \{\mathbf{w}_j\}_{j=1}^{m}$: m frequencies in $\mathbb{R}^n$, learned on a portion of the dataset as indicated in another paper by the authors [3].

The sketching operator for the CKM algorithm is given by :

$$\text{Sk}(\mathbf{Y}, \beta) = \left[\sum_{l=1}^{L} \beta_l e^{-i\mathbf{w}_j^T \mathbf{y}_l} \right]_{j=1}^{m} \in \mathbb{C}^m \tag{3}$$

The sketching operator produces a sketch of size m, independent of the dataset size.

This sketching operator can be expressed as a linear $\mathbf{A}$ operator with respect to probability distributions.

$$\mathbf{A}p_{\mathbf{Y},\beta} = \left[\mathbb{E}_{\mathbf{y} \sim p_{\mathbf{Y},\beta}} e^{-i\mathbf{w}_j^T \mathbf{y}} \right]_{j=1}^{m} \in \mathbb{C}^m \tag{4}$$

We therefore note that $p_{\mathbf{Y},\beta} = \sum_{l=1}^{L} \beta_l \delta_{y_l}$ and that when the weight vector is only indicated it is a uniform weight distribution $p_{\mathbf{Y}} = \frac{1}{L} \sum_{l=1}^{L} \delta_{y_l}$.

3.3 Compressive K-Means

Let be a dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, with $\mathbf{x}_i \in \mathbb{R}^n$. We want to find the set $\mathbf{C} = \{\mathbf{c}_k\}_{k=1}^{K}$ of K centroids, with $\mathbf{c}_k \in \mathbb{R}^n$.

CKM proposes a new formulation of K-Means given by :

$$\text{argmin}_{\mathbf{C},\alpha} \|\text{Sk}(\mathbf{X}, \frac{\mathbf{1}}{N}) - \text{Sk}(\mathbf{C}, \alpha)\|_2^2 \tag{5}$$

with $\alpha \geq 0$ and $\sum_{k=1}^{K} \alpha_k = 1$. The problem can also be formulated as :

$$\text{argmin}_{\mathbf{C},\alpha} \|\mathbf{z} - \mathbf{A}p_{\mathbf{C},\alpha}\|_2^2 \tag{6}$$

with $\mathbf{z} = \mathbf{A}p_{\mathbf{X}}$. This formulation is somewhat inspired by the SSE, but is not the same. In CKM, the objective is to minimize the difference between the data sketch and the centroid sketch. The CKM algorithm is given in 2.

4 EXPERIMENTATIONS

4.1 Preliminary 2D tests

The CKM method attempts to tackle large datasets in high dimensions. Before tackling very high dimensions, we decided to tackle a two-dimensional problem. We generate 3 separable Gaussian distributions of 10,000 points each. $N = 3 \times 10^4$ and $K = 3$ clusters. The dataset is shown in Figure 3.

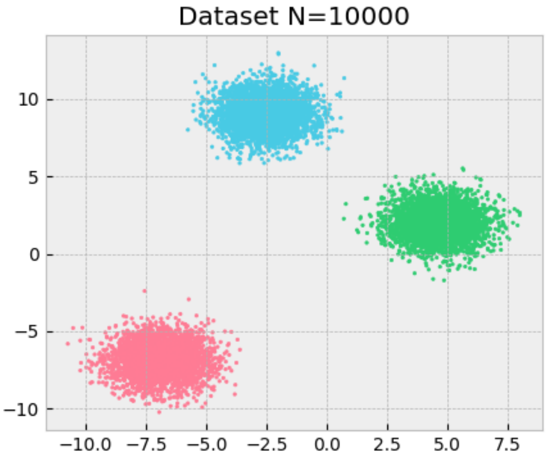

Figure 3: Dataset with 3 separable Gaussian distributions of 10,000 points each.

The dataset corresponds to dataset 1 in Figure 2, for which the K-Means algorithm easily finds the centroids.

We use cross-validation to find the sketch size that minimizes the SSE, for the following sketch sizes: KN, $2KN$, $3KN$, $4KN$. We also evaluate the clustering induced by the different sketch sizes on all the other metrics mentioned above. We report the results of these evaluations in Figure 4, and in Figure 5 the clustering induced by the sketch size that minimizes the SSE. With the configuration of this experiment, we conclude that CKM has difficulty clustering on 2D data.

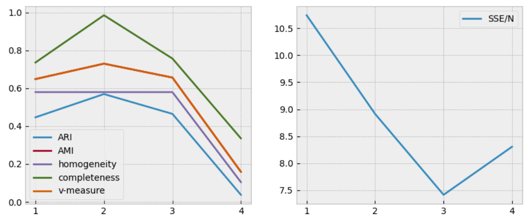

Figure 4: Evaluation of CKM on 2D data as a function of sketch size (KN).

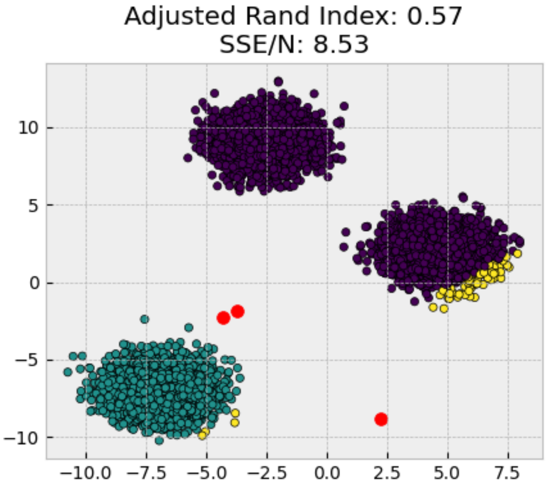

Figure 5: CKM clustering of 2D data, minimizing SSE.

4.2 Clustering of digits dataset

As part of this experiment, we recovered datasets of digit images. Our experiments were based on the digits dataset $N = 1797$, $K = 10$ clusters, $n = 8x8 = 64$ features. In these experiments, we attempt to compare the K-Means and CKM algorithms in terms of different metrics and different types of initialization.

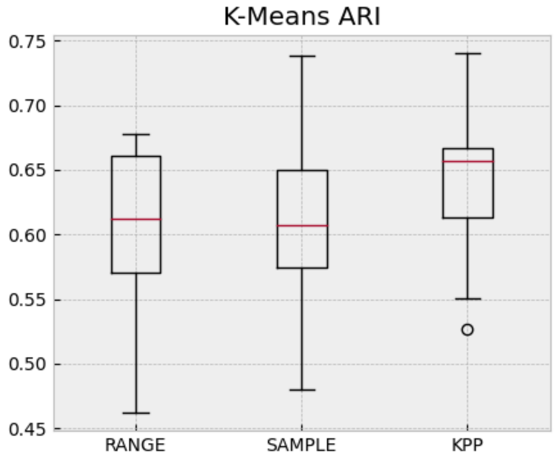

Figure 6: Evaluation of the K-Means algorithm on digits data with the ARI metric.

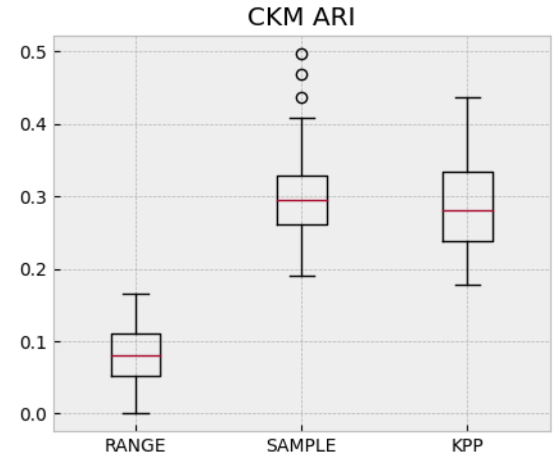

Figure 7: Evaluation of the CKM algorithm on digits data with the ARI metric.

For the K-Means and CKM algorithms, three different types of centroid initialization can be distinguished:

- **Random/Range**: random initialization of centroids. In the case of CKM, each centroid is drawn uniformly, taking into account the bounds passed to the algorithm.
- **Sample**: initialize centriodes by randomly selecting K examples from the dataset.
- **K-means++**: initialization given by K-Means ++ algorithm [1]. For CKM, select $c = x_i$ from the data with a probability inversely proportional to its distance to the current set of centroids.

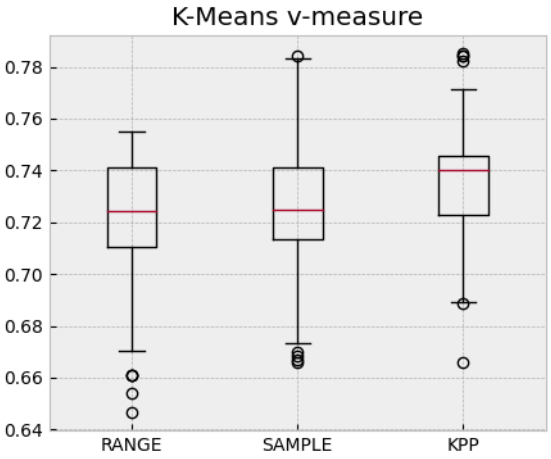

Figure 8: Evaluation of the K-Means algorithm on digits data with the v-mesure metric.

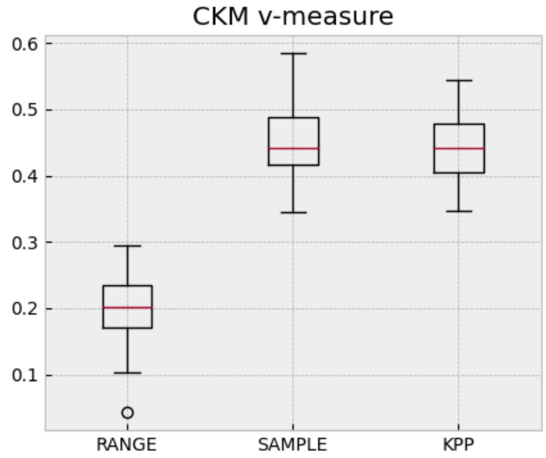

Figure 9: Evaluation of the CKM algorithm on digits data with the v-mesure metric.

To compare them on the different metrics, we ran the K-Means and CKM algorithms 100 times on the digits dataset for the three different types of initialization. In Figure 6 and 7, we report the ARI scores of the K-Means and CKM algorithms respectively; in Figure 8 and 9, we report the v-measure scores; and in Figure 10 and 11, we report the SSE/N scores. We note that with regard to clustering evaluation metrics, both algorithms are sensitive to centroid initialization types. For all metrics, the K-means++ initialization gives better clustering on average than the sample initialization, which in turn gives better clustering on average than the random initialization.

Metric	K-Means	CKM
ARI		
RANGE	**0.605** (0.054)	0.084 (0.040)
SAMPLE	**0.609** (0.048)	0.298 (0.061)
KPP	**0.642** (0.045)	0.288 (0.058)
AMI		
RANGE	**0.718** (0.025)	0.194 (0.053)
SAMPLE	**0.720** (0.024)	0.442 (0.049)
KPP	**0.732** (0.021)	0.435 (0.050)
Homogeneity		
RANGE	**0.709** (0.029)	0.160 (0.052)
SAMPLE	**0.711** (0.027)	0.434 (0.051)
KPP	**0.728** (0.024)	0.427 (0.051)
Completeness		
RANGE	**0.733** (0.021)	0.284 (0.050)
SAMPLE	**0.735** (0.021)	0.464 (0.046)
KPP	**0.743** (0.018)	0.456 (0.048)
V-measure		
RANGE	**0.721** (0.025)	0.202 (0.053)
SAMPLE	**0.723** (0.024)	0.448 (0.048)
KPP	**0.735** (0.021)	0.441 (0.049)
SSE/N		
RANGE	**25.099** (0.214)	59.227 (1.494)
SAMPLE	**25.132** (0.238)	35.637 (0.938)
KPP	**25.044** (0.168)	35.664 (1.014)

Table 2: Results of the ARI, AMI, Homogeneity, Completeness, V-measure, and SSE/N evaluation metrics for clustering with K-Means and CKM on the digits dataset. Values are presented with mean and standard deviation in parentheses.

In addition, we find that for all measures, the K-Means algorithm gives better clustering on average than the CKM algorithm. We report in Table 2 the results of the comparisons of the two algorithms according to all the metrics on the digits dataset.

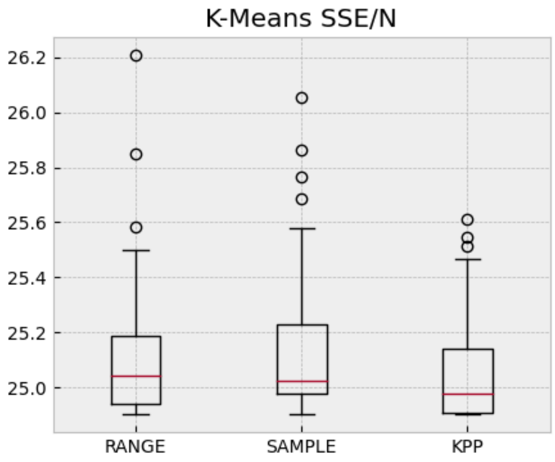

Figure 10: Evaluation of the K-Means algorithm on digits data with the SSE metric.

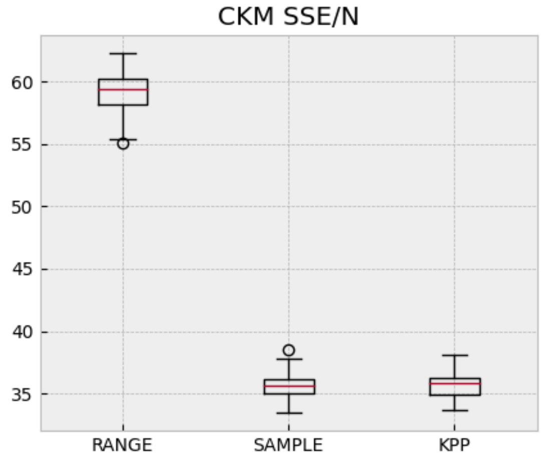

Figure 11: Evaluation of the CKM algorithm on digits data with the SSE metric.

5 CONCLUSION

Clustering is an important task in machine learning. In an unsupervised way, it is very often relevant to partition data into subsets sharing the same characteristics. A well-known clustering algorithm is the K-Means algorithm. However, this is an initialization-sensitive algorithm whose complexity increases proportionally with data size. It is therefore highly relevant to propose efficient algorithms that reduce the complexity of the K-Means algorithm when the data is voluminous and in very high dimensions. Solutions such as combining K-Means with a dimension reduction algorithm such as PCA [2] come to mind.

CKM is an algorithm that attempts to solve the problem of the complexity of the K-Means algorithm for large data. However, based on the experiments we have carried out, the K-Means algorithm presents the best results than the CKM algorithm. The results of the CKM algorithm are not far from the K-Means algorithm. Perhaps an optimal calculation of the sketch would allow better results to be obtained.

REFERENCES

[1] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The Advantages of Careful Seeding. In *ACM-SIAM Symposium on Discrete Algorithms*. 1027–1035.

[2] Chris Ding and Xiaofeng He. 2004. K-means Clustering via Principal Component Analysis. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*. https://icml.cc/Conferences/2004/proceedings/papers/262.pdf

[3] N. Keriven, A. Bourrier, R. Gribonval, and P. Pérez. 2016. Sketching for Large-Scale Learning of Mixture Models. *arXiv preprint arXiv:1606.02838* (2016), 1–50.

[4] Nicolas Keriven, Nicolas Tremblay, Yann Traonmilin, and Rémi Gribonval. 2016. Compressive K-means. *arXiv preprint arXiv:1610.08738* (2016). https://arxiv.org/pdf/1610.08738

[5] Stuart P. 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