

PAPER • OPEN ACCESS

## Research on K-means Algorithm Optimization based on Compression Learning

To cite this article: Cai Shuai *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **569** 052038

View the [article online](#) for updates and enhancements.

# Research on K-means Algorithm Optimization based on Compression Learning

Cai Shuai<sup>1</sup>, Zhu Lei<sup>1\*</sup>, Weijun Zeng<sup>1</sup>, Re Yu<sup>1</sup>, and Zhao Xiao<sup>1</sup>

<sup>1</sup>Institute of Communications Engineering, Army Engineering University of PLA, Nan Jing, Jiang Su, 210001, China

\*Corresponding author's e-mail: cs290448832@163.com

**Abstract.** The K-means algorithm is one of the classical algorithms of clustering. However, as the data set increases, the computational cost of clustering becomes higher. The orthogonal matching pursuit algorithm is a classic signal reconstruction algorithm. The paper improves its algorithm based on compression learning and applies it to the K-means algorithm, which uses the sketch of the original data set to estimate the cluster center. The experiment results show that the clustering effect of this method is similar to that of K-means algorithm, because the size of the sketch is independent of the size of the original data set, only related to the number of centroids K and the dimension n of the data, which reduces the computational complexity of the algorithm. For large data sets, experiments show that the improved algorithm is more optimized than the traditional K-means algorithm.

## 1. Introduction

The K-means algorithm is a partitioning based clustering algorithm. The algorithm is a typical partitioning algorithm that is easy to understand, easy to implement, theoretically reliable, and widely used. Compared with other algorithms, the k-means clustering algorithm has the advantages of high efficiency and strong scalability when processing large data sets. The algorithm can not only process data sets in numerical form, but also can be applied to image features and texts. The algorithm uses iterative update method to determine the cluster center, which is beneficial to the selection of the initial center point for getting closer to the real cluster center point, and the target function will become smaller and smaller, and the clustering effect is getting better and better. The K-means algorithm uses the sum of squared errors (SSE) as the objective function for measuring clustering quality effects.

$$SSE(X, C) = \sum_{i=1}^N \min_k \|x_i - c_k\|^2 \quad (1)$$

The Lloyd-Max algorithm is a classical method of performing K-means clustering. However, as the training data set grows, its computational cost becomes too high. So a compressed version of K-means (CKM) is proposed, which estimates the cluster center from the sketch. It estimates the cluster center from the sharply compressed representation of the training data set. This paper use the variant of the Compressed Learning Orthogonal Matching Tracking Replacement Algorithm (CLOMPR) to retrieve the centroid from the sketch which is algorithm originally used for large-scale Gaussian Mixture Model (GMM) estimation. Its complexity reads  $O(nmK^2)$ , so once the sketch is calculated, the dependency of N is completely eliminated. The complexity can further reduced by taking advantage of fast transformations or embedding in lower dimensions as a pre-processing step.



## 2. Orthogonal Matching Pursuit and Replacement Algorithm (OMPR)

The OMP algorithm guarantees the optimality of each iteration and reduces the number of iterations. But it only picks one atom in each iteration to update the set of atoms, which inevitably pays a huge rebuild time. The number of iterations is closely related to the sparsity  $K$  or the number of samples  $M$ . As it increases, the time cost will also increase significantly.

As a heuristic algorithm, a greedy approach is proposed. This method is inspired by orthogonal matching pursuit (OMP) and its variant OMP with Replacement (OMPR), which is used in each iteration. The support is extended iteratively by selecting the atoms most relevant to the residual. OMPR runs more times than OMP (usually  $2K$  instead of  $K$ ). In the spirit of CoSAMP, it extends support to the required sparsity before supporting, and then enforces it with hard threshold processing steps on each iteration.

As described below, the algorithm involves several modifications to the OMPR.

- Maximizes the associated real part instead of its modulus to avoid a negative correlation between atoms and residuals. Then, Perform non-negative least squares (NNLS) minimization instead of classical least squares minimization.
- Perform maximization in step 1, where the gradient rise maximum value  $\theta$  is randomly initialized, resulting in a local maximum of the correlation between the atom and the residual. Please note that atoms are standardized during the search process, just like OMP.
- Add step 5 to further reduce the cost function by minimizing the gradient drop initialized with the current parameter  $(\Theta, \alpha)$ .

## 3. Optimization method of k-means algorithm based on compression learning

In this paper, a heuristic method is present for finding the centroid  $C$  from the sketch of dataset  $X$ , where the size of  $m$  does not depend on  $N$ . More precisely, it is a sketch process  $Sk$  that defines a set of weighted vectors in  $R^n$  to be converted into vectors in  $C^m$ . And derive the centroid by finding a set  $C$  of weighted points with a sketch close to the data set  $X$  with uniform weight:

$$\arg \min_{C, \alpha} \|Sk(X, 1/N) - Sk(C, \alpha)\|_2^2, \quad \text{With } \alpha \geq 0, \quad \sum_{k=1}^K \alpha_k = 1 \quad (2)$$

As it can be seen, sketching a dataset requires only one pass through  $X$ , and it can also benefit from distributed or online computing.

### 3.1 Acquisition of sketches

Given  $m$  frequency vectors  $\Omega = \{\omega_1, \dots, \omega_m\}$ , The sketch of data set  $Y$  with  $L$  data is defined as

$$Sk(Y, \beta) = \left[ \sum_{l=1}^L \beta_l e^{-i\omega_l^T y_l} \right]_{j=1}^m \in \mathbb{C}^m \quad (3)$$

The sketch process can be reconfigured as operation  $A$ , which is linear with respect to the probability distribution. This operation is a sampling of the characteristic function of the probability distribution  $p$  of the frequency  $\omega_1, \dots, \omega_m$ . Define  $p_{Y, \beta} = \sum_{l=1}^L \beta_l \delta_{y_l}$ . The question (2) can be changed to:

$$\arg \min_{C, \alpha} \left\| \hat{z} - A p_{C, \alpha} \right\|_2^2 \quad (4)$$

Where  $\hat{z} = A \hat{p} x$  is sketch for the dataset. Empirical distribution of data  $\hat{p} x = \frac{1}{N} \sum_{i=1}^N \delta_{x_i}$ , In random

Fourier sampling, the frequency  $\omega_1, \dots, \omega_m$  comes from the frequency distribution  $\Lambda$ . In the previous work, an adaptive radius frequency distribution  $\Lambda$  was proposed, which is based on the GMM background. In the work [2], through the calculation of the data set (a small part), the small sketch and its adaptive regression are proposed, and the algorithm for selecting the proportional parameter is

proposed.

### 3.2 CKM algorithm

The CLOMPR algorithm is a heuristic algorithm for finding solutions to problems that have been proposed for Gaussian mixture model estimation in previous work. It is a greedy algorithm inspired by orthogonal matching pursuit (OMP) and its variant OMPR, which contains more iterations than OMP and has additional hard threshold processing steps.

Table 1. Algorithm: CLOMP for k-means(CKM).

<p><b>data:</b> Sketch <math>\hat{z}</math>, frequencies <math>\Omega</math>, parameter <math>K</math>, bounds <math>l, u</math> ;</p> <p><b>Result:</b> Centroids <math>C</math>, weights <math>\alpha</math></p>
<p><math>\hat{r} \leftarrow \hat{z}; C \leftarrow \emptyset</math> ;</p> <p>For <math>t \leftarrow 1</math> to <math>2K</math> do</p> <p>    <b>Step1:</b> Find a new centroid</p> $c \leftarrow \max_{\text{imize}_c} \left( \text{Re} \left\langle \frac{A\delta_c}{\ A\delta_c\ }, \hat{r} \right\rangle, l, u \right)$ <p>    <b>Step 2:</b> Expand support</p> $C \leftarrow C \cup \{c\}$ <p>    <b>Step 3:</b> Enforce sparsity by Hard Thresholding if <math>t &gt; K</math></p> <p>        if <math> C  &gt; K</math> then</p> $\beta \leftarrow \arg \min_{\beta \geq 0} \left\  \hat{z} - \sum_{k=1}^{ C } \beta_k \frac{A\delta_{c_k}}{\ A\delta_{c_k}\ } \right\ $ <p>        Select <math>K</math> largest entries <math>\beta_{i_1}, \dots, \beta_{i_K}</math></p> <p>        Reduce the Support <math>C \leftarrow \{c_{i_1}, \dots, c_{i_K}\}</math></p> <p>    <b>Step 4:</b> Project to find <math>\alpha</math></p> $\alpha \leftarrow \arg \min_{\alpha \geq 0} \left\  \hat{z} - \sum_{k=1}^{ C } \alpha_k \frac{A\delta_{c_k}}{\ A\delta_{c_k}\ } \right\ $ <p>    <b>Step 5:</b> Global gradient descent</p> $C, \alpha \leftarrow \min_{\text{imize}_{C, \alpha}} \left( \left\  \hat{z} - \sum_{k=1}^{ C } \alpha_k \frac{A\delta_{c_k}}{\ A\delta_{c_k}\ } \right\ , l, u \right)$ <p>    <b>Step 6:</b> Update residual: <math>\hat{r} \leftarrow \hat{z} - \sum_{k=1}^{ C } \alpha_k \frac{A\delta_{c_k}}{\ A\delta_{c_k}\ }</math></p>

### 3.3 Complexity of the CKM algorithm

In order to calculate the sketch, you must perform multiplication  $w^T X$ , Where  $X = [x_1, \dots, x_N]$  and  $W = [\omega_1, \dots, \omega_N]$  are matrices of data and frequency. It can be done in a distributed manner by splitting the data set over several computational units and averaging the resulting sketches so that the complete data does not need to be stored in a single location. People can also use GPU computing for very large-scale matrix multiplication [10].

Some technologies may further reduce these complexities. As detailed in [9], most of the operations in CKM can be scaled down to  $W$  and  $W^T$  to perform multiplication. It is also possible to reduce the size  $n$  to  $O(\log K)$  using random projection [8] as a pre-processing step.

Finally, the experimental results (see Section 4.3) indicate that the size of the sketch only needs to be linearly proportional to the number of parameters,  $m \approx O(nK)$ . Combined with all these results, it is possible to calculate a sketch in  $O(KNT^{-1}(\log K)^2)$ , where  $T$  is the number of parallel computing units and CKM is performed in  $O(K^3(\log K)^2)$ .

## 4. Experiments

### 4.1 Experimental environment

The MATLAB implementation of CKM provided by [6] is compared with the k-means function of Matlab that implements Lloyd-Max. Firstly, artificial clustering data drawn from a mixture of  $K$  units of Gaussian with a dimension  $n$  of uniform weights is used. Unless otherwise stated,  $N=3 \cdot 10^5$  points are generated from  $K = 10$  clusters, where  $n = 10$ . Use  $m = 1000$  frequencies unless otherwise stated. Each result averaged over 100 experiments.

The second problem is the spectral clustering of the MNIST dataset. In fact, in order to test the performance of our method on large datasets, the original  $7 \cdot 10^4$  image are used, and the toolbox infMNIST original image proposed in [7] are used to create the image.

### 4.2 CKM performance analysis

Lloyd-Max is usually randomly initialized several times and retains the set of centroids that produce the lowest SSE. In the CKM algorithm, SSE can't be accessed because the data is discarded after the sketch is calculated. Therefore, when performing CKM repetitions several times, minimize the centroid set of the cost function (4)

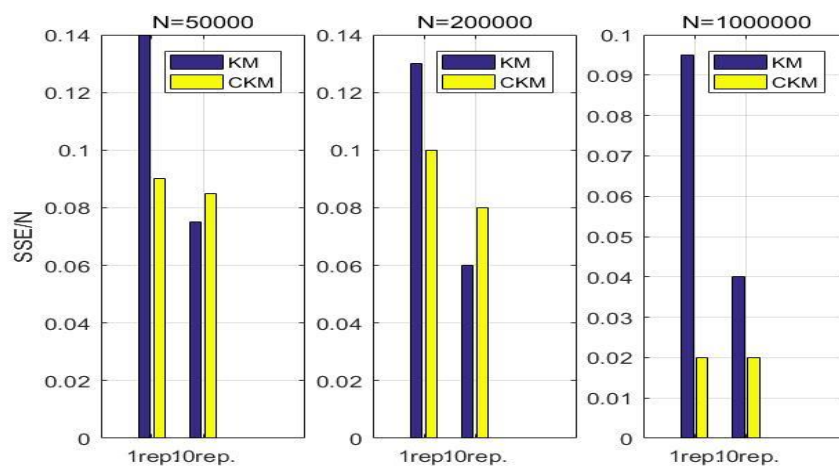


Figure 1. On the MNIST dataset, the mean and variance of more than 100 experiments of SSE were divided by  $N$ , for 1 or 10 replicates.

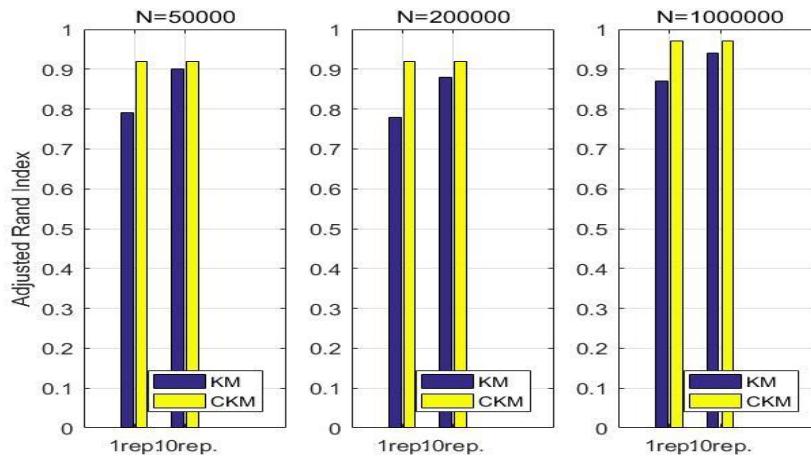


Figure 2. Adjusted Rand Index is used to compare clustering results on MNIST for 1 or 10 iterations.

As expected, k-means benefited a lot when executed multiple times, while CKM was more stable between repetitions. This allows CKM to run in practice with fewer repetitions than the actual k-means (more). Furthermore, for large data sets ( $N = 1000000$ ), the performance of CKM has a negligible variance and a negligible difference between 1 and 10 repetitions. Therefore, although the size  $m$  of the sketch remains fixed to all  $N$ , this method is actually more efficient when applied to large data sets. Moreover, in all cases, CKM is superior to k-means in classification. This may mean that the proposed cost function is more adaptive to this particular task than SSE.

#### 4.3 Frequency number $m$ analysis

Conscious assessment is needed on the effective frequency  $m$  for CKM.

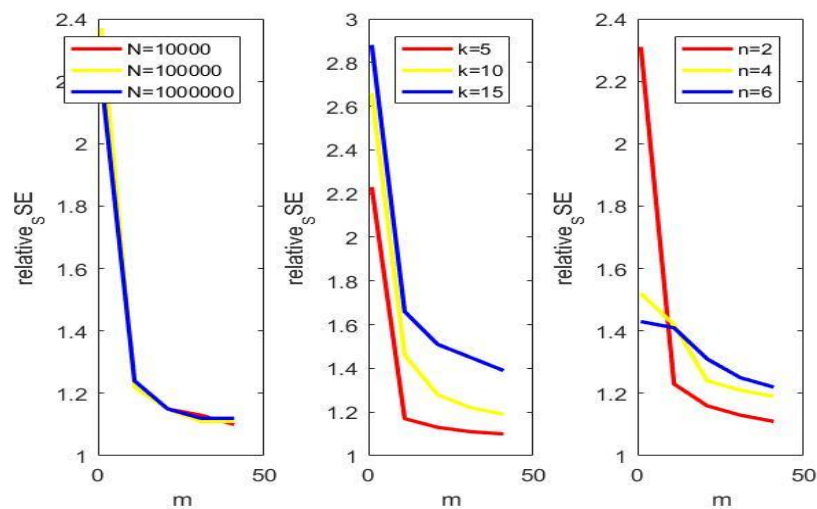


Figure 3. Relative SSE (i.e. SSE obtained with CKM divided by that obtained with k-means) on Gaussian data, The relationship between the size of  $m$  and the relationship between  $k$  and  $N$ .

In the left, when  $K=7$  and  $n=2$ , It can be seen that  $m$  is independent of  $N$  in the case of large data sets. In the middle, when  $N = 10000$  and  $n = 2$  remains unchanged,  $m$  is linear with  $k$ . In the right, when  $N = 10000$  and  $k = 7$  remains unchanged,  $m$  is linear with  $n$ .

A recent preliminary theoretical result on GMM [2] implies that for a fixed error level, the number of frequencies required increases in proportion to the number of parameters  $m \approx O(Kn)$ . It is assumed that the same phenomenon may be valid for K-means clustering.

## 5. Conclusion

This paper proposes a method of executing the K-means algorithm on a large dataset, where the centroid is derived from the sketch of the dataset. This problem is related to generalized compressed sensing. The results show that although the proposed goals are not directly related to traditional SSE costs, the approach is advantageous compared to the usual algorithms of K-means. Although the size of the sketch does not depend on  $N$ , the proposed algorithm is more efficient when applied to large data sets, such as complexity on the MNIST data set, SSE and classification performance, compared to the usual K-means. Looking ahead, the proposed method can be combined with dimensionality reduction and fast conversion to further speed up the method.

## References

- [1] N. Keriven, A. Bourrier, R. Gribonval, and P. P´er`ez. (2015) Sketching for Large-Scale Learning of Mixture Models. IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP).
- [2] N. Keriven, A. Bourrier, R. Gribonval, and P. P´er`ez. (2016) Sketching for Large-Scale Learning of Mixture Models. arXiv preprint arXiv:1606.02838, pp. 1–50.
- [3] S. Uw, A. Ng, M. Jordan, and Y. Weiss. (2001) On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems 14, pp. 849–856.
- [4] L. Le Magoarou and R. Gribonval. (2016). Flexible Multi-layer Sparse Approximations of Matrices and Applications. IEEE Journal of Selected Topics in Signal Processing, vol. 10, no. 4, pp. 688–700.
- [5] C. Boutsidis, A. Zouzias, and P. Drineas. (2010). Random Projections for k-means Clustering. Advances in Neural Information and Processing Systems (NIPS), pp. 298–306.
- [6] N. Keriven, N. Tremblay, and R. Gribonval. (2016) Sketch MLbox : a Matlab toolbox for large-scale learning of mixture models. <http://sketchml.gforge.inria.fr>.
- [7] G. Loosli, S. Canu, and L. Bottou. (2007) Training Invariant Support Vector Machines using Selective Sampling. Large Scale Kernel Machines, pp. 301–320.
- [8] B. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. Lanckriet. (2010) Hilbert space embeddings and metrics on probability measures. The Journal of Machine Learning Research, vol. 11, pp. 1517–1561.
- [9] D. J. Sutherland, J. B. Oliva, P. Barnabas, and J. Schneider. (2015) Linear-time Learning on Distributions with Approximate Kernel Embeddings. arXiv:1509.07553, pp. 1–10.
- [10] P. Zhang and Y. Gao. (2015) Matrix Multiplication on High-Density Multi-GPU Architectures: Theoretical and Experimental Investigations Peng. ISC High Performance, vol. 1, pp. 17–30. Springer International Publishing.