Randomized Dimensionality Reduction for k-Means Clustering

Ramil Gizatullin
r.gizatullin@innopolis.ru

***Abstract*— This document provides a description of algorithms and experimental results obtained during the implementation of *"Randomized Dimensionality Reduction for k-Means Clustering"* article. Dimensionality Reduction is a popular technique that overcomes the problem of high dimensionality of data when using k-means clustering, making the latter work computationally efficient. It encompasses the union of two approaches: feature selection and feature extraction. The article mentioned above presents the first provably accurate feature selection method for k-means clustering and, in addition, two feature extraction methods. During the work these proposed algorithms were implemented, as well as known methods based on random projections and singular value decomposition (SVD) and tested on both synthetic data and real-world datasets.**

I. INTRODUCTION

One of the most well-known clustering algorithm is the so-called k-means algorithm or Lloyds method [1]. It is an iterative expectation-maximization type approach that attempts for a given a set of points and a positive integer k corresponding to the number of clusters, split the points into k clusters such that the total sum of the squared Euclidean distances of each point to its nearest cluster center is minimized.

Even though this method is considered as very effective, its performance suffers from the high dimensionality of data, which become a common in recent years. These drawbacks can be overcome by using feature selection and feature extraction techniques. Feature selection selects a (small) subset of the actual features of the data, whereas feature extraction constructs a (small) set of artificial features based on the original features. The original article *"Randomized Dimensionality Reduction for k-Means Clustering"* [2] proposes new approaches to feature selection and feature extraction for k-means clustering described in a section 2.

II. ALGORITHMS DESCRIPTION

For the matrix $\mathbf{A}$ let $||\mathbf{A}||_F$ be the Frobenius norm $\sqrt{\sum_{i,j} \mathbf{A}_{i,j}}$

A. *k-Means clustering problem*

Let $\mathbf{A} \in R^{m \times n}$ be input dataset (representing m data points rows described with respect to n features columns), $k > 0$ - number of clusters. Let $\mathbf{X} \in R^{m \times k}$ be cluster indicator matrix, representing S-clustering of $\mathbf{A}$:

$$\mathbf{X}_{ij} = \begin{cases} 1/\sqrt{s_j} & \mathbf{X}_{(i)} \in cluster\ S_j \\ 0 & else \end{cases}$$

there s_j - number of elements in the cluster S_j. Then the problem of k-Means problem can be formulated as the following problem of searching $\mathbf{X}_{opt}$:

$$\mathbf{X}_{opt} = \arg\min_{\mathbf{X} \in \mathcal{X}} ||\mathbf{A} - \mathbf{X}\mathbf{X}^T\mathbf{A}||_F^2$$

with the optimal value of objective:

$$\mathcal{F}(\mathbf{A}, \mathbf{X}_{opt}) = \min_{\mathbf{X} \in \mathcal{X}} ||\mathbf{A} - \mathbf{X}\mathbf{X}^T\mathbf{A}||_F^2 =$$

$$= ||\mathbf{A} - \mathbf{X}_{opt}\mathbf{X}_{opt}^T\mathbf{A}||_F^2 = F_{opt}$$

An algorithm is called a γ-approximation for the k-Means clustering problem ($\gamma \geq 1$) if it returns the indicator matrix $\mathbf{X}_\gamma$ such that with probability at least $1 - \delta_\gamma$:

$$||\mathbf{A} - \mathbf{X}_\gamma\mathbf{X}_\gamma^T\mathbf{A}||_F^2 \leq \gamma \min_{\mathbf{X} \in \mathcal{X}} ||\mathbf{A} - \mathbf{X}\mathbf{X}^T\mathbf{A}||_F^2 =$$

$$= \gamma\mathcal{F}(\mathbf{A}, \mathbf{X}_{opt}) = \gamma F_{opt}$$

γ is so-called approximation ratio, that will be presented further in form of $c + \varepsilon$

Lets note that one iteration of k-Means algorithm takes $O(mnk)$ time - it has linear dependency of n, so dimensionality reduction can reduce running time significantly, e.g. using approximate algorithm with random projections results in $O(mk^2/\varepsilon^2)$ time

B. *Preliminaries*

Singular value decomposition (**SVD**) is an algorithm, that gets top k right singular vectors of $\mathbf{A}$ ($\mathbf{V} \in R^{n \times k}$) for an $O(mn\min(m, n))$ complexity, so it can be used for the feature extraction algorithm. The latter SVD-type result argues that $O(k/\varepsilon^2)$ dimensions (singular vectors) suffice for an $1 + \varepsilon$ approximation ratio

Approximate Singular Value Decomposition (**FastFrobeniusSVD**) is a method that speed ups usual SVD algorithm with an approximation ratio $2 + \varepsilon$. Its time complexity is $O(mnk/\varepsilon)$ and for $r = k + \frac{k}{\varepsilon}$ it constructs matrix $\mathbf{Z}$ as following:

1) Generate an $n \times r$ standard Gaussian matrix $\mathbf{R}$ whose entries are i.i.d. $\mathcal{N}(0, 1)$ variable
2) $\mathbf{Y} = \mathbf{A}\mathbf{R} \in R^{m \times r}$
3) Orthonormalize the columns of $\mathbf{Y}$ to construct the matrix $\mathbf{Q} \in R^{m \times r}$
4) Let $\mathbf{Z} \in R^{n \times k}$ be the top k right singular vectors of $\mathbf{Q}^T\mathbf{A} \in R^{r \times n}$

Randomized Sampling: let $\mathbf{X} \in R^{n \times k}$ with $n > k$. Then:

1) Compute set of sampling probabilities $p_i = \frac{||\mathbf{X}_{(i)}||_2^2}{||\mathbf{X}||_F^2}$, $\sum_{i=1}^n p_i = 1$

2) For the positive r initialize sampling matrix $\mathbf{\Omega} \in R^{n\times r}$ and rescaling matrix $\mathbf{S} \in R^{r\times r}$ with zeros
3) For $t = 1, \dots, r$ pick an integer i_t from set $\{1, \dots, n\}$ with the probability p_i and set $\mathbf{\Omega}_{i_t t} = 1$ $\mathbf{S}_{tt} = 1/\sqrt{rp_{i_t}}$
4) Return $\mathbf{\Omega}, \mathbf{S}$

C. *Feature Selection with Randomized Sampling*

Time complexity $O(mnk/\varepsilon)$ with an approximation ratio $3 + \varepsilon$

Input: Dataset $\mathbf{A} \in R^{m\times n}$, number of clusters k, and $0 < \varepsilon < 1/3$.

Output: $\mathbf{C} \in R^{m\times r}$ with $r = O(klog(k)/\varepsilon^2)$ rescaled features.

1) Let $\mathbf{Z}$ = FastFrobeniusSVD($\mathbf{A}, k, \varepsilon$), $\mathbf{Z} \in R^{n\times k}$
2) Let $\mathbf{\Omega}, \mathbf{S}$ = RandomizedSampling($\mathbf{Z}, r$), $\mathbf{\Omega} \in R^{n\times r}$, $\mathbf{Z} \in R^{r\times r}$
3) Return $\mathbf{C} = \mathbf{A\Omega S} \in R^{m\times r}$

D. *Feature Extraction with Random Projections*

Time complexity $O(mnk/\varepsilon^2)$ with an approximation ratio $2 + \varepsilon$

Input: Dataset $\mathbf{A} \in R^{m\times n}$, number of clusters k, and $0 < \varepsilon < 1/3$.

Output: $\mathbf{C} \in R^{m\times r}$ with $r = O(k/\varepsilon^2)$ new features.

1) Compute $\mathbf{R} \in R^{n\times r}$ in the following way:

$$\mathbf{R} = \begin{cases} +1/\sqrt{r} & w.p.\ 0.5 \\ -1/\sqrt{r} & w.p.\ 0.5 \end{cases}$$

2) Return $\mathbf{C} = \mathbf{AR} \in R^{m\times r}$

E. *Feature Extraction with Approximate SVD*

Time complexity $O(mnk/\varepsilon)$ with an approximation ratio $2 + \varepsilon$

Input: Dataset $\mathbf{A} \in R^{m\times n}$, number of clusters k, and $0 < \varepsilon < 1$.

Output: $\mathbf{C} \in R^{m\times k}$

1) Let $\mathbf{Z}$ = FastFrobeniusSVD($\mathbf{A}, k, \varepsilon$), $\mathbf{Z} \in R^{n\times k}$
2) Return $\mathbf{C} = \mathbf{AZ} \in R^{m\times k}$

III. EXPERIMENTAL SETUP

The proposed algorithms were implemented in python with the usage of numpy, sklearn and scipy packages. All the experiments were performed on a machine with a dual core 2.9 Ghz processor and 8 GB of RAM.

A. *Dimensionality Reduction Methods*

Given m points described with respect to n features and the number of clusters k, the goal is to select or construct r features on which k-means will be executed. The number of features to be selected or extracted is part of the input as well. We test the performance of the algorithms proposed in the original article for various values of r, and compare them. Proposed methods are summarized below:

1) **Randomized Sampling with Exact SVD (Sampl/SVD)**. This corresponds to the "Feature Selection with Randomized Sampling" algorithm with the modification in the first step of the algorithm there matrix Z is calculated with exact SVD to contain exactly the top k right singular vectors of A
2) **Randomized Sampling with Approximate SVD (Sampl/ApproxSVD)** corresponding to the "Feature Selection with Randomized Sampling" algorithm
3) **Random Projections (RP)**. Here "Feature Extraction with Random Projections" algorithm is used
4) **SVD**. This is "Feature Extraction with Approximate SVD" algorithm with the modification, using the exact SVD algorithm to calculate the matrix Z.
5) **Approximate SVD (ApprSVD)**. This corresponds to "Feature Extraction with Approximate SVD" algorithm

B. *k-means method*

In all subsequent experiments the Lloyds k-means algorithm were used. It was executed as well on the original dataset as on the "reduced" version, containing r features. Since MATLAB implementation used in the original article is slightly differs from the sklearn method, to obtain comparable objective and accuracy results some parameters were changed: we run 1000 iterations with 10 different random initializations and return the best outcome over all repetitions, i.e. we run the following command, kmeans(A, k, init='random', n_init=10, max_iter=1000).

C. *Datasets*

Experiments were performed on a few real-world and synthetic datasets used in the original paper. Synthetic dataset was generated as follows: k = 5 centers were chosen uniformly at random from the 2000-dimensional hypercube of side length 4 as the ground truth centers. Then 1000 points were generated from a normal distribution of variance one, centered at each of the real centers, such that each cluster contains 200 points. This dataset will be referred as Synth.

Two real-world datasets were used, namely USPS and COIL20. The USPS digit dataset [3] contains grayscale pictures of handwritten digits. There are 1100 data points per digit, each has 256 dimensions. The coefficients of the data points have been normalized between 0 and 1. The COIL20 [4] dataset contains 1400 images of 20 objects (the images of each objects were taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images). The size of each image is 128x128 pixels, with 256 grey levels per pixel. Just like in the previous dataset, coefficients of the data points have been normalized.

D. *Evaluation Methodology*

To assess the quality of all methods the normalized objective function $\mathcal{F}$ of the k-means clustering problem were used, i.e.

$$\mathcal{F} = \mathcal{F}/||\mathbf{A}||_F^2$$

In addition, the mis-classification accuracy of the clustering result based on the labels from the input data were computed. Given a data point X_i, let r_i and s_i be the obtained cluster label and the label provided by the data corpus, respectively. Then accuracy is defined as follows:

$$AC = \frac{\sum_i^m \delta(s_i, map(r_i))}{m}$$

where m is the total number of data points and $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and $map(r_i)$ is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data corpus. The most straightforward method for search of the best mapping is to iterate through all $O(k!)$ clusters permutations, which becomes extremely slow for $k > 7$. To solve this problem Kuhn-Munkres [5] algorithm, which is also known as Hungarian algorithm, were used. It allows to solve this problem in $O(k^3)$ time.

Finally, running times (in seconds) of both the dimensionality reduction procedure and the k-means algorithm applied on the low-dimensional projected space is reported. All the reported quantities correspond to the average values of five independent executions.

IV. RESULTS

The results of the experiments can be seen in Figures 1-9. Figures 1-3, 4-6 and 7-9 depict performance of clustering on the synthetic, USPS digit and COIL20 datasets respectively. Experiments were made with relative small number of dimensions of $r = 5, 10, \ldots, 100$. Even using such small values of r we can obtain comparable to the original k-Means algorithm.

At the synthetic dataset even $r = 20$ is enough to get same results as the original k-Means. This happens because our clusters are well-separated. Most of the methods work ten times faster than k-Means, except the SVD, because it has huge time complexity $O(mn \min(m, n))$, and Approximate SVD with time complexity $O(mnr/\varepsilon)$ (Fig 1-3).

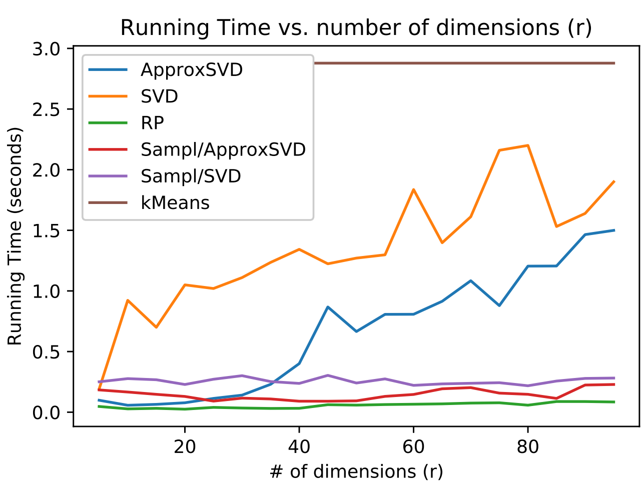

Fig. 1. Synth - Running time

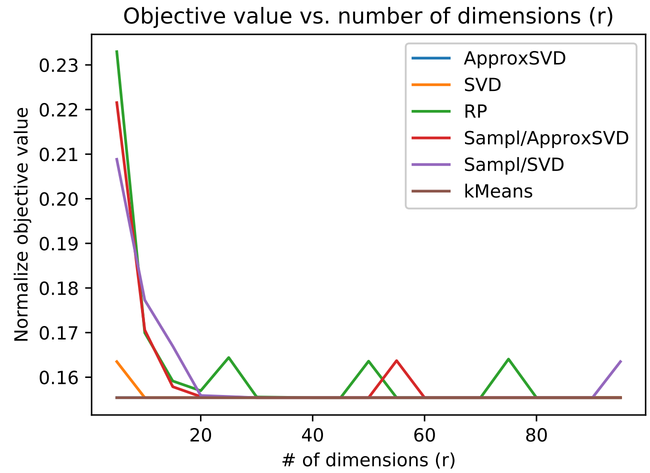

Fig. 2. Synth - Objective

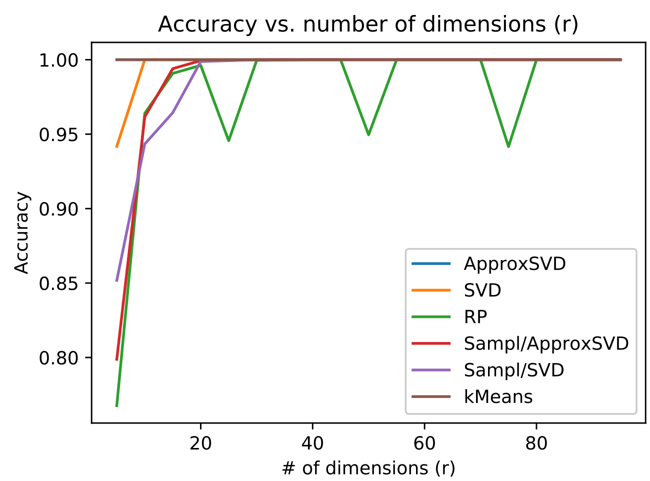

Fig. 3. Synth - Accuracy of clustering

For the USPS situation is slightly different. The ”lightweight” algorithms, such as Randomized Projections, Randomized Sampling with Approximate SVD and Randomized Sampling with Exact SVD require more number of dimensions to perform good, while for SVD and Approximate SVD $r = 20$ is still enough (Fig 4-6).

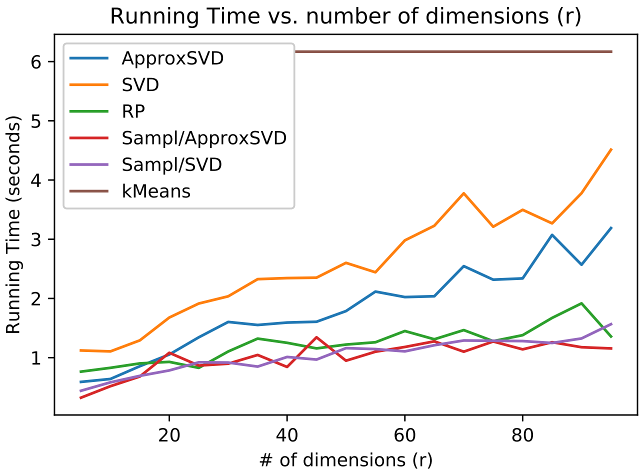

Fig. 4. USPS - Running time

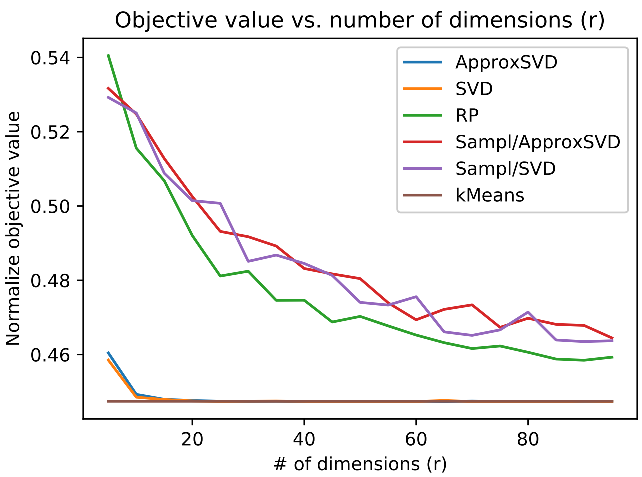

Fig. 5. USPS - Objective

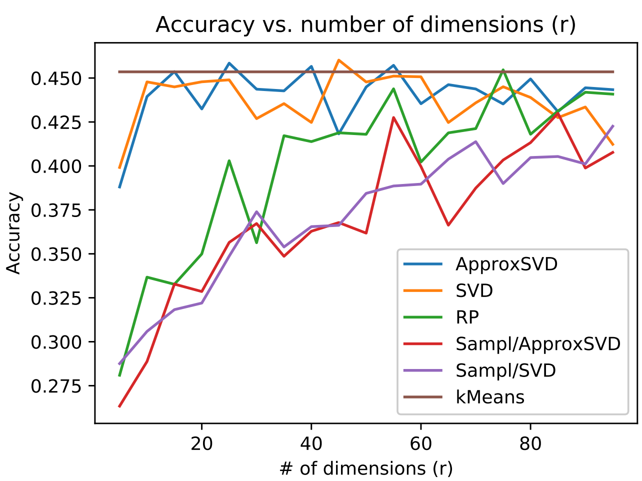

Fig. 6. USPS - Accuracy of clustering

Almost the same situation can be observed on the COIL20 dataset. However, number of dimensions here is $n = 16384$, so SVD and Randomized Sampling with Exact SVD take much time to execute even for $r = 20$ features, while Random Projections and Randomized Sampling with Approximate SVD show the same results for $r = 100$ features, but spend 10 and 5 times less time respectively (Fig 7-9).

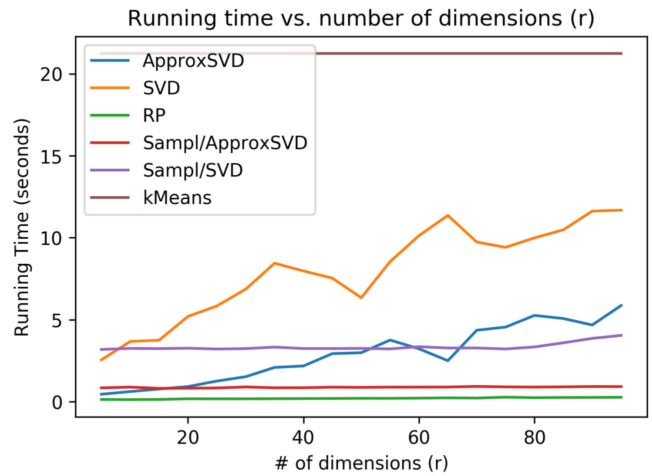

Fig. 7. COIL20 - Running time

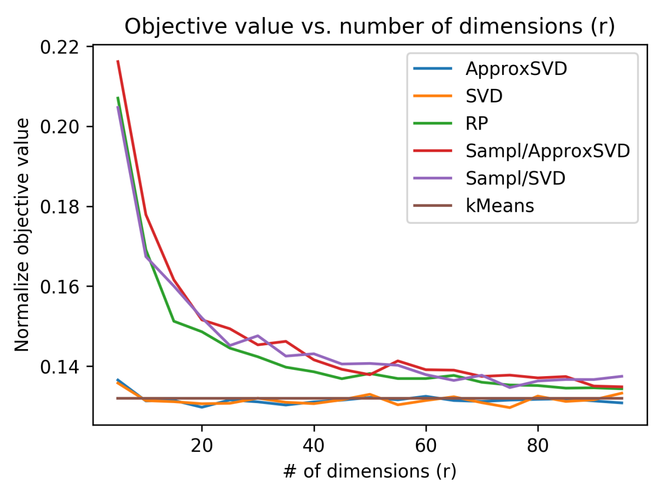

Fig. 8. COIL20 - Objective

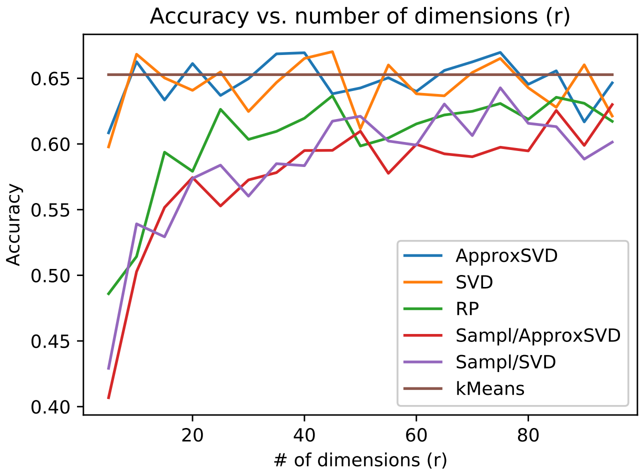

Fig. 9. COIL20 - Accuracy of clustering

The overall results show that the proposed methods perform well. Randomized Sampling with Approximate SVD and Randomized Projections methods demonstrated very high computational speed, slightly depending on r with the comparable results with original k-Means, that makes them indispensable tool when working with very high-dimensional data ($r > 1M$). Approximate SVD behavior is slightly different - its computational time highly depends on r, but its objective value and accuracy tend to converge faster.

V. Conclusion

The original article algorithms were implemented and evaluated on three same datasets, mentioned in the article with the same metrics. Obtained results are close to the original paper result, as well as proposed theoretical bounds. Even though that empirical results are far from exhaustive, they are able to indicate that the feature selection and feature extraction algorithms presented in the original article achieve a satisfactory empirical performance.

References

[1] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[2] C. Boutsidis, A. Zouzias, M. W. Mahoney, and P. Drineas, "Randomized dimensionality reduction for k-means clustering," *IEEE Transactions on Information Theory*, vol. 61, no. 2, pp. 1045–1062, 2015.

[3] J. J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 5, pp. 550–554, 1994.

[4] S. A. Nene, S. K. Nayar, H. Murase *et al.*, "Columbia object image library (coil-20)," 1996.

[5] M. D. Plummer and L. Lovász, *Matching theory*. Elsevier, 1986, vol. 29.