A RANDOMIZED APPROACH TO EFFICIENT KERNEL CLUSTERING

Farhad Pourkamali-Anaraki⋆ *Stephen Becker*†

⋆ Department of Electrical, Computer, and Energy Engineering
† Department of Applied Mathematics
University of Colorado at Boulder, Boulder, CO 80309 USA

ABSTRACT

Kernel-based K-means clustering has gained popularity due to its simplicity and the power of its implicit non-linear representation of the data. A dominant concern is the memory requirement since memory scales as the square of the number of data points. We provide a new analysis of a class of approximate kernel methods that have more modest memory requirements, and propose a specific one-pass randomized kernel approximation followed by standard K-means on the transformed data. The analysis and experiments suggest the method is accurate, while requiring drastically less memory than standard kernel K-means and significantly less memory than Nyström based approximations.

***Index Terms*—** Kernel methods, Unsupervised learning, Low-rank approximation, Randomized algorithm

1. INTRODUCTION

Kernel-based approaches are popular methods for supervised and unsupervised learning [1]. The (i, j) entry of a kernel matrix $\mathbf{K}$ represents the inner product between the representations of data points $\mathbf{x}_i$ and $\mathbf{x}_j$ in a lifted space, and this lifting allows one to use linear techniques in the higher (or infinite) dimensional space which correspond to non-linear techniques in the original space. For example, the two classes of data in Fig. 1 are not linearly separable, but they become linearly separable after applying a suitable kernel (Fig. 2).

With n data points, the kernel matrix is $n \times n$, and for large modern data sets this is infeasible to store or compute with. For this reason, there is a long history of low-rank approximations of $\mathbf{K}$, starting with incomplete Cholesky factorizations in [2], and excellently summarized in [3]. In particular, [3] argues that the design of the approximation must go hand-in-hand with the learning algorithm, and proceeds to analyze the case of kernel regression.

The aim of this paper is to analyze kernel approximations in hard clustering and suggest a specific one-pass randomized kernel approximation. This approximation is based on a one-pass variant of the popular randomized approach described in [4]. Such an approach had previously been used to approximate a small inner $m \times m$ matrix in the Nyström method [5], but with accuracy limited by the large sampling requirements of the Nyström approach. Instead, we precondition the kernel matrix in a streaming manner and *then* sample, which allows us to take drastically fewer samples while maintaining excellent clustering performance (cf. Table 1). Our algorithm is not necessarily faster than the Nyström approach, but has lower memory requirements, such as around 10 times lower memory for both the synthetic data in Table 1 and the real data in Fig. 3. A particular benefit to our proposal is that it consists of a distinct preprocessing phase followed by the standard K-means algorithm on transformed data, thus allowing one to leverage existing algorithm libraries.

1.1. Notation

We denote column vectors with lower-case bold letters and matrices with upper-case bold letters. Let $\|\mathbf{K}\|_F$ and $\|\mathbf{K}\|_2$ denote the Frobenius norm and spectral norm respectively. Moreover, $\|\mathbf{K}\|_* = \mathrm{tr}((\mathbf{K}^T\mathbf{K})^{1/2})$ represents the trace norm, where $\mathrm{tr}(\cdot)$ is the trace operator.

Also, we represent the entry in the i-th row and the j-th column of $\mathbf{K}$ as $[\mathbf{K}]_{ij}$. We let $\mathbf{e}_i$ denote the i-th vector of the canonical basis in $\mathbb{R}^K$, where entries are all zero except for the i-th one which is 1.

2. PRELIMINARIES

2.1. K-means Clustering

Consider a collection of n data samples in $\mathbb{R}^p$, where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p\times n}$ represents the data matrix. The K-means algorithm is a popular hard clustering technique that splits the data set into a known number of K clusters. The resulting K-partition $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$ is a collection of K non-empty pairwise disjoint sets that covers the data set. Moreover, each cluster $\mathcal{S}_k$ is represented using a vector $\boldsymbol{\mu}_k \in \mathbb{R}^p$ that is associated with the k-th cluster.

Hence, the K-means objective is to find the optimal K-partition by minimizing the total sum of the squared Euclidean distances of each data sample to the closest cluster:

$$\mathcal{F}(\mathcal{S}) = \sum_{i=1}^{n}\sum_{k=1}^{K} t_{ik}\|\mathbf{x}_i - \boldsymbol{\mu}_k\|_2^2 \tag{1}$$

where $t_{ik} \in \{0, 1\}$ is a binary indicator variable and $\mathbf{t}_i = [t_{i1}, \ldots, t_{iK}]^T$ is the k-th canonical basis vector in $\mathbb{R}^K$ if and only if $\mathbf{x}_i$ belongs to the k-th cluster $\mathcal{S}_k$.

Minimizing the objective function in (1) is known to be NP-hard, so the standard approach is to look for an approximate solution by an *iterative* method that guarantees convergence only to a local minimum [6]. In the first step, the assignment of data samples is updated with $\{\boldsymbol{\mu}_k\}_{k=1}^K$ held fixed. In the next step, the cluster representatives $\{\boldsymbol{\mu}_k\}_{k=1}^K$ are updated based on the most recent assignment.

K-means clustering works well only if all pairs of clusters are linearly separable, and does not perform well on finding non-linearly

separable clusters of varying densities and distributions [7]. A prominent approach to tackle this problem is to employ a non-linear distance function using the kernel trick [8] in machine learning. The resulting algorithm is the so-called Kernel K-means which we explain in the next section.

2.2. The Kernel K-means Algorithm

Kernel-based methods have provided a straightforward way to deal with non-linear structure in datasets. To be formal, each $\mathbf{x}_i$ is mapped to a higher dimensional feature space using the non-linear mapping $\mathbf{\Phi}, \mathbf{x}_i \mapsto \mathbf{\Phi}(\mathbf{x}_i)$ for i=1, . . . ,n. The kernel trick is based on the observation that many algorithms only need to compute the inner product between data points, and not see the data points themselves. Thus the trick avoids the explicit mapping by allowing one to compute inner products between mapped data points in the feature space using a non-linear similarity measure used in Euclidean space $\mathbb{R}^p$:

$$\langle \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j) \rangle = \kappa(\mathbf{x}_i, \mathbf{x}_j), \quad \forall i, j \in \{1, \ldots, n\} \tag{2}$$

where $\kappa(\cdot, \cdot)$ is a Mercer kernel function such that κ induces a positive semidefinite matrix $[\mathbf{K}]_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ for all input data $\{\mathbf{x}_i\}_{i=1}^n$. Examples of such kernels include polynomial kernels $\kappa(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + \gamma)^d$ and Gaussian radial basis function kernels $\kappa(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|_2^2)$ with parameters $\gamma \in \mathbb{R}^+$ and $d \in \mathbb{N}$ [9, 10].

The Kernel K-means algorithm finds a K-partition of the mapped data $\{\mathbf{\Phi}(\mathbf{x}_i)\}_{i=1}^n$ by minimizing:

$$\mathcal{L}(\mathcal{S}) = \sum_{i=1}^{n} \sum_{k=1}^{K} t_{ik} \|\mathbf{\Phi}(\mathbf{x}_i) - \boldsymbol{\mu}_k\|_2^2. \tag{3}$$

The optimization problem of minimizing (3) can be solved using the same iterative procedure of K-means. To see this, consider the centroid of the j-th cluster $\boldsymbol{\mu}_j = \frac{1}{|\mathcal{S}_j|} \sum_{\mathbf{\Phi}(\mathbf{x}_l) \in \mathcal{S}_j} \mathbf{\Phi}(\mathbf{x}_l)$. This centroid cannot be computed explicitly, but we can compute the distance between each mapped data sample $\mathbf{\Phi}(\mathbf{x}_i)$ and the centroid:

$$\begin{aligned} &\|\mathbf{\Phi}(\mathbf{x}_i) - \boldsymbol{\mu}_j\|_2^2 = \langle \mathbf{\Phi}(\mathbf{x}_i) - \boldsymbol{\mu}_j, \mathbf{\Phi}(\mathbf{x}_i) - \boldsymbol{\mu}_j \rangle \\ &= [\mathbf{K}]_{ii} - \frac{2}{|\mathcal{S}_j|} \sum_{\mathbf{\Phi}(\mathbf{x}_l) \in \mathcal{S}_j} [\mathbf{K}]_{il} + \frac{1}{|\mathcal{S}_j|^2} \sum_{\mathbf{\Phi}(\mathbf{x}_l), \mathbf{\Phi}(\mathbf{x}_l') \in \mathcal{S}_j} [\mathbf{K}]_{ll'}. \end{aligned} \tag{4}$$

Hence, we see that Kernel K-means is an iterative algorithm that requires access to the full kernel matrix $\mathbf{K}$ without the need to explicitly map the data points. However, this clustering technique requires the storage and handling of a large kernel matrix $\mathbf{K}$ in each iteration. Therefore, the quadratic complexity of $\mathcal{O}(n^2)$ to store or $\mathcal{O}(n^2 p)$ to compute the kernel matrix for each iteration makes the Kernel K-means algorithm non-scalable to large data sets.

2.3. Prior Work on Efficient Kernel-Based Learning

Much research has focused on approximating the kernel matrix using a low-rank decomposition; more recent works focus on a sum of low-rank and diagonal or low-rank and sparse decompositions, but similar memory and computation considerations apply. Note that the kernel matrix is a symmetric positive semidefinite matrix. Thus, its eigenvalue decomposition can be used to express a low-rank approximation:

$$\mathbf{K} \approx \mathbf{U}_r \mathbf{\Lambda}_r \mathbf{U}_r^T = \left(\mathbf{U}_r \mathbf{\Lambda}_r^{1/2}\right) \left(\mathbf{\Lambda}_r^{1/2} \mathbf{U}_r^T\right) = \mathbf{Y}^T \mathbf{Y} \tag{5}$$

where $\mathbf{\Lambda}_r \in \mathbb{R}^{r \times r}$, $r < n$, is a diagonal matrix containing the top r eigenvalues of $\mathbf{K}$ in descending order and $\mathbf{U}_r \in \mathbb{R}^{n \times r}$ contains the associated orthonormal eigenvectors. Note that the eigenvalues of the kernel matrix are non-negative since $\mathbf{K}$ is positive semidefinite. The decomposition $\mathbf{K} \approx \mathbf{Y}^T \mathbf{Y}$ in (5) essentially *linearizes* the kernel matrix. Therefore, one can directly work with the new samples $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$ embedded in Euclidean space $\mathbb{R}^r$, while respecting the non-linear similarities in the kernel matrix $\mathbf{K}$. Hence, this technique can be viewed as a preprocessing stage that eliminates the need to store and manipulate the kernel matrix during the learning process. Moreover, it is shown that the eigenvalue decomposition of the kernel matrix can be used to infer the number of clusters [11].

However, direct eigenvalue decomposition of large kernel matrices is often a demanding task that requires $\mathcal{O}(n^2)$ space and $\mathcal{O}(n^3)$ time. The popular Nyström method is an efficient algorithm to find low-rank approximations of large symmetric positive semidefinite matrices. The original Nyström method that was introduced in [12] is based on sampling a small subset of m columns of $\mathbf{K}$ using uniform sampling without replacement. This one-pass algorithm was revisited in [13], where a data-dependent non-uniform sampling distribution was presented that requires at least two passes over the kernel matrix. The variants of the Nyström method have been proposed in the literature to analyze various sampling strategies, including [14, 15, 16]. The recent paper [17] reviews different kinds of Nyström methods for large-scale machine learning.

3. LINEARIZED KERNEL K-MEANS CLUSTERING

In this section, we analyze the quality of Kernel K-means clustering under the low-rank decomposition of the kernel matrix $\mathbf{K} = \mathbf{Y}^T \mathbf{Y}$ given in (5). This low-rank decomposition can be done by using any low-rank approximation technique and we do not make any assumptions in this section. Our analysis compares the optimal solution of Kernel K-means on the new samples $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n] \in \mathbb{R}^{r \times n}$ with the optimal solution of Kernel K-means on $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$. To do this, we first present an alternative formulation of the Kernel K-means objective function in equation (3). This type of analysis that we follow in this paper is commonly used in the literature for K-means clustering, e.g., [18, 19].

To begin, let us consider the matrix of mapped data samples $\mathbf{\Phi}(\mathbf{X}) = [\mathbf{\Phi}(\mathbf{x}_1), \ldots, \mathbf{\Phi}(\mathbf{x}_n)]$. We also define the cluster indicator matrix $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_n] \in \mathbb{R}^{K \times n}$, where each row corresponds to a cluster. Each column of $\mathbf{C}$ represents the cluster membership of $\mathbf{\Phi}(\mathbf{x}_j)$ with only one nonzero entry such that $\mathbf{c}_j = (1/\sqrt{|\mathcal{S}_i|})\mathbf{e}_i$ if and only if $\mathbf{\Phi}(\mathbf{x}_j)$ belongs to the i-th cluster $\mathcal{S}_i$; denote the set of all such indicator matrices by $\mathcal{C}$. Given the matrix $\mathbf{C}$, the matrix product $\mathbf{\Phi}(\mathbf{X})\mathbf{C}^T$ consists of K centroids as columns, where the i-th column is $\sqrt{|\mathcal{S}_i|}\boldsymbol{\mu}_i$. Hence, $\mathbf{\Phi}(\mathbf{X})\mathbf{C}^T\mathbf{c}_j = (1/\sqrt{|\mathcal{S}_i|})(\sqrt{|\mathcal{S}_i|}\boldsymbol{\mu}_i) = \boldsymbol{\mu}_i$ selects the centroid of the i-th cluster $\mathcal{S}_i$ that $\mathbf{\Phi}(\mathbf{x}_j)$ belongs to it. Thus, the Kernel K-means objective function (3) can be written as:

$$\begin{aligned} \mathcal{L}(\mathbf{C}) &= \sum_{j=1}^{n} \|\mathbf{\Phi}(\mathbf{x}_j) - \mathbf{\Phi}(\mathbf{X})\mathbf{C}^T\mathbf{c}_j\|_2^2 = \|\mathbf{\Phi}(\mathbf{X}) - \mathbf{\Phi}(\mathbf{X})\mathbf{C}^T\mathbf{C}\|_F^2 \\ &= \mathrm{tr}\left((\mathbf{I}_{n\times n} - \mathbf{C}^T\mathbf{C})\mathbf{K}(\mathbf{I}_{n\times n} - \mathbf{C}^T\mathbf{C})\right) \end{aligned} \tag{6}$$

where we used $\|\mathbf{A}\|_F^2 = \mathrm{tr}(\mathbf{A}^T\mathbf{A})$.

Next, we present some properties of the matrix $\mathbf{C}$ defined above. Note that under the mild assumption that every cluster has at least one member, the cluster indicator matrix has K orthonormal rows, i.e., $\mathbf{C}\mathbf{C}^T = \mathbf{I}_{K\times K}$. This follows from the normalization in our definition of $\mathbf{C}$ and the fact that Kernel K-means is a hard clustering algorithm. As a result, we get $(\mathbf{C}^T\mathbf{C})^2 = \mathbf{C}^T\mathbf{C}$ which shows that both $\mathbf{C}^T\mathbf{C}$ and $(\mathbf{I}_{n\times n} - \mathbf{C}^T\mathbf{C})$ are projection matrices.

In the following theorem, we characterize the accuracy of Kernel K-means under the low-rank decomposition of the kernel matrix $\mathbf{K}$. In this case, one should replace the kernel matrix $\mathbf{K}$ with its low-rank decomposition $\mathbf{Y}^T\mathbf{Y}$ in the reformulated objective function (6). This process can also be viewed as applying standard K-means on the new samples $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$ in $\mathbb{R}^r$. Before stating the result, we emphasize that even though finding an optimal solution for K-means clustering is computationally difficult (NP-hard), the set of possible K-partitions of a finite number of data samples is finite. Thus, an optimal solution exists regardless of the difficulty in finding the optimal solution.

Theorem 1. *Let $\mathbf{C}^*$ be an optimal solution of Kernel K-means in feature space using the full kernel matrix $\mathbf{K}$:*

$$\mathbf{C}^* \in \arg\min_{\mathbf{C}\in\mathcal{C}} \mathcal{L}(\mathbf{C}) \tag{7}$$

where $\mathcal{L}(\mathbf{C}) = \operatorname{tr}\left((\mathbf{I}_{n\times n} - \mathbf{C}^T\mathbf{C})\mathbf{K}(\mathbf{I}_{n\times n} - \mathbf{C}^T\mathbf{C})\right)$. Moreover, let $\widehat{\mathbf{C}}$ be an optimal solution of the approximate Kernel K-means using an approximation $\widehat{\mathbf{K}} = \mathbf{K} - \mathbf{E}$ with $\widehat{\mathbf{K}} = \mathbf{Y}^T\mathbf{Y}$ positive semidefinite:

$$\widehat{\mathbf{C}} \in \arg\min_{\mathbf{C}\in\mathcal{C}} \operatorname{tr}\left((\mathbf{I}_{n\times n} - \mathbf{C}^T\mathbf{C})\mathbf{Y}^T\mathbf{Y}(\mathbf{I}_{n\times n} - \mathbf{C}^T\mathbf{C})\right). \tag{8}$$

Then, we have:

$$\mathcal{L}(\widehat{\mathbf{C}}) - \mathcal{L}(\mathbf{C}^*) \leq 2\|\mathbf{E}\|_* \tag{9}$$

where $\mathbf{E}$ is the low-rank approximation error of the kernel matrix $\mathbf{K}$. Furthermore, if $\widehat{\mathbf{K}}$ is the best rank r approximation to $\mathbf{K}$, then $\mathbf{E}$ is positive semidefinite and we can improve Eq. (9) *to*

$$\mathcal{L}(\widehat{\mathbf{C}}) - \mathcal{L}(\mathbf{C}^*) \leq \operatorname{tr}(\mathbf{E}). \tag{10}$$

Proof. The proof follows from the properties of the cluster indicator matrix $\mathbf{C}$ and applying Hölder's inequality, i.e., the trace norm and spectral norm are dual. □

This theorem indicates that the optimal objective value under the approximate matrix is not far from the true objective value. The bound is tight to within a small constant, as one can construct adversarial examples in any dimension where $\mathcal{L}(\widehat{\mathbf{C}}) - \mathcal{L}(\mathbf{C}^*) \geq \frac{1}{2}\|\mathbf{E}\|_*$.

4. THE PROPOSED METHOD

In this section, we present an efficient method for Kernel K-means clustering on large-scale data sets. The key component of our method is to use highly efficient and accurate low-rank approximation techniques that require just a *single pass* over the kernel matrix to eliminate the need to store or recompute large kernel matrices. In [4], a class of randomized algorithms were proposed to construct low-rank approximations of large matrices, and tail bounds given on the spectral and Frobenius norm of the error $\mathbf{E} = \mathbf{K} - \widehat{\mathbf{K}}$ (from which one can bound $\|\mathbf{E}\|_*$ as well). We show how these randomized methods can be employed to achieve improved performance compared to the standard Nyström approximation in applications with limited memory resources.

First, we briefly explain the one-pass eigenvalue decomposition algorithm from [4]. Given a symmetric matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$, the first step is to find a good basis for both the column space and the row space of $\mathbf{K}$. To this end, a standard Gaussian random matrix $\mathbf{\Omega} \in \mathbb{R}^{n\times r'}$ whose entries are i.i.d. $\mathcal{N}(0,1)$ variables is generated with $r'=(r+l)$ for some rank r and oversampling l. The oversampling parameter is often used to increase the accuracy of the method. Then,

Algorithm 1 One-Pass Kernel K-means

Input: kernel matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$, rank r, oversampling l, number of clusters K, number of iterations
Output: cluster indicator matrix $\mathbf{C}$

1: $r' \leftarrow r + l$, $\mathbf{R} \in \mathbb{R}^{n\times r'}$: random sampling matrix
2: $\mathbf{W} \in \mathbb{R}^{n\times r'} \leftarrow (\mathbf{R}^T\mathbf{HDK})^T$
3: find an orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{n\times r}$ by QR decomposition or r leading left singular vectors of $\mathbf{W}$
4: solve $\mathbf{B}(\mathbf{Q}^T\mathbf{\Omega}) = (\mathbf{Q}^T\mathbf{W})$
5: $\mathbf{B} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$
6: $\mathbf{Y} = \mathbf{\Sigma}^{1/2}\mathbf{V}^T\mathbf{Q}^T \in \mathbb{R}^{r\times n}$
7: perform standard K-means on $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$ in $\mathbb{R}^r$

$\mathbf{W} = \mathbf{K}\mathbf{\Omega}$ is computed and one finds $\mathbf{Q} \in \mathbb{R}^{n\times r}$ whose columns form an orthonormal basis for the range of $\mathbf{W}$; this can be done by computing the r leading left singular vectors of $\mathbf{W}$ or via the QR decomposition. Therefore, we have $\mathbf{K} \approx \mathbf{Q}(\mathbf{Q}^T\mathbf{K}\mathbf{Q})\mathbf{Q}^T$ for which a naive approach requires one more pass over $\mathbf{K}$ to compute $\mathbf{B} = \mathbf{Q}^T\mathbf{K}\mathbf{Q}$ and find its low-rank decomposition. However, the matrix $\mathbf{B}$ can be computed by solving the equation $\mathbf{B}(\mathbf{Q}^T\mathbf{\Omega}) = (\mathbf{Q}^T\mathbf{W})$ without revisiting $\mathbf{K}$ [4]. Finally, the eigenvalue decomposition of $\mathbf{B} \in \mathbb{R}^{r\times r}$ yields the desired rank r approximation of $\mathbf{K}$.

The major drawback of this approach is the memory and computation burden associated with the Gaussian random matrix $\mathbf{\Omega}$. Computing the matrix $\mathbf{W}$ takes $\mathcal{O}(n^2r')$ time which is quadratic in the number of samples. To address this problem, the Gaussian random matrix is replaced with a much more efficient structured random matrix $\mathbf{\Omega} = \mathbf{DHR}$ [20, 4]. The matrix $\mathbf{D} = \mathbf{D}^T \in \mathbb{R}^{n\times n}$ is a stochastic diagonal matrix whose entries on the main diagonal are random variables drawn uniformly from $\{\pm 1\}$. The matrix $\mathbf{H} = \mathbf{H}^T \in \mathbb{R}^{n\times n}$ is the Hadamard matrix for which matrix-vector multiplication can be implemented in $\mathcal{O}(n\log(n))$ complexity, hence it is inexpensive to multiply and store compared to the Gaussian matrix. The matrix $\mathbf{H}$ is not stored explicitly, and applying $\mathbf{H}$ to a matrix is efficient in parallel and distributed environments; our implementation uses the pthread library and sees a 11 times speedup over the non-parallel version when using 16 threads. The sub-sampling matrix $\mathbf{R} \in \mathbb{R}^{n\times r'}$ consists of r' columns of the identity matrix $\mathbf{I}_{n\times n}$ drawn uniformly random without replacement.

Our proposed efficient Kernel K-means clustering method is presented in Alg. 1. We emphasize that our method requires only one pass over the columns of the kernel matrix $\mathbf{K}$, and that batches of columns of $\mathbf{K}$ can be constructed on-the-fly, so the entire kernel matrix $\mathbf{K}$ is never formed in memory: the minimal memory requirement is $\mathcal{O}(r'n)$. Furthermore, the computation cost in each iteration of K-means is $\mathcal{O}(rnK)$ and the rank r determines the error in the low-rank approximation of the kernel matrix $\mathbf{K}$.

We also compare various aspects of our proposed approach with the Nyström method. The basic idea behind the Nyström method is to sample m columns from the kernel matrix $\mathbf{K}$. It is obvious that the more columns are sampled, the more accurate the resulting low-rank approximation is. The sampling strategy plays an important role in the accuracy of the Nyström method. The basic method originally proposed by [12] is a one-pass algorithm that employs uniform sampling without replacement. Hence, our method in this paper is similar to the standard one-pass Nyström method as they both use uniform sampling without replacement. However, our method takes advantage of the preconditioning transformation of the kernel matrix $\mathbf{K} \mapsto (\mathbf{HD})\mathbf{K}$ before sub-sampling. As shown in [20], the transfor-

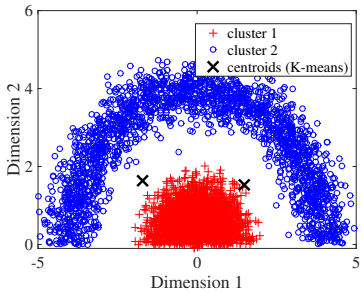
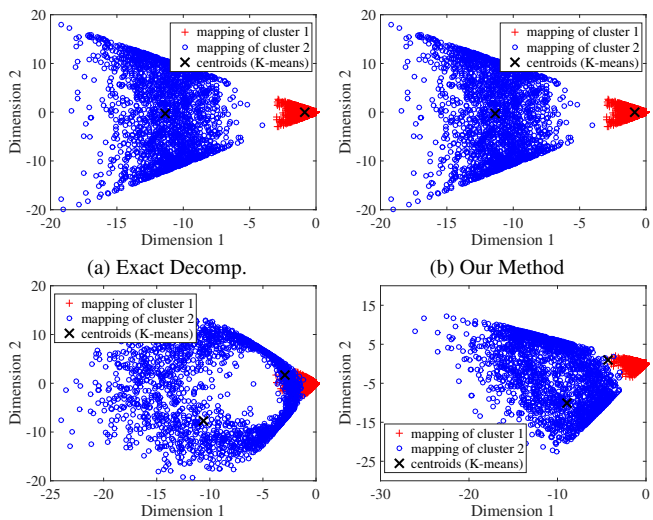


Fig. 1. Original data. The centroids estimated by K-means are shown, which are clearly unhelpful at finding the true clusters.



(a) Exact Decomp. (b) Our Method

(c) Nyström, m=20 (d) Nyström, m=100

Fig. 2. Mapping of the original data using the low-rank approximation $\mathbf{K} \approx \mathbf{Y}^T\mathbf{Y}$. Applying K-means on the mapped data $\mathbf{Y}$ obtained by (a) the exact eigenvalue decomposition, or (b) our method, identifies the two underlying clusters accurately.

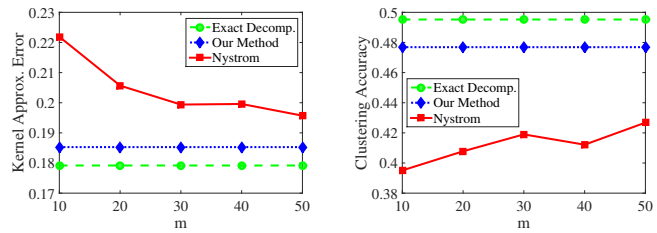
mation $\mathbf{HD}$ equilibrates row norms which eliminates the necessity to use more sophisticated sampling strategies.

To gain some intuition, we consider a synthetic data set shown in Fig. 1. This data set consists of $n = 4000$ samples in $\mathbb{R}^2$ that are non-linearly separable but not linearly separable. Thus, standard K-means is not able to identify these two clusters, and the two centroids selected by standard K-means do not describe the true clusters.

We compute the kernel matrix $\mathbf{K}$ whose elements are obtained by using the polynomial kernel of order $d = 2$, i.e., $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle^2$. The rank parameter is $r = 2$ and our goal is to find $\mathbf{Y} \in \mathbb{R}^{2\times n}$ using the low-rank approximation of the kernel matrix. Our proposed method with the oversampling parameter $l = 10$ is compared with the standard Nyström method with both $m = 20$ and $m = 100$ sampled columns. We show the visualization of the new samples $\{\mathbf{y}_i\}_{i=1}^{4000}$ in Fig. 2. Note that the accuracy of our approach is almost identical to the exact $r = 2$ eigenvalue decomposition of the kernel matrix. Moreover, we see that our method is able to identify these two clusters using the standard K-means on the mapped samples $\mathbf{Y}$. However, the Nyström method does not provide an accurate solution even for a large value of $m = 100 \approx 8r'$. We also compare the normalized kernel approximation error defined as

Table 1. Accuracy of Kernel K-means methods, on data from Fig. 1, $r = 2$. For reference, (non-kernel) K-means has only 0.53 accuracy. Our method takes the equivalent of $m = 12$ columns in the Nyström approach.

Method	Kernel Approx. Error	Clustering Accuracy
Exact Decomposition	0.40	0.99
Our Method	0.40	0.99
Nyström, m=20	0.56	0.74
Nyström, m=100	0.44	0.75



(a) Kernel Approx. Error (b) Clustering Accuracy

Fig. 3. Results for image segmentation data set. Plot of (a) normalized approximation error of the kernel matrix (b) kernel clustering accuracy for varying number of sampled columns m.

$\|\mathbf{K} - \widehat{\mathbf{K}}\|_F / \|\mathbf{K}\|_F$ and the clustering accuracy in Table 1. We see that our method outperforms the Nyström technique on this data set.

Finally, we present the experimental evaluation of our method in Alg. 1 on a real-world data set. Our proposed approach is implemented in MATLAB (with the Hadamard code in C/mex) and compared against the standard Nyström method as well as the exact eigenvalue decomposition. We use the MATLAB `kmeans` function with 10 different initializations, the maximum number of iterations is set to 20 and $r = 2$ is used for low-rank approximations of kernel matrices. Since Nyström and our method are stochastic, we re-run each experiment 100 times and report the average over these trials.

We consider the image segmentation data set that can be downloaded from the UCI Repository. This data set contains $n = 2310$ instances from $K = 7$ outdoor images. Each instance represents a 3×3 region with $p = 19$ attributes that are normalized to unit ℓ_2 norm. Here, we choose the homogeneous polynomial kernel of order $d = 2$, i.e., $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle^2$.

Fig. 3(a) shows the normalized approximation error of the kernel matrix $\|\mathbf{K} - \widehat{\mathbf{K}}\|_F / \|\mathbf{K}\|_F$ for varying number of sampled columns m and fixed oversampling parameter $l = 5$ in our method. As we see, sampling $r' = 2 + 5 = 7$ rows of the preconditioned kernel matrix $(\mathbf{HD})\mathbf{K}$ leads to a more accurate decomposition that sampling $m = 50 \approx 7r'$ columns of the kernel matrix $\mathbf{K}$ in the Nyström method. Moreover, the accuracy of our approach is very close to the optimal exact eigenvalue decomposition.

In Fig. 3(b), the clustering accuracy of our method is compared with the other kernel clustering techniques. Again, we see that our approach has higher accuracy than the the Nyström decomposition approach. In this example, the accuracy of full Kernel K-means ($r = \text{rank}(\mathbf{K})$) is 0.46 and both our method and the approximate Kernel K-means using the exact eigenvalue decomposition (with $r = 2$) have higher accuracy than the full Kernel K-means.

5. CONCLUSIONS

We considered a class of approximate Kernel K-means algorithms in which the kernel matrix is replaced by its low-rank approximation. Our theoretical analysis provides insights into the effect of the approximation on the objective function of Kernel K-means, showing that the optimal objective value under the low-rank approximation is not far from the true objective value. Our theoretical result is applicable to any low-rank approximation technique.

Furthermore, we introduced a specific one-pass randomized algorithm for Kernel K-means. Some benefits of our approach are ease of implementation, tunable accuracy vs. memory/speed tradeoff using the parameter r, and low-memory requirements. The parameter r is typically chosen with cross-validation on a subset of data.

6. REFERENCES

[1] B. Schölkopf and A. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, 2001.

[2] S. Fine and K. Scheinberg, "Efficient SVM training using low-rank kernel representations," *J. Machine Learning Research*, vol. 2, pp. 243–264, 2001.

[3] F. Bach, "Sharp analysis of low-rank kernel matrix approximations," *J. Machine Learning Research, workshop and conf. proceed.*, vol. 30, 2013.

[4] N. Halko, P. Martinsson, and J. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[5] M. Li, J. Kwok, and B. Lü, "Making large-scale Nyström approximation possible," in *27th International Conference on Machine Learning*, 2010, pp. 631–638.

[6] C. Bishop, *Pattern recognition and machine learning*, Springer, 2006.

[7] I. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 551–556.

[8] A. Aizerman, E. Braverman, and L. Rozoner, "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and remote control*, vol. 25, pp. 821–837, 1964.

[9] H. Van Nguyen, V. Patel, N. Nasrabadi, and R. Chellappa, "Design of non-linear kernel dictionaries for object recognition," *IEEE Transactions on Image Processing*, pp. 5123–5135, 2013.

[10] F. Pourkamali-Anaraki and S. Hughes, "Kernel compressive sensing," in *IEEE International Conference on Image Processing*, 2013, pp. 494–498.

[11] M. Girolami, "Mercer kernel-based clustering in feature space," *IEEE Transactions on Neural Networks*, pp. 780–784, 2002.

[12] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proceedings of the 14th Annual Conference on Neural Information Processing Systems*, 2001, pp. 682–688.

[13] P. Drineas and M. Mahoney, "On the Nyström method for approximating a gram matrix for improved kernel-based learning," *The Journal of Machine Learning Research*, pp. 2153–2175, 2005.

[14] S. Kumar, M. Mohri, and A. Talwalkar, "Sampling techniques for the Nyström method," in *International Conference on Artificial Intelligence and Statistics*, 2009, pp. 304–311.

[15] K. Zhang and J. Kwok, "Clustered Nyström method for large scale manifold learning and dimension reduction," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1576–1587, 2010.

[16] S. Si, C. Hsieh, and I. Dhillon, "Memory efficient kernel approximation," in *International Conference on Machine Learning (ICML)*, 2014, pp. 701–709.

[17] S. Sun, J. Zhao, and J. Zhu, "A review of Nyström methods for large-scale machine learning," *Information Fusion*, vol. 26, pp. 36–48, 2015.

[18] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *SODA*, 2007, pp. 1027–1035.

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

[20] J. Tropp, "Improved analysis of the subsampled randomized Hadamard transform," *Advances in Adaptive Data Analysis*, pp. 115–126, 2011.