The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Randomized Clustered Nyström for Large-Scale Kernel Machines

Farhad Pourkamali-Anaraki
University of Colorado Boulder
farhad.pourkamali@colorado.edu

Stephen Becker
University of Colorado Boulder
stephen.becker@colorado.edu

Michael B. Wakin
Colorado School of Mines
mwakin@mines.edu

Abstract

The Nyström method is a popular technique for generating low-rank approximations of kernel matrices that arise in many machine learning problems. The approximation quality of the Nyström method depends crucially on the number of selected landmark points and the selection procedure. In this paper, we introduce a randomized algorithm for generating landmark points that is scalable to large high-dimensional data sets. The proposed method performs K-means clustering on low-dimensional random projections of a data set and thus leads to significant savings for high-dimensional data sets. Our theoretical results characterize the tradeoffs between accuracy and efficiency of the proposed method. Moreover, numerical experiments on classification and regression tasks demonstrate the superior performance and efficiency of our proposed method compared with existing approaches.

Introduction

Kernel methods have been widely used in various learning problems such as classification and regression. Well-known examples include support vector machines (SVM) (Cortes and Vapnik 1995; Suykens and Vandewalle 1999), kernel principal component analysis (KPCA) (Schölkopf, Smola, and Müller 1998), and kernel ridge regression (KRR) (Saunders, Gammerman, and Vovk 1998). The main idea behind kernel-based learning is to map the input data points into a feature space, where all pairwise inner products can be computed via a nonlinear kernel function that satisfies Mercer's condition (Schölkopf and Smola 2001). The lifted representation of the input data points may lead to better performance on learning problems (Yan and Sarkar 2016).

To be formal, let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$ be a data matrix that contains n data points in $\mathbb{R}^p$ as its columns. The inner products in feature space are calculated using a nonlinear kernel function $\kappa(\cdot, \cdot)$:

$$K_{ij} \overset{\text{def}}{=} \kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle, \quad i, j = 1, \ldots, n, \quad (1)$$

where $\Phi : \mathbf{x} \mapsto \Phi(\mathbf{x})$ is the kernel-induced feature map. A popular choice is the Gaussian kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / c\right)$, with the parameter $c > 0$. The pairwise inner products are stored in the symmetric positive semidefinite (SPSD) *kernel matrix* $\mathbf{K} \in \mathbb{R}^{n \times n}$. However, it takes $\mathcal{O}(n^2)$ memory to store the full kernel matrix and subsequent processing of the kernel matrix within the learning process is often computationally quite expensive.

A well-studied approach to tackle these challenges is to use a *low-rank approximation* of the kernel matrix, where the best rank-r approximation $\mathbf{K}_{(r)} = \mathbf{U}_r \mathbf{\Lambda}_r \mathbf{U}_r^T$ is computed via the eigenvalue decomposition (EVD) for $r \leq \text{rank}(\mathbf{K})$. The diagonal matrix $\mathbf{\Lambda}_r \in \mathbb{R}^{r \times r}$ contains the r leading eigenvalues, and the columns of $\mathbf{U}_r \in \mathbb{R}^{n \times r}$ span the top r-dimensional eigenspace of $\mathbf{K}$. Since $\mathbf{K}$ is SPSD, we have a low-rank approximation in the form of:

$$\mathbf{K} \approx \mathbf{K}_{(r)} = \mathbf{L}\mathbf{L}^T, \;\; \mathbf{L} = \mathbf{U}_r \mathbf{\Lambda}_r^{1/2} \in \mathbb{R}^{n \times r}. \qquad (2)$$

When the target rank r is small and chosen independently of n, the benefits of this low-rank approximation are twofold. First, the complexity of storing the matrix $\mathbf{L}$ is $\mathcal{O}(nr)$, which is only linear in the data set size n. The reduction of memory requirements from quadratic to linear results in significant memory savings. Second, the low-rank approximation leads to substantial computational savings within the learning process. In this paper, we focus on two important problems, namely KPCA and KRR, and explain the benefits of the low-rank approximation of the kernel matrix.

Problem 1: KPCA for feature extraction A key component of many learning tasks is the *preprocessing* step in which a concise set of low-dimensional features are constructed to facilitate their analysis. For example, let us consider a classification task with n training data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and a testing data point $\mathbf{x}$ in $\mathbb{R}^p$. To extract features via KPCA, the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ of training data and its best rank-r approximation are computed according to (2), i.e., $\mathbf{K} \approx \mathbf{K}_{(r)} = \mathbf{L}\mathbf{L}^T$. Let $\widehat{\Phi}(\mathbf{x}_j) \in \mathbb{R}^r$ denote the j-th column of $\mathbf{L}^T \in \mathbb{R}^{r \times n}$ and observe that the low-rank approximation *linearizes* $\mathbf{K}$ because $K_{ij} \approx \langle \widehat{\Phi}(\mathbf{x}_i), \widehat{\Phi}(\mathbf{x}_j) \rangle$. Thus, $\widehat{\Phi}(\mathbf{x}_j)$ can be viewed as a mapping of the input data $\mathbf{x}_j$ to an r-dimensional feature space (Zhang et al. 2012; Pourkamali-Anaraki and Becker 2016):

$$\begin{aligned}\widehat{\Phi}(\mathbf{x}_j) =& \mathbf{\Lambda}_r^{1/2} \mathbf{U}_r^T \mathbf{e}_j = \mathbf{\Lambda}_r^{-1/2} \mathbf{U}_r^T \mathbf{K} \mathbf{e}_j \\ =& \mathbf{\Lambda}_r^{-1/2} \mathbf{U}_r^T [\kappa(\mathbf{x}_1, \mathbf{x}_j), \kappa(\mathbf{x}_2, \mathbf{x}_j), \ldots, \kappa(\mathbf{x}_n, \mathbf{x}_j)]^T,\end{aligned} \quad (3)$$

where $\mathbf{e}_j \in \mathbb{R}^n$ is the j-th vector of the canonical basis and we used $\mathbf{U}_r^T \mathbf{K} = \mathbf{\Lambda}_r \mathbf{U}_r^T$. Similarly, we can extract an r-dimensional feature vector for the testing data $\mathbf{x}$ by replacing $\mathbf{x}_j$ with $\mathbf{x}$ in (3). After the preprocessing step, one can employ any off-the-shelf classification method, such as K-nearest neighbors, on the constructed features in $\mathbb{R}^r$.

Problem 2: KRR Consider a set of instance-label pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$. KRR performs linear ridge regression on $\{(\Phi(\mathbf{x}_i), y_i)\}_{i=1}^n$, cf. (1). Since the explicit form of Φ is not known, KRR proceeds by generating $\boldsymbol{\alpha}^*$ which solves the following dual optimization problem (Saunders, Gammerman, and Vovk 1998):

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} + \lambda \boldsymbol{\alpha}^T \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T \mathbf{y}, \tag{4}$$

where $\mathbf{K} \in \mathbb{R}^{n \times n}$ is the kernel matrix with $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{y} = [y_1, \ldots, y_n]^T \in \mathbb{R}^n$ is the response vector, and $\lambda > 0$ is the regularization parameter. In the *prediction* stage, the response value for a testing data point $\mathbf{x}$ is computed as $\sum_{i=1}^n \boldsymbol{\alpha}_i^* \kappa(\mathbf{x}_i, \mathbf{x})$. The problem in (4) admits the closed-form solution $\boldsymbol{\alpha}^* = (\mathbf{K} + \lambda \mathbf{I}_{n \times n})^{-1} \mathbf{y}$, where it costs $\mathcal{O}(n^3)$ to compute the matrix inversion. To reduce this cost, one can use the low-rank approximation in (2) and the Sherman-Morrison-Woodbury formula to find an approximate solution $\widehat{\boldsymbol{\alpha}} \approx \boldsymbol{\alpha}^*$ (Cortes, Mohri, and Talwalkar 2010):

$$\begin{aligned} \widehat{\boldsymbol{\alpha}} &= \left(\mathbf{K}_{(r)} + \lambda \mathbf{I}_{n \times n}\right)^{-1} \mathbf{y} \\ &= \lambda^{-1} \left(\mathbf{I}_{n \times n} - \mathbf{L} \left(\mathbf{L}^T \mathbf{L} + \lambda \mathbf{I}_{r \times r} \right)^{-1} \mathbf{L}^T \right) \mathbf{y}. \end{aligned} \tag{5}$$

Here, one needs only to invert a much smaller matrix of size $r \times r$. The computational cost of $\mathbf{L}^T \mathbf{L}$ is $\mathcal{O}(nr^2)$ and the cost of matrix inversion is $\mathcal{O}(r^3)$. Thus, the computational cost is noticeably reduced.

Although the low-rank approximation of $\mathbf{K}$ is a promising approach to trade-off accuracy for scalability, an eigenvalue decomposition has at least quadratic time complexity and takes $\mathcal{O}(n^2)$ space. To address this issue, one line of prior work is centered around *approximating* the best rank-r approximation, but assumes ready access to $\mathbf{K}$; see (Halko, Martinsson, and Tropp 2011) for a survey. However, $\mathbf{K}$ is typically unknown in kernel methods and the cost to form $\mathbf{K}$ using using standard kernel functions is $\mathcal{O}(pn^2)$, which is extremely expensive for large high-dimensional data sets.

For this reason, the Nyström method (Williams and Seeger 2001) has been a popular technique to compute a low-rank approximation of kernel matrices. The Nyström method works by selecting a small set of points referred to as *landmark points*, and computes the kernel similarities between the input data points and landmark points. Hence, the performance of the Nyström method depends crucially on the number of selected landmark points as well as the procedure according to which these landmark points are selected (Kumar, Mohri, and Talwalkar 2012; Sun, Zhao, and Zhu 2015).

Contributions In this paper, we present an efficient method for landmark selection in the Nyström method that scales well to large *high-dimensional* data sets. The proposed method generates a set of landmark points based on low-dimensional random projections of the data. Our theoretical results characterize the *tradeoffs* between the accuracy of the low-rank approximations and the memory/computational savings. Specifically, for a fixed accuracy level, we show that the dimension of the projected data is *independent* of the ambient dimension. Extensive numerical experiments are provided to demonstrate the performance and efficiency of our method on three tasks: (1) low-rank approximation of kernel matrices, (2) classification using KPCA, and (3) KRR. We also examine the out-of-sample extension problem for classification and regression on two data sets with dimensionality up to $p = 150{,}360$. It is observed that our method generates low-rank approximations that are as accurate as the ones obtained by the exact eigenvalue decomposition (i.e., the best rank-r approximation), but runs in roughly the same amount of time as the simplest landmark selection technique (i.e., uniform sampling). Furthermore, we empirically investigate a variant of our method that is suitable for scenarios when only *one pass* over the input data is allowed.

Notation We denote column vectors with lower-case bold letters and matrices with upper-case bold letters; $\mathbf{I}_{n \times n}$ is the identity matrix of size $n \times n$. The Frobenius norm for a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ is $\|\mathbf{A}\|_F = (\sum_{i=1}^n \sum_{j=1}^m A_{ij}^2)^{1/2}$, where A_{ij} represents the (i,j)-th entry of $\mathbf{A}$. The Moore-Penrose pseudo-inverse of $\mathbf{A}$ is denoted by $\mathbf{A}^\dagger$.

Background and Related Work

In this section, we explain how the Nyström method generates the rank-r approximation of kernel matrices. Also, a few related landmark selection techniques are discussed.

The Nyström Method

Consider a set of input data points $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$ and let $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_m] \in \mathbb{R}^{p \times m}$ be a set of m landmark points in $\mathbb{R}^p$, often but not always chosen from among the columns of $\mathbf{X}$, as we will discuss later. The Nyström method first constructs two matrices $\mathbf{C} \in \mathbb{R}^{n \times m}$ and $\mathbf{W} \in \mathbb{R}^{m \times m}$, where $C_{ij} = \kappa(\mathbf{x}_i, \mathbf{z}_j)$ and $W_{ij} = \kappa(\mathbf{z}_i, \mathbf{z}_j)$. Next, it uses both $\mathbf{C}$ and $\mathbf{W}$ to construct an approximation of the kernel matrix as $\mathbf{K} \approx \mathbf{G} = \mathbf{C}\mathbf{W}^\dagger\mathbf{C}^T$, where $\mathbf{G}$ has rank at most m. Although the final goal is to find an approximation that has rank no greater than r, it is often preferred to select $n \gg m > r$ landmark points and then restrict the resultant approximation to have rank at most r. The intuition is that selecting $m > r$ landmark points for the target rank r and then restricting the approximation to a lower rank-r space has a regularization effect, which leads to an improved rank-r approximation. We will thoroughly examine this observation in our numerical experiments.

In order to restrict the rank of $\mathbf{G}$ from m to r and find approximate eigenvalues/eigenvectors of $\mathbf{K}$ in linear time with respect to n, one can compute the *thin* QR decomposition of $\mathbf{C}$; $\mathbf{C} = \mathbf{Q}\mathbf{R}$, where $\mathbf{Q} \in \mathbb{R}^{n \times m}$ has m orthonormal columns and $\mathbf{R} \in \mathbb{R}^{m \times m}$ is an upper triangular matrix. Then, the eigenvalue decomposition of the $m \times m$ matrix $\mathbf{R}\mathbf{W}^{\dagger}\mathbf{R}^T$ is computed, $\mathbf{R}\mathbf{W}^{\dagger}\mathbf{R}^T = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$, where the diagonal matrix $\mathbf{\Sigma} \in \mathbb{R}^{m \times m}$ contains m eigenvalues in descending order, and the columns of $\mathbf{V} \in \mathbb{R}^{m \times m}$ are the eigenvectors. Thus, we get:

$$\mathbf{G} = \mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^T = \mathbf{Q}\left(\mathbf{R}\mathbf{W}^{\dagger}\mathbf{R}^T\right)\mathbf{Q}^T = \left(\mathbf{Q}\mathbf{V}\right)\mathbf{\Sigma}\left(\mathbf{Q}\mathbf{V}\right)^T. \tag{6}$$

Since $\mathbf{Q}$ and $\mathbf{V}$ have orthonormal columns, $\mathbf{Q}\mathbf{V} \in \mathbb{R}^{n \times m}$ contains m orthonormal eigenvectors of $\mathbf{G}$. As a result, the rank-r approximation of $\mathbf{G}$ can be computed using the r leading eigenvalues $\mathbf{\Sigma}_r \in \mathbb{R}^{r \times r}$ and the corresponding eigenvectors $\mathbf{Q}\mathbf{V}_r \in \mathbb{R}^{n \times r}$, where $\mathbf{V}_r$ contains the first r columns of $\mathbf{V}$. This means that the estimates of the top r eigenvalues and eigenvectors of the kernel matrix $\mathbf{K}$ from the Nyström approximation $\mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^T$ are $\widehat{\mathbf{U}}_r = \mathbf{Q}\mathbf{V}_r$ and $\widehat{\mathbf{\Lambda}}_r = \mathbf{\Sigma}_r$ (Pourkamali-Anaraki and Becker 2017a; Tropp et al. 2017; Wang, Gittens, and Mahoney 2017).

Landmark Selection Techniques

The importance of landmark points in the Nyström method has driven much recent work into various probabilistic and deterministic selection techniques in order to improve the accuracy of Nyström-based approximations; see (Sun, Zhao, and Zhu 2015) for a comprehensive survey.

The simplest and most common selection method is uniform sampling without replacement (Williams and Seeger 2001). In this case, each data point is sampled with the same probability, i.e., $p_i = \frac{1}{n}$, for $i = 1, \ldots, n$. The advantage of this technique is the low computational complexity associated with sampling landmark points. However, it has been shown that uniform sampling does not take into account the nonuniform structure of many data sets (Bach 2013). Therefore, sampling mechanisms based on nonuniform distributions have been proposed to address this problem. In this line of work, a popular technique is sampling landmark points with respect to statistical leverage scores, which requires performing (approximate) EVD of the kernel matrix $\mathbf{K}$ (Gittens and Mahoney 2016). Moreover, an alternative nonuniform sampling based on Determinantal Point Processes (DPP) was introduced in (Li, Jegelka, and Sra 2016). However, these landmark selection techniques require computing and storing the entire kernel matrix $\mathbf{K}$, which negates one of the principal benefits of the Nyström method.

Recently, a nonuniform sampling technique motivated by KRR was proposed (Alaoui and Mahoney 2015). This method uses what are known as the λ-ridge leverage scores, for a ridge regression problem on $\mathbf{K}$ with regularization λ. In this case, m data points are sampled with probabilities proportional to the diagonal entries of $\mathbf{K}(\mathbf{K} + \lambda\mathbf{I}_{n \times n})^{-1}$. The exact computation of this quantity is as expensive as solving the original KRR problem, thus a large body of theoretical work computes approximate λ-ridge leverage scores, including (Calandriello, Lazaric, and Valko 2016; 2017).

The Clustered Nyström method (Zhang, Tsang, and Kwok 2008; Zhang and Kwok 2010) is a non-probabilistic approach that uses out-of-sample extensions to select informative landmark points. The key observation of their work is that the Nyström approximation error depends on the *quantization error* of encoding the entire data set with the landmark points. In the following, we restate the main result of Clustered Nyström on the approximation error in terms of the Frobenius norm.

Proposition 1 (Clustered Nyström (Zhang and Kwok 2010))**.** *Assume that κ satisfies the following property:*

$$\left(\kappa(\mathbf{a},\mathbf{b}) - \kappa(\mathbf{c},\mathbf{d})\right)^2 \leq \eta\left(\|\mathbf{a}-\mathbf{c}\|_2^2 + \|\mathbf{b}-\mathbf{d}\|_2^2\right), \tag{7}$$

for $\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d} \in \mathbb{R}^p$ and η is a constant depending on κ. Consider the data set $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$ and the landmark set $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_m] \in \mathbb{R}^{p \times m}$ which partitions $\mathbf{X}$ into m clusters $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$. Let $\mu(\mathbf{x}_i)$ denote the closest landmark point to each data point $\mathbf{x}_i$, i.e., $\mu(\mathbf{x}_i) = \arg\min_{\mathbf{z}_j \in \{\mathbf{z}_1,\ldots,\mathbf{z}_m\}} \|\mathbf{x}_i - \mathbf{z}_j\|_2$. The Nyström approximation error is upper bounded:

$$\|\mathbf{K} - \mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^T\|_F \leq \eta_1\sqrt{E(\mathbf{X},\mathcal{S})} + \eta_2 E(\mathbf{X},\mathcal{S}), \tag{8}$$

where η_1 and η_2 are two constants and $E(\mathbf{X},\mathcal{S})$ is the total quantization error of encoding each data point $\mathbf{x}_i$ with the closest landmark point $\mu(\mathbf{x}_i)$, i.e., $E(\mathbf{X},\mathcal{S}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \mu(\mathbf{x}_i)\|_2^2$.

It is shown that for a number of widely used kernel functions, e.g., Gaussian kernels, the property in (7) is satisfied (Zhang and Kwok 2010). Based on Proposition 1, the Clustered Nyström method tries to minimize the total quantization error—and thus the Nyström approximation error—by applying the K-means clustering algorithm to the n input data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$. In K-means clustering (Bishop 2006), the input data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are partitioned into a collection $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ of m disjoint and nonempty sets (each representing a cluster) such that their union covers the entire data set. The resulting m *cluster centroids* are then chosen as the landmark points to generate the low-rank approximation $\mathbf{G} = \mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^T$. One benefit of the approach is that the full kernel matrix $\mathbf{K}$ is *never* formed.

Randomized Clustered Nyström

The Clustered Nyström method has been shown to be a powerful technique for generating highly accurate low-rank approximations compared to uniform sampling and other sampling methods (Kumar, Mohri, and Talwalkar 2012). However, the main drawbacks of this method are the high memory and computational complexities associated with performing K-means clustering on large high-dimensional data sets.

To introduce our proposed method, we begin by explaining the process of generating landmark points in the Clustered Nyström method. As mentioned in the previous section, the Nyström approximation error depends on the total quantization error of encoding each data point with the closest landmark point. Thus, landmark points are chosen to be centroids resulting from the K-means clustering algorithm. Given an initial set of m centroids $\{\boldsymbol{\mu}_j\}_{j=1}^{m} \in \mathbb{R}^p$, the K-means clustering algorithm iteratively updates assignments and cluster centroids as follows:

1. Update assignments for $i = 1, \ldots, n$: $\mathbf{x}_i \in \mathcal{S}_j \Leftrightarrow j \in \arg\min_{j' \in \{1,\ldots,m\}} \|\mathbf{x}_i - \boldsymbol{\mu}_{j'}\|_2$

2. Update cluster centroids for $j = 1, \ldots, m$: $\boldsymbol{\mu}_j = \frac{1}{|\mathcal{S}_j|} \sum_{\mathbf{x}_i \in \mathcal{S}_j} \mathbf{x}_i$

where $|\mathcal{S}_j|$ denotes the number of data points in $\mathcal{S}_j$ and $\boldsymbol{\mu}_j$ is the sample mean of the j-th cluster.

For large high-dimensional data sets, the memory requirements and computational cost of performing K-means clustering become expensive (Pourkamali-Anaraki and Becker 2017b). First, the K-means algorithm requires several passes on the entire data set and thus the data should often be stored in a *centralized* location which takes $\mathcal{O}(pn)$ memory. Second, the time complexity of K-means clustering is $\mathcal{O}(pnm)$ *per iteration* to partition n data points into m clusters and typically at least 10 (if not 50 or 100) iterations are needed. Hence, the high dimensionality of massive data sets presents a considerable challenge to the design of efficient alternatives for the Clustered Nyström method.

Our strategy builds on recent work in random projections (Achlioptas 2003; Mahoney 2011; Woodruff 2014) to construct a new set of data with compressed features. For some parameter $p' < p$, the data matrix $\mathbf{X}$ is multiplied on the left by a random zero-mean matrix $\mathbf{H} \in \mathbb{R}^{p' \times p}$:

$$\widehat{\mathbf{X}} = \mathbf{H}\mathbf{X} = [\mathbf{H}\mathbf{x}_1, \ldots, \mathbf{H}\mathbf{x}_n] \in \mathbb{R}^{p' \times n}, \;\; p' < p. \qquad (9)$$

The columns of $\widehat{\mathbf{X}} = [\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_n]$ are known as sketches and the random map $\mathbf{H}$ preserves the geometry of the data under certain conditions (Tropp 2011). The task of clustering is then performed on these low-dimensional data points by minimizing $E(\widehat{\mathbf{X}}, \mathcal{S}) = \sum_{i=1}^{n} \|\widehat{\mathbf{x}}_i - \mu(\widehat{\mathbf{x}}_i)\|_2^2$, which partitions the data points in the reduced space into m clusters. After finding the partition in the reduced space, the same partition is used on the original data points to compute cluster centroids in $\mathbb{R}^p$.

The Proposed Method

In this paper, we introduce a random-projection-type Clustered Nyström method, called Randomized Clustered Nyström, for generating landmark points. In the first step, a *random sign* matrix $\mathbf{H} \in \mathbb{R}^{p' \times p}$ whose entries are independent realizations of $\{\pm 1/\sqrt{p'}\}$ Bernoulli random variables is constructed:

$$H_{ij} = \begin{cases} +1/\sqrt{p'} & \text{with probability } 1/2, \\ -1/\sqrt{p'} & \text{with probability } 1/2. \end{cases} \qquad (10)$$

Next, $\mathbf{H}\mathbf{X}$ is computed to find the sketches $\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_n$ in $\mathbb{R}^{p'}$. The standard implementation of matrix multiplication costs $\mathcal{O}(p'pn)$. The matrix multiplication can also be performed in parallel which leads to noticeable accelerations in practice (Halko, Martinsson, and Tropp 2011). Moreover, it is possible to use the mailman algorithm (Liberty and Zucker 2009) which takes advantage of the binary nature of $\mathbf{H}$ to further speed up the matrix multiplication. In our experiments, we use Intel MKL BLAS version 11.2.3 which is bundled with MATLAB, which we found to be sufficiently optimized and to not form a bottleneck in the computational cost, and for this reason we did not pursue asymptotically faster sketches such as the Hadamard transform (Ailon and Chazelle 2009).

Algorithm 1 Randomized Clustered Nyström

```
Input: data set X, number of landmark points $m$, sketching dimension $p' < p$
Output: landmark points Z
Generate a random sign matrix $\mathbf{H} \in \mathbb{R}^{p' \times p}$ as in (10)
Compute $\widehat{\mathbf{X}} = \mathbf{H}\mathbf{X} \in \mathbb{R}^{p' \times n}$
Perform K-means clustering on $\widehat{\mathbf{X}} = [\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_n]$ to get $\widehat{\mathcal{S}}^{opt}$
Compute the sample mean in the original space, cf. (11)
$\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_m] \in \mathbb{R}^{p \times m}$
```

In the second step, the K-means clustering algorithm is performed on $\widehat{\mathbf{X}} = [\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_n]$ to partition the data, i.e., $\widehat{\mathcal{S}}^{opt} \approx \arg\min_{\mathcal{S}} E(\widehat{\mathbf{X}}, \mathcal{S})$, where $\widehat{\mathcal{S}}^{opt} = \{\widehat{\mathcal{S}}_1^{opt}, \ldots, \widehat{\mathcal{S}}_m^{opt}\}$ is the resulting m-partition. We cannot guarantee that K-means returns the globally optimal partition as the problem is NP-hard (Dasgupta 2008) but seeding using K-means++ (Arthur and Vassilvitskii 2007) guarantees a partition with expected objective within a $\log(m)$ factor of the optimal one, and other variants of K-means, under mild assumptions (Ostrovsky et al. 2012), can either efficiently guarantee a solution within a constant factor of optimal, or guarantee solutions arbitrarily close to optimal, so-called polynomial-time approximation schemes (PTAS). Lastly, the landmark points are generated as:

$$\mathbf{z}_j = \frac{1}{|\widehat{\mathcal{S}}_j^{opt}|} \sum_{\mathbf{x}_i \in \widehat{\mathcal{S}}_j^{opt}} \mathbf{x}_i, \;\; j = 1, \ldots, m. \qquad (11)$$

The proposed method is summarized in Algorithm 1. Let us define the compression factor γ as the ratio of parameter p' to the ambient dimension p. Regarding the memory complexity, our method requires only two passes on the original data set, the first to compute the low-dimensional sketches, and the second to compute the sample means. In fact, our Randomized Clustered Nyström only stores the low-dimensional sketches which takes $\mathcal{O}(p'n)$ space, whereas Clustered Nyström has memory complexity of $\mathcal{O}(pn)$, meaning our method reduces the memory complexity by a factor of $1/\gamma$. In terms of time complexity, the computation cost of K-means on the dimension-reduced data in our method is $\mathcal{O}(p'nm)$ per iteration compared to the cost $\mathcal{O}(pnm)$ in the Clustered Nyström method, so the speedup is up to $1/\gamma$ (the exact amount depends on the number of iterations, since we must amortize the cost of the one-time matrix multiply $\mathbf{H}\mathbf{X}$).

Theoretical Guarantees

The following theorem presents an error bound on the Nyström approximation for a set of landmark points generated via our Randomized Clustered Nyström method.

Theorem 1 (Randomized Clustered Nyström)**.** *Assume that the kernel function* κ *satisfies* (7). *Consider the data set* $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$ *and* $\mathbf{K} \in \mathbb{R}^{n \times n}$ *with entries* $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$*. The optimal partitioning of* $\mathbf{X}$ *into* m

clusters is denoted by $\mathcal{S}^{opt}$, i.e., $\mathcal{S}^{opt} = \arg\min_{\mathcal{S}} E(\mathbf{X}, \mathcal{S})$, where $E(\mathbf{X}, \mathcal{S}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \mu(\mathbf{x}_i)\|_2^2$.

Let us generate a random sign matrix $\mathbf{H} \in \mathbb{R}^{p' \times p}$ as in (10) *with $p' = \mathcal{O}(m/\varepsilon^2)$ for some parameter $\varepsilon \in (0, 1/3)$. The Randomized Clustered Nyström method computes $\widehat{\mathbf{X}} = \mathbf{H}\mathbf{X}$ to generate a set of m landmark points $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_m]$ by partitioning of $\widehat{\mathbf{X}} \in \mathbb{R}^{p' \times n}$ into m clusters. We assume that the partitioning $\widehat{\mathcal{S}}^{opt}$ of $\widehat{\mathbf{X}}$ leads to $E(\widehat{\mathbf{X}}, \widehat{\mathcal{S}}^{opt})$ within a constant factor of the optimal value, cf. (Arthur and Vassilvitskii 2007; Ostrovsky et al. 2012). Given $\mathbf{C}$ and $\mathbf{W}$ whose entries are $C_{ij} = \kappa(\mathbf{x}_i, \mathbf{z}_j)$ and $W_{ij} = \kappa(\mathbf{z}_i, \mathbf{z}_j)$, the Nyström approximation error is bounded with probability at least* 0.96 *over the randomness of* $\mathbf{H}$*:*

$$\begin{aligned} \mathcal{E} \overset{def}{=} & \|\mathbf{K} - \mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^T\|_F \\ \leq & \eta_1 \sqrt{(2+\varepsilon) E(\mathbf{X}, \mathcal{S}^{opt})} + \eta_2 (2+\varepsilon) E\left(\mathbf{X}, \mathcal{S}^{opt}\right), \end{aligned} \tag{12}$$

where η_1 and η_2 are two positive constants.

Proof. Based on Proposition 1, we get the following approximation error for our randomized method:

$$\mathcal{E} \leq \eta_1 \sqrt{E(\mathbf{X}, \widehat{\mathcal{S}}^{opt})} + \eta_2 E(\mathbf{X}, \widehat{\mathcal{S}}^{opt}), \tag{13}$$

where $\widehat{\mathcal{S}}^{opt}$ is the optimal partitioning of the reduced data $\widehat{\mathbf{X}}$ and $E(\mathbf{X}, \widehat{\mathcal{S}}^{opt})$ represents the quantization error when $\widehat{\mathcal{S}}^{opt}$ is used to cluster the high-dimensional data $\mathbf{X}$. We assume the partitioning in the reduced data set is within a constant factor of optimal, so this constant is absorbed into η_1 and η_2. In (Boutsidis et al. 2015), it is shown that by choosing $p' = \mathcal{O}(m/\varepsilon^2)$ dimensions for the random projection matrix $\mathbf{H}$, the following inequality holds with probability at least 0.96 over the randomness of $\mathbf{H}$: $E(\mathbf{X}, \widehat{\mathcal{S}}^{opt}) \leq (2+\varepsilon) E(\mathbf{X}, \mathcal{S}^{opt})$. Thus, employing this inequality in (13) completes the proof. □

Theorem 1 reveals important insights about the performance of our method. Although our algorithm generates landmark points based on the random projections of data, we can relate the approximation error to the total quantization error of partitioning the *original* data. In fact, the worst-case bounds for our proposed method with $p' = \mathcal{O}(m)$ and the original Clustered Nyström method (Proposition 1) are roughly similar. Thus, the dimension of reduced data p' is independent of the ambient dimension p and depends only on m (the number of landmark points) and ε (the distortion factor). As a result, for high-dimensional data, the dimension of reduced data p' can be fixed based on the desired number of landmark points and accuracy.

Experimental Results

In this section, we present experimental results comparing our Randomized Clustered Nyström (Algorithm 1) with state-of-the-art methods. Our proposed approach is implemented in MATLAB with the C/mex implementation for computing the sample mean. To perform K-means clustering, we use MATLAB's built-in function `kmeans` and the maximum number of iterations is set to 10.

Our Randomized Clustered Nyström (denoted by Ours) is compared with six kernel approximation methods:

1. The eigenvalue decomposition (denoted by EVD), where the best rank-r approximation of $\mathbf{K}$ is computed.
2. The standard Nyström method (denoted by Uniform), where landmark points are selected uniformly at random without replacement.
3. The Clustered Nyström method (denoted by Clustered Nys), where landmark points are generated using centroids resulting from K-means clustering on the original data (Zhang and Kwok 2010).
4. The Nyström method based on leverage score sampling (denoted by Lev), which requires performing EVD of the kernel matrix $\mathbf{K}$ (Gittens and Mahoney 2016).
5. The Nyström method based on landmark points selected using Determinantal Point Processes (denoted by DPP) (Li, Jegelka, and Sra 2016).
6. The Nyström method based on λ-ridge leverage scores (denoted by λ-Ridge Lev) for the regression task, where landmark points are selected with probabilities proportional to the diagonal entries of the matrix $\mathbf{K}(\mathbf{K} + \lambda \mathbf{I}_{n \times n})^{-1}$ (Alaoui and Mahoney 2015).

The exact computations of leverage scores and λ-ridge leverage scores require computing the eigenvalue decomposition of the kernel matrix and matrix inversion, respectively. Thus, the complexity of these two landmark selection techniques is at least quadratic with respect to the data set size n, in addition to the cost of forming the entire kernel matrix $\mathcal{O}(pn^2)$. To reduce the computational burden, there exist some algorithms to compute approximate leverage scores. For this reason, we exclude the running time for leverage sampling.

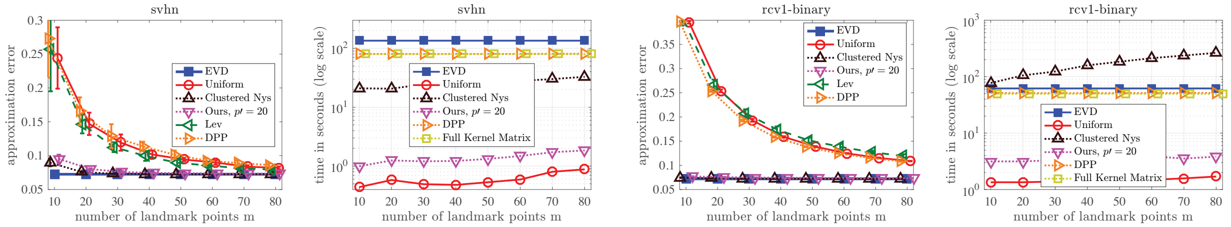
In the following, we examine the quality and generalization performance of the kernel approximation methods on classification and regression tasks using three benchmark high-dimensional data sets from the LIBSVM archive (Chang and Lin 2011):

- `svhn`: $p = 3{,}072$ and $n = 60{,}000$
- `rcv1-binary`: $p = 47{,}236$ and $n = 20{,}242$
- `E2006-tfidf`: $p = 150{,}360$ and $n = 6{,}000$

In all experiments, based on (Zhang and Kwok 2010), the Gaussian kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / c\right)$ is used with the parameter c chosen as the averaged squared distances between all the data points and sample mean.

Task 1: Kernel Approximation Error

We examine the quality of low-rank approximations in the form of $\mathbf{K} \approx \mathbf{L}\mathbf{L}^T$, where $\mathbf{L} \in \mathbb{R}^{n \times r}$ for fixed rank $r = 10$ and varying numbers of landmark points m. Although both benchmark data sets are high-dimensional, we set $p' = 20$ independent of the original dimension p, as suggested by Theorem 1. The accuracy is measured by the normalized kernel approximation error: $\|\mathbf{K} - \mathbf{L}\mathbf{L}^T\|_F / \|\mathbf{K}\|_F$. We report the mean and standard deviation of the approximation error over 20 trials in Fig. 1a and Fig. 1b. These results show



(a) `svhn`

(b) `rcv1-binary`

Figure 1: Kernel approximation error and runtime over 20 trials on `svhn` and `rcv1-binary` data sets.

that the accuracies of our method and Clustered Nyström reach the accuracy of the best rank-r approximation (EVD) even for small values of m, e.g., $m = 2r$. Uniform sampling and two nonuniform sampling techniques (leverage score and DPP) do not reach this accuracy even if a large number of landmark points are used, such as $m = 8r = 80$ for `rcv1-binary`. Therefore, in this example, our randomized method is more accurate than the state-of-the-art landmark selection techniques based on nonuniform sampling.

Moreover, based on Fig. 1a and Fig. 1b, it is observed that the runtime of our method is reduced at least by one order of magnitude compared to the Clustered Nyström method. In fact, our approach runs in roughly the same amount of time as uniform sampling, while achieving an accuracy comparable to the best rank-r approximation. It is worth pointing out that the actual time just to compute the full kernel matrix is about 80 and 50 seconds for the `svhn` and `rcv1-binary` data sets, respectively, while our method takes only about 1.2 and 3.1 seconds to find an accurate rank-r approximation of the kernel matrix.

Task 2: Classification via KPCA

In this experiment, we demonstrate the performance and generalization error of our method on a classification task using the features extracted via KPCA as described in the Introduction. We randomly sample $n_{train} = 0.8n$ data points from `rcv1-binary` for training and the remaining $n_{test} = 0.2n$ data points for testing. The K-nearest neighbors classifier is employed to classify the test data based on the features extracted via KPCA using 10 nearest neighbors with the fixed parameter $r = 20$ and two values of $p' = 20$ and $p' = 100$.

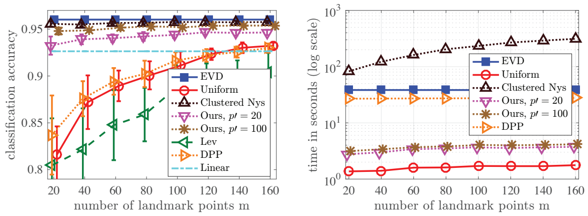


Figure 2: Classification accuracy and runtime over 20 trials using KPCA features on `rcv1-binary`.

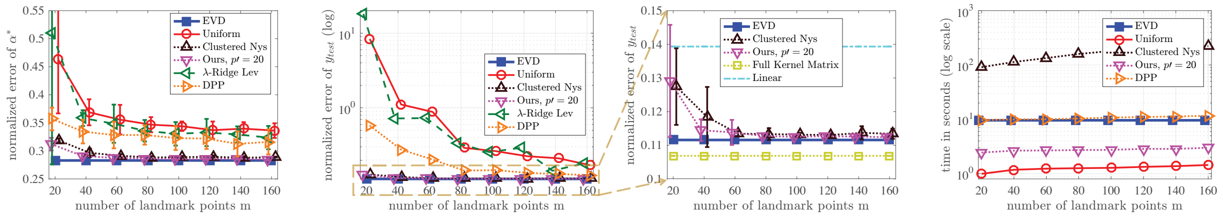
As we see in Fig. 2, our method outperforms the uniform and nonuniform sampling techniques by almost 10 percent in classification accuracy when the number of landmark points is small, e.g., $m = 20$. The classification accuracy of the same classifier on the original data set without using KPCA features is about 0.92, thus the rank-r approximation of $\mathbf{K}$ using our method leads to more accurate classification of the input data points for all values of m. Moreover, we see that our method achieves an almost 30 times speedup over the Clustered Nyström method when $m = 20$. Thus, our approach outperforms other landmark selection techniques using the same amount of time.

Task 3: KRR

In this experiment, we study the quality and generalization error of the kernel approximation methods when used with KRR. We randomly sample $n_{train} = 0.8n$ data points from `E2006-tfidf` for training and the remaining $n_{test} = 0.2n$ for testing. We set the rank parameter $r = 20$, regularization $\lambda = 2^{-4}$, and $p' = 20$. The values of r and λ are chosen such that KRR with the Gaussian kernel function achieves a better accuracy than the linear regression.

In the *training* process, the normalized approximation error of the solution in (5) is defined as $\|\boldsymbol{\alpha}^* - \widehat{\boldsymbol{\alpha}}\|_2 / \|\boldsymbol{\alpha}^*\|_2$, where $\boldsymbol{\alpha}^*$ is the exact solution to the optimization problem in (4). We report the mean and standard deviation of the approximation error over 20 trials in Fig. 3a. These results show that our method is more accurate than those based on uniform sampling, DPP sampling, and λ-ridge leverage scores. In fact, the accuracy of our method and Clustered Nyström reaches the accuracy of the best rank-r approximation of $\mathbf{K}$ for $m = 2r = 40$.

In the *prediction* stage, the response vector for the test data points, i.e., $\widehat{\mathbf{y}}_{test} \in \mathbb{R}^{n_{test}}$, is estimated using the approximate solution $\widehat{\boldsymbol{\alpha}}$. The normalized estimation error is defined as $\|\mathbf{y}_{test} - \widehat{\mathbf{y}}_{test}\|_2 / \|\mathbf{y}_{test}\|_2$ and we report the mean of the approximation error over 20 trials in Fig. 3b. It is observed that our method and Clustered Nyström significantly outperform the other landmark selection techniques for small values of m. Moreover, the accuracy of our method and Clustered Nyström reaches the accuracy of the best rank-r approximation obtained via EVD for the number of landmark points $m = 3r = 60$. In this figure, “Full Kernel Matrix” shows the normalized estimation error of $\mathbf{y}_{test}$ when $\boldsymbol{\alpha}^*$ is computed using the full kernel matrix $\mathbf{K}$ without any low-rank approximation, cf. (4).



(a) normalized error of $\boldsymbol{\alpha}^*$

(b) normalized error of $\mathbf{y}_{test}$

(c) runtime

Figure 3: KRR on `E2006-tfidf`. The normalized error of $\boldsymbol{\alpha}^*$, $\mathbf{y}_{test}$, and runtime over 20 trials are reported.

Fig. 3c shows average runtime for varying values of m. We see that our method achieves an almost 50 times speedup over Clustered Nyström when $m = 60$. Hence, our method is more accurate and efficient than the other landmark selection techniques in this example.

One-pass Variant of Algorithm 1

In some cases, it is not feasible to store and process large high-dimensional data in the main memory or RAM. Thus, such data sets are often analyzed in a streaming fashion, where the data points are presented sequentially without making extra passes over the data (Mitliagkas, Caramanis, and Jain 2013; Gilbert, Park, and Wakin 2012). In this experiment, we show that Randomized Clustered Nyström (Algorithm 1) can be implemented in single pass over the data.

To see this, note that our proposed method maintains a low-dimensional sketch of each data point $\mathbf{x}_i \in \mathbb{R}^p$ as $\widehat{\mathbf{x}}_i = \mathbf{H}\mathbf{x}_i \in \mathbb{R}^{p'}$, $i = 1, \ldots, n$. Therefore, the required memory/storage space to store $\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_n$ is reduced by a factor of $p/p' > 1$. The K-means clustering algorithm is then performed on these sketches to find m clusters $\widehat{\mathcal{S}}_1^{opt}, \ldots, \widehat{\mathcal{S}}_m^{opt}$ and the corresponding cluster centroids:

$$\widehat{\mathbf{z}}_j = \frac{1}{|\widehat{\mathcal{S}}_j^{opt}|} \sum_{\widehat{\mathbf{x}}_i \in \widehat{\mathcal{S}}_j^{opt}} \widehat{\mathbf{x}}_i, \quad j = 1, \ldots, m. \tag{14}$$

Because the sketching approximately preserves Euclidean distances, for kernel functions that depend only on the Euclidean distance, like the Gaussian kernel, the one-pass variant of our method *directly* computes matrices $\mathbf{C} \in \mathbb{R}^{n\times m}$ and $\mathbf{W} \in \mathbb{R}^{m\times m}$ in the Nyström method based on the low-dimensional sketches:

$$C_{ij} = \kappa(\widehat{\mathbf{x}}_i, \widehat{\mathbf{z}}_j), \quad W_{ij} = \kappa(\widehat{\mathbf{z}}_i, \widehat{\mathbf{z}}_j). \tag{15}$$

Next, we examine the quality of low-rank approximations in the form of $\mathbf{K} \approx \mathbf{L}\mathbf{L}^T$, where $\mathbf{L} \in \mathbb{R}^{n\times r}$ for fixed rank $r = 3$ on `E2006-tfidf`. The one-pass variant of our method with $p' = 20$ is compared with the standard Nyström method, where landmark points are selected uniformly at random from the input data set. We exclude the other kernel approximation methods since they require many passes over the input data set. Note that even the standard Nyström method requires more than one pass over the data to form the matrix $\mathbf{C}$; one partial pass over the original data set to select landmark points and one full pass over the original data to compute the matrix $\mathbf{C}$.

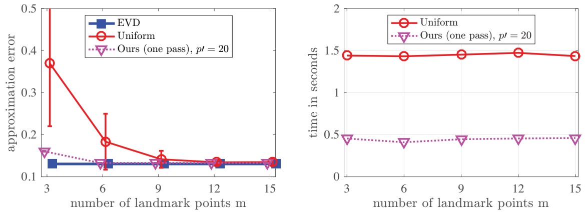


Figure 4: Kernel approximation error and runtime over 20 trials on `E2006-tfidf` data set.

In Fig. 4, we see that our method clearly outperforms the uniform sampling technique for small values of m, such as $m = 3$. Meanwhile, our method reduces the computational cost by a factor of 3. Thus, we see that the one-pass variant of our randomized method leads to more accurate and efficient low-rank approximations with only a single pass over the data. Providing theoretical guarantees for the one-pass variant is an important question for future work.

Acknowledgments

The first two authors are supported by the Bloomberg Data Science Research Grant Program.

References

Achlioptas, D. 2003. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences* 66(4):671–687.

Ailon, N., and Chazelle, B. 2009. The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing* 39:302–322.

Alaoui, A., and Mahoney, M. 2015. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems (NIPS)*, 775–783.

Arthur, D., and Vassilvitskii, S. 2007. k-means++: The advantages of careful seeding. In *ACM-SIAM symposium on Discrete algorithms (SODA)*, 1027–1035.

Bach, F. 2013. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, 185–209.

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

Boutsidis, C.; Zouzias, A.; Mahoney, M.; and Drineas, P. 2015. Randomized dimensionality reduction for K-means clustering. *IEEE Transactions on Information Theory* 61(2):1045–1062.

Calandriello, D.; Lazaric, A.; and Valko, M. 2016. Analysis of Nyström method with sequential ridge leverage score sampling. In *Uncertainty in Artificial Intelligence*.

Calandriello, D.; Lazaric, A.; and Valko, M. 2017. Distributed adaptive sampling for kernel matrix approximation. In *International Conference on Artificial Intelligence and Statistics*.

Chang, C., and Lin, C. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2(3):27:1–27:27.

Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine learning* 20(3):273–297.

Cortes, C.; Mohri, M.; and Talwalkar, A. 2010. On the impact of kernel approximation on learning accuracy. In *AISTATS*, 113–120.

Dasgupta, S. 2008. The hardness of K-means clustering. Technical Report CS2008-0916, UCSD.

Gilbert, A.; Park, J.; and Wakin, M. 2012. Sketched SVD: Recovering spectral features from compressive measurements. *arXiv preprint arXiv:1211.0361*.

Gittens, A., and Mahoney, M. 2016. Revisiting the Nyström method for improved large-scale machine learning. *Journal of Machine Learning Research* 1–65.

Halko, N.; Martinsson, P.; and Tropp, J. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review* 53(2):217–288.

Kumar, S.; Mohri, M.; and Talwalkar, A. 2012. Sampling methods for the Nyström method. *Journal of Machine Learning Research* 13:981–1006.

Li, C.; Jegelka, S.; and Sra, S. 2016. Fast DPP sampling for Nyström with application to kernel methods. In *International Conference on Machine Learning*, 2061–2070.

Liberty, E., and Zucker, S. 2009. The mailman algorithm: A note on matrix–vector multiplication. *Information Processing Letters* 109(3):179–182.

Mahoney, M. 2011. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning* 3(2):123–224.

Mitliagkas, I.; Caramanis, C.; and Jain, P. 2013. Memory limited, Streaming PCA. In *Advances in Neural Information Processing Systems (NIPS)*, 2886–2894.

Ostrovsky, R.; Rabani, Y.; Schulman, L. J.; and Swamy, C. 2012. The effectiveness of Lloyd-type methods for the K-means problem. *Journal of the ACM* 59(6):28.

Pourkamali-Anaraki, F., and Becker, S. 2016. A randomized approach to efficient kernel clustering. In *IEEE Global Conference on Signal and Information Processing*, 207–211.

Pourkamali-Anaraki, F., and Becker, S. 2017a. Improved fixed-rank Nyström approximation via QR decomposition: Practical and theoretical aspects. *arXiv preprint arXiv:1708.03218*.

Pourkamali-Anaraki, F., and Becker, S. 2017b. Preconditioned data sparsification for big data with applications to PCA and K-means. *IEEE Transactions on Information Theory* 63(5):2954–2974.

Saunders, C.; Gammerman, A.; and Vovk, V. 1998. Ridge regression learning algorithm in dual variables. In *International Conference on Machine Learning*, 515–521.

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

Schölkopf, B.; Smola, A.; and Müller, K. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10(5):1299–1319.

Sun, S.; Zhao, J.; and Zhu, J. 2015. A review of Nyström methods for large-scale machine learning. *Information Fusion* 26:36–48.

Suykens, J., and Vandewalle, J. 1999. Least squares support vector machine classifiers. *Neural Processing Letters* 9(3):293–300.

Tropp, J.; Yurtsever, A.; Udell, M.; and Cevher, V. 2017. Fixed-rank approximation of a positive-semidefinite matrix from streaming data. In *Advances in Neural Information Processing Systems (NIPS)*.

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

Wang, S.; Gittens, A.; and Mahoney, M. 2017. Scalable kernel K-means clustering with Nyström approximation: Relative-error bounds. *arXiv preprint arXiv:1706.02803*.

Williams, C., and Seeger, M. 2001. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, 682–688.

Woodruff, D. 2014. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science* 10(1–2):1–157.

Yan, B., and Sarkar, P. 2016. On robustness of kernel clustering. In *Advances in Neural Information Processing Systems (NIPS)*, 3098–3106.

Zhang, K., and Kwok, J. 2010. Clustered Nyström method for large scale manifold learning and dimension reduction. *IEEE Transactions on Neural Networks* 21(10):1576–1587.

Zhang, K.; Lan, L.; Wang, Z.; and Moerchen, F. 2012. Scaling up kernel SVM on limited resources: A low-rank linearization approach. In *International Conference on Artificial Intelligence and Statistics*, 1425–1434.

Zhang, K.; Tsang, I.; and Kwok, J. 2008. Improved Nyström low-rank approximation and error analysis. In *International Conference on Machine learning*, 1232–1239.