Clustered Nyström Method for Large Scale Manifold Learning and Dimension Reduction

Kai Zhang and James T. Kwok

Abstract— **Kernel (or similarity) matrix plays a key role in many machine learning algorithms such as kernel methods, manifold learning, and dimension reduction. However, the cost of storing and manipulating the complete kernel matrix makes it infeasible for large problems. The Nyström method is a popular sampling-based low-rank approximation scheme for reducing the computational burdens in handling large kernel matrices. In this paper, we analyze how the approximating quality of the Nyström method depends on the choice of landmark points, and in particular the encoding powers of the landmark points in summarizing the data. Our (non-probabilistic) error analysis justifies a "clustered Nyström method" that uses the k-means clustering centers as landmark points. Our algorithm can be applied to scale up a wide variety of algorithms that depend on the eigenvalue decomposition of kernel matrix (or its variant), such as kernel principal component analysis, Laplacian eigenmap, spectral clustering, as well as those involving kernel matrix inverse such as least-squares support vector machine and Gaussian process regression. Extensive experiments demonstrate the competitive performance of our algorithm in both accuracy and efficiency.**

Index Terms— **Dimension reduction, eigenvalue decomposition, kernel matrix, low-rank approximation, manifold learning, Nyström method, sampling.**

I. Introduction

KERNEL matrix plays an important role in many learning algorithms by providing an abundant description of the similarity relations in the data. It has demonstrated huge success in modeling real-world data with highly complex nonlinear structures. For example, in kernel methods, such as the support vector machines (SVMs) [1], kernel Fisher discriminant analysis [2], and kernel principal component analysis [3], the input data is mapped (via the kernel-induced feature map φ) to a very high dimensional Hilbert space, where scalar products are obtained efficiently through kernel evaluations. In manifold learning and dimensionality reduction (such as locally linear embedding [4], isomap [5], Laplacian eigenmap [6], and spectral clustering [7]–[9]), the eigenvectors of the kernel (or similarity) matrix reveal intrinsic structures of the data. The kernels can also be applied in complex network analysis as recently proposed by [10].

However, given a set of n sample points, the use of the kernel matrix necessitates storing and manipulating an $n \times n$ symmetric, positive (semi-)definite kernel matrix. The resultant complexities, namely quadratic in terms of space and (usually) cubic in terms of time, can be quite demanding for large problems. This poses a big challenge for practical applications. A useful way to alleviate the memory and computational burdens is to utilize the rapidly decaying spectra of kernel matrices [11] and perform low-rank approximation. Given a kernel matrix $K \in \mathbb{R}^{n\times n}$ whose rank m is much lower than n (i.e., $m \ll n$), it can be represented by

$$K = LL' \tag{1}$$

where $L \in \mathbb{R}^{n\times m}$. The computational and memory requirements associated with handling the matrix L will be much lower than those with the complete kernel matrix K. On the other hand, even when the kernel matrix has almost full rank, it might still be possible to approximate it by a low-rank positive semidefinite matrix [12], where the equality in (1) then becomes an approximation.

Low-rank approximation of the kernel matrix is useful in many different ways. For example, in each iteration of the interior point method (as applied to SVM), the most expensive step is in the solving of the linear system $(K + \delta I)u = w$, where $\delta > 0$ is a regularization parameter and I is the $n \times n$ identity matrix. In general, this requires $O(n^3)$ time and $O(n^2)$ space. Given the low-rank approximation in (1), however, one can efficiently solve this linear system by utilizing the Sherman–Morrison–Woodbury formula

$$(K + \sigma I)^{-1} \simeq \frac{1}{\sigma}\left(I - L(\sigma I + L'L)^{-1}L'\right). \tag{2}$$

These can be reduced to $O(nm^2)$ time and $O(nm)$ space [12]. Besides, Gaussian processes [13] and the least-squares SVM (LS-SVM) [14] also require solving such a linear system. Therefore, they can benefit from the efficient matrix inversion (2) in computing the solutions [15].

Another application of the low-rank approximation is to reconstruct the eigensystem of a matrix. We will make use of the following proposition.

Proposition 1: Given the low-rank approximation $K \simeq LL'$ (1), the top m eigenvectors $U = [u_1, \ldots, u_m]$ of K can be obtained as $U \simeq LV\Lambda^{-(1/2)}$, where $V, \Lambda \in \mathbb{R}^{m\times m}$ are from the eigenvalue decomposition of the $m \times m$ matrix $L'L = V\Lambda V'$.

Manuscript received August 10, 2009; revised December 6, 2009, March 15, 2010, June 28, 2010, and July 30, 2010; accepted July 31, 2010. Date of publication August 30, 2010; date of current version October 6, 2010. This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region under Grant 614508.

K. Zhang is with the Life Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA (e-mail: kzhang2@lbl.gov).

J. T. Kwok is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: jamesk@cs.ust.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2010.2064786

The proof is in [16].

Therefore, low-rank approximation is useful to manifold learning and dimensionality reduction algorithms that rely on heavily on the eigenvectors of the kernel matrix. Examples include kernel principal components analysis (KPCA) [3], Laplacian eigenmap [6], multidimensional scaling (MDS) [17], spectral clustering [8], normalized cut [18], isomap [5], and kernel linear discriminant analysis [2].

Theoretically, the optimal rank-k approximation (w.r.t. to the spectral or Frobenius norm) is provided by the eigenvalue decomposition. However, finding this optimal low-rank approximation is impractical for large problems and efficient alternatives are needed. In recent years, enormous effort has been devoted to this topic, which fall into three categories. The first is greedy sampling, in which samples are chosen incrementally to minimize (an upper bound) the approximation error. Examples include fast greedy approximation [19], [20], greedy spectral embedding [21], and incomplete Cholesky decomposition (ICD) [22], [23], [12]. The second is the Nyström method, which samples subset of rows/columns of the kernel matrix to approximate the kernel matrix (as well as its eigensystem). The third is randomized algorithm that designs column/row sampling probabilities to achieve provable probabilistic bounds [24]–[27].

In terms of both time and memory, the Nyström method is the most efficient (as discussed in Section II-B). It has been successfully applied to the Gaussian processes [15], spectral clustering [16], [28], and MDS [29]–[31]. The commonest sampling schemes are random sampling [16], [29], [15] and furthest point sampling [30]. On the theoretical side, probabilistic error bounds have been studied in [26] and [32]. One problem with the probabilistic sampling scheme is that the sampling probabilities are sometimes computed on the basis of the norms of the rows/columns of the kernel matrix, which requires at least $O(n^2)$ time and space. This is infeasible for large problems. The probabilistic error bound is typically derived for a specific sampling scheme. Therefore, from practitioner's side, it is hard to use the error bound as a general criterion or guidance to gain insights into new designs. Empirically, it can be even worse than the random sampling scheme [25], [26].

In this paper, we pursue a different style of error analysis that gives a more direct, non-probabilistic delineation on how the landmark points affect the Nyström low-rank approximation error. Our key finding is that the Nyström low-rank approximation error depends crucially on the quantization error induced by encoding the sample set with the landmark points. This suggests that, besides applying greedy or probabilistic sampling, the landmark points can be simply chosen as the k-means cluster centers. We call it "clustered Nyström method." The k-means-based sampling works on the $n \times d$ data matrix, and avoids the storage and manipulation of the $n \times n$ kernel matrix. It is more efficient than probabilistic sampling schemes based on the norms of the rows/columns of the kernel matrix. On the other hand, our analysis opens the possibility of tackling the computational complexities of spectral methods via data coding techniques.

The complexity of k-means is only linear in the sample size and dimensionality, and, as will be shown in our experimental evaluations, only a few iterations suffice in practice. It demonstrates very encouraging performance which is often consistently better than other variants of the Nyström method in approximating the kernel matrix. We apply this clustered Nyström method for low-rank approximation in a number of manifold learning and dimensionality reduction algorithms, such as KPCA and Laplacian eigenmap. All demonstrate its competitive performance in recovering the intrinsic low-dimensional structures of the data.

The rest of this paper is organized as follows. Section II briefly reviews related works on low-rank approximation of symmetric semidefinite kernel matrices. Section III presents an analysis showing how the Nyström low-rank approximation error is affected by the landmark points, and then proposes the use of the k-means algorithm in the sampling step. The resultant variant will be called the clustered Nyström method. In Section IV, we experimentally compare our approach with a number of state-of-the-art low-rank decomposition techniques for manifold learning and dimensionality reduction, as well as supervised methods Gaussian process (GP) regression. The last section gives concluding remarks. Preliminary results have been reported in our conference paper [33].

II. Related Works

In this section, we give a brief review on the three categories of low-rank approximation techniques. These are the greedy approaches (Section II-A), Nyström methods (Section II-B), and randomized algorithms (Section II-C). A short comparison of their computational complexities is given in Section II-D. Moreover, while there are large-scale numerical solvers for eigenvalue decomposition (such as the ARPACK package [34]), they are most suited when: 1) the input matrix is highly structured (or sparse), and 2) computing the product of the input matrix and a vector is inexpensive (for example, $O(n)$, where n is the size of the input matrix). Typically, these conditions do not hold in our context. Therefore, sparse eigensolvers will not be the focus of this paper.

A. Greedy Approaches

Greedy approaches have been applied in several fast algorithms for approximating the kernel matrix. In [19], the kernel matrix K is approximated by the subspace spanned by a subset of its columns as $K \simeq K_I T$, where $K_I \in \mathbb{R}^{n \times m}$, I is the set containing indices of the selected m columns, and $T \in \mathbb{R}^{m \times n}$ is the coefficients. In each iteration, the column that maximally reduces the error $(K - K_I T)$ is selected. The algorithm terminates when the estimated error bound is below a threshold. It takes $O(m^2 ln)$ time using a random subset of size l in each iteration from which the optimal column is chosen. A similar scheme is used in [20], where the residue of approximation error is used to guide the sampling.

In [21], a greedy sampling scheme is proposed for fast spectral embedding. The samples are greedily selected according to their distances to the subspace spanned by the current dictionary. If the distance is below a threshold, the

sample will be skipped, otherwise, it will be included in the dictionary. After constructing the dictionary, all the examples are projected on it. This algorithm scales as $O(m^2n)$.

Another well-known greedy approach is the ICD [22], [23], [12]. Note that any psd matrix Q can be represented by the Cholesky factorization $Q = LL'$, where $L \in \mathbb{R}^{n\times n}$ is a lower triangular matrix [35]. In case Q is only positive semidefinite (but not of full rank), it is still possible to compute an incomplete Cholesky factorization LL' with $L \in \mathbb{R}^{n\times m}$, where $m < n$ is the rank of Q or a prespecified number. The ICD has a complexity $O(m^2n)$ and has been adopted successfully to scale up the training of the standard SVM and LS-SVM [23].

B. The Nyström Method

The Nyström method was originally designed to solve integral equations [36]. It chooses a subset of samples, called the landmark points, to approximately compute the kernel eigenfunctions. The most popular sampling scheme for the Nyström method is random sampling, which leads to fast versions of kernel machines [15] and spectral clustering [16]. In [29], several variants of MDS are all shown to be related to the Nyström approximation. Consider the integral equation that defines the kernel eigenfunction

$$\int p(y)k(x,y)\phi_i(y)dy = \lambda_i\phi_i(x) \tag{3}$$

where $p(\cdot)$ is the probability density function, k is a positive definite kernel function, and $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ and $\phi_1, \phi_2, \ldots$ are the eigenvalues and eigenfunctions of the integral equation, respectively. Given a set of i.i.d. samples $\{x_1, x_2, \ldots, x_q\}$ drawn from $p(\cdot)$, the basic idea is to approximate the integral by its empirical average: $(1/q)\sum_{j=1}^{q} k(x, x_j)\phi_i(x_j) \simeq \lambda_i\phi_i(x)$. Choosing x in the above equation from $\{x_1, x_2, \ldots, x_q\}$ leads to an eigenvalue decomposition $K^{(q)}U^{(q)} = U^{(q)}\Lambda^{(q)}$, where $K_{ij}^{(q)} = k(x_i, x_j)$ for $i, j = 1, 2, \ldots, q$, $U^{(q)} \in \mathbb{R}^{q\times q}$ has orthonormal columns and $\Lambda^{(q)} \in \mathbb{R}^{q\times q}$ is a diagonal matrix. The eigenfunctions ϕ_i's and eigenvalues λ_i's in (3) can be approximated by $U^{(q)}$ and $\Lambda^{(q)}$ as $\phi_i(x_j) \simeq \sqrt{q}U_{ji}^{(q)}, \lambda_i \simeq \lambda_i^{(q)}/q$ [15]. Using different subset sizes (q's), the Nyström method thus produces different approximations to λ_i and ϕ_i in the integral equation (3). In particular, the Nyström method using a small q can also be deemed as approximating the Nyström method using a large q. Denote the sample set by $\mathcal{X} = \{x_i\}_{i=1}^{n}$, with the corresponding $n \times n$ kernel matrix K. Then the Nyström method that randomly chooses a subset $\mathcal{Z} = \{z_i\}_{i=1}^{m}$ of m landmark points will approximate the eigensystem of the full kernel matrix $K\Phi_K = \Phi_K\Lambda_K$ by [15]

$$\Phi_K \simeq \sqrt{\frac{m}{n}}E\Phi_{\mathcal{Z}}\Lambda_{\mathcal{Z}}^{-1}, \quad \Lambda_K \simeq \frac{n}{m}\Lambda_{\mathcal{Z}}. \tag{4}$$

Here, $E \in \mathbb{R}^{n\times m}$ with $E_{ij} = k(x_i, z_j)$, and $\Phi_{\mathcal{Z}}, \Lambda_{\mathcal{Z}} \in \mathbb{R}^{m\times m}$ contain the eigenvectors and eigenvalues of $W \in \mathbb{R}^{m\times m}$ where $W_{ij} = k(z_i, z_j)$. Using the approximations in (4), K can be reconstructed as

$$K \simeq \Phi_K\Lambda_K\Phi_K' = EW^{-1}E'. \tag{5}$$

Equation (5) is the basis for Nyström low-rank approximation [16], [15] and is sometimes called the matrix completion view [16]. Our theoretical analysis will also be based on this formulation. Recently an ensemble Nyström method is proposed in [37], which is further generalized in [38].

In practice, there are two ways to obtain approximate eigenvalues/vectors of the kernel matrix by the Nyström method [16]. One is to directly use the Nyström extension (4), which simply involves computing E and W, performing the eigen decomposition of W and then extending its eigensystem to that of the complete kernel matrix. This is further extended to the density-weighted version in [39] and [40]. However, the resultant eigenvectors are not guaranteed to be orthogonal. The second approach utilizes the matrix completion view (5) to first obtain a low-rank approximation of $K \simeq LL'$ with $L = EW^{-(1/2)}$, and then apply Proposition 1 to obtain an orthogonal set of approximate eigenvectors. As has been shown in [20], the lack of orthogonality can adversely affect the approximation quality. Therefore, the second approach is preferred in obtaining more accurate solutions although it increases the computational time. In terms of complexity, though, the two methods are the same, i.e., $O(m^2n)$.

C. Randomized Algorithms

Randomized algorithms [24], [25], [27] are aimed at designing column/row sampling probabilities that achieve provable probabilistic bounds for the factorization of arbitrary-shaped matrices. An example is [26] in the context of Nyström low-rank approximations. The idea is to replace the random sampling with a nonuniform data-dependent sampling probabilities p_i's. The chosen rows and columns are re-weighted by p_i's and then plugged into the standard Nyström method (Section II-B). The probabilities p_i's can be computed in different ways. For example, one choice is $p_i = G_{ii}^2/\sum_{i=1}^{n} G_{ii}^2$, where G is the Gram matrix to be approximated. In the sequel, this will be called sampling scheme I. Note that for stationary kernels, the diagonal entries of the corresponding Gram matrix are all the same and so this scheme reduces to uniform sampling. Another sampling scheme proposed by [41] uses $p_i = \|G^{(i)}\|/\|G\|_F$, where $G^{(i)}$ is the ith column of the Gram matrix. In the sequel, this will be called sampling scheme II.

D. Computational Complexities

Greedy approaches usually take $O(m^2n)$ time and $O(mn)$ memory. For the randomized algorithm in [41], sampling scheme II needs to access the whole kernel matrix and its computational complexity is the highest [$O(n^2)$ time and space]. This can be reduced to $O(m^2n)$ time and $O(mn)$ space with the use of scheme I. In comparison, the Nyström method needs $O(m^2n)$ time and $O(mn)$ space. Furthermore, the intermediate matrices [E and W in (5)] needed in the Nyström method can be simply computed on demand. Thus, they need not be stored and this can greatly reduce the memory requirement on very large problems. In contrast, for greedy approaches, the intermediate matrices have to be incrementally updated and stored.

III. Clustered Nyström Method for Low-Rank Approximation

This section presents our key analysis on how the Nyström approximation error depends on the choice of the landmark points. We first point out in Section III-A an important observation. Then, we derive in Sections III-B–III-D an error bound in a more general setting based on the "clustered" data model. This error bound leads to important insights on the design of an efficient sampling scheme, i.e., the use of the k-means clustering centers as the landmark points. We call it "clustered Nyström method" and discuss it in Section III-E.

A. Observation

Proposition 2: Given $\mathcal{X} = \{x_i\}_{i=1}^n$ and the landmark point set $\mathcal{Z} = \{z_j\}_{j=1}^m$, the Nyström reconstruction of kernel entry $K(x_i, x_j)$ is exact if there exist two landmark points such that $z_p = x_i$, and $z_q = x_j$.

Proof: Let $K_{x_k,\mathcal{Z}} \in \mathbb{R}^m$ be the vector of kernel evaluations between sample x_k and all the landmark points in $\mathcal{Z}$. Then, using (5), the Nyström reconstruction of $K(x_i, x_j)$ is $K_{x_i,\mathcal{Z}} W^{-1} K'_{x_j,\mathcal{Z}}$, where $W \in \mathbb{R}^{m\times m}$ is the kernel matrix defined on the landmark set $\mathcal{Z}$. Let $W^{(k)}$ be the kth row of W, then $K_{x_i,\mathcal{Z}} = W^{(p)}$ and $K_{x_j,\mathcal{Z}} = W^{(q)}$, since $x_i = z_p$ and $x_j = z_q$. As a result, the reconstructed entry is $W^{(p)} W^{-1} (W^{(q)})' = W_{pq} = K(z_p, z_q) = K(x_i, x_j)$. ∎

Proposition 2 indicates that the landmark points should be chosen to have sufficient overlap with the data. However, it is often impossible to use a small landmark set to represent every sample accurately.

B. Approximation Error on Sub-Kernel Matrix

In this section, we apply a "clustered" data model to analyze the quality of Nyström low-rank approximation. Here, the data clusters can be naturally obtained by assigning each sample to the closest landmark point. As will be seen, this model allows the derivation of an explicit error bound for the Nyström approximation.

Again, suppose that the landmark set is $\mathcal{Z} = \{z_i\}_{i=1}^m$, and the whole sample set $\mathcal{X}$ is partitioned into m disjoint clusters $\mathcal{S}_k$'s. Let $c(i)$ be the function that maps each sample $x_i \in \mathcal{X}$ to the closest landmark point $z_{c(i)} \in \mathcal{Z}$, where $c(i) = \arg\min_{j=1,2,\ldots,m} \|x_i - z_j\|$. We assume that each cluster has T samples. If the cluster sizes differ, we add "virtual samples" to each cluster such that all the clusters have the same size $T = \max_{k=1}^m |\mathcal{S}_k|$. The virtual samples added to cluster $\mathcal{S}_k$ are chosen as the landmark point z_k for that cluster. Our goal is to study the approximation error in (5)

$$\mathcal{E} = \left\| K - E W^{-1} E' \right\|_F \tag{6}$$

where $\|\cdot\|_F$ denotes the matrix Frobenius norm. As can be seen in the sequel, the virtual samples added will not induce additional quantization error but may loosen the bound.

First, consider the simpler notion of *partial approximation error* defined as follows:

Definition 1: Suppose that each cluster has T samples. Repeat the following sampling process T times: At time t, randomly pick one sample from each of the m clusters, and denote the set of m samples obtained by $\mathcal{X}_{\mathcal{I}_t}$. Consequently, $\mathcal{X} = \{\mathcal{X}_{\mathcal{I}_1} \cup \mathcal{X}_{\mathcal{I}_2} \cup \ldots \cup \mathcal{X}_{\mathcal{I}_T}\}$, and the kernel matrix defined on $\mathcal{X}$ can be decomposed into T^2 $m \times m$ submatrices. Let $K_{\mathcal{I}_i,\mathcal{I}_j}$ be the submatrix of K defined on $(\mathcal{X}_{\mathcal{I}_i}, \mathcal{X}_{\mathcal{I}_j})$, and $E_{\mathcal{I}_i,\mathcal{Z}}$ be the submatrix of E defined on $(\mathcal{X}_{\mathcal{I}_i}, \mathcal{Z})$, and $W \in \mathbb{R}^{m\times m}$ be the kernel submatrix defined on $\mathcal{Z}$. The *partial approximation error* is defined as the difference between $K_{\mathcal{I}_i,\mathcal{I}_j}$ and its Nyström approximation measured w.r.t. the Frobenius norm

$$\mathcal{E}_{\mathcal{I}_i,\mathcal{I}_j} = \|K_{\mathcal{I}_i,\mathcal{I}_j} - E_{\mathcal{I}_i,\mathcal{Z}} W^{-1} E'_{\mathcal{I}_j,\mathcal{Z}}\|_F. \tag{7}$$

In the following, we assume that the kernel k satisfies the following property:

$$(k(a,b) - k(c,d))^2 \le C_{\mathcal{X}}^k \left(\|a-c\|^2 + \|b-d\|^2 \right) \tag{8}$$

$\forall a, b, c, d$, where $C_{\mathcal{X}}^k$ is a constant depending on k and the sample set $\mathcal{X}$. It will be shown in Section III-D that this assumption is valid on a number of commonly used kernels.

Proposition 3: For a kernel k satisfying property (8), the partial approximation error $\mathcal{E}_{\mathcal{I}_i,\mathcal{I}_j}$ is bounded by

$$\begin{aligned}\mathcal{E}_{\mathcal{I}_i,\mathcal{I}_j} \le{}& \sqrt{2mC_{\mathcal{X}}^k (e_{\mathcal{I}_i} + e_{\mathcal{I}_j})} + \sqrt{mC_{\mathcal{X}}^k e_{\mathcal{I}_i}} \\ &+ \sqrt{mC_{\mathcal{X}}^k e_{\mathcal{I}_j}} + mC_{\mathcal{X}}^k \sqrt{e_{\mathcal{I}_i} e_{\mathcal{I}_j}} \left\| W^{-1} \right\|_F \end{aligned} \tag{9}$$

where $e_{\mathcal{I}_i}$ is the quantization error induced by encoding each sample in $\mathcal{X}_{\mathcal{I}_i}$ by the closest landmark point in $\mathcal{Z}$

$$e_{\mathcal{I}_i} = \sum_{x_i \in \mathcal{X}_{\mathcal{I}_i}} \left\| x_i - z_{c(i)} \right\|^2. \tag{10}$$

Proof: Note that $K_{\mathcal{I}_i,\mathcal{I}_j}$, $E_{\mathcal{I}_i,\mathcal{Z}}$ $E_{\mathcal{I}_j,\mathcal{Z}}$, and W are all kernel matrices defined on $(\mathcal{X}_{\mathcal{I}_i}, \mathcal{X}_{\mathcal{I}_j})$, $(\mathcal{X}_{\mathcal{I}_i}, \mathcal{Z})$ $(\mathcal{X}_{\mathcal{I}_j}, \mathcal{Z})$ and $(\mathcal{Z}, \mathcal{Z})$, respectively. Define the following "difference" matrices:

$$\begin{aligned} A_{\mathcal{I}_i,\mathcal{I}_j} &= K_{\mathcal{I}_i,\mathcal{I}_j} - W, \\ B_{\mathcal{I}_i,\mathcal{Z}} &= E_{\mathcal{I}_i,\mathcal{Z}} - W, \\ C_{\mathcal{I}_j,\mathcal{Z}} &= E_{\mathcal{I}_j,\mathcal{Z}} - W. \end{aligned} \tag{11}$$

We first show that they have bounded Frobenius norms. Without loss of generality, we specify the indices as follows: $K_{\mathcal{I}_i,\mathcal{I}_j}(p,q) = k(x_{\mathcal{I}_i(p)}, x_{\mathcal{I}_j(q)})$, $E_{\mathcal{I}_i,\mathcal{Z}}(p,q) = k(x_{\mathcal{I}_i(p)}, z_q)$, $E_{\mathcal{I}_j,\mathcal{Z}}(p,q) = k(x_{\mathcal{I}_j(p)}, z_q)$, and $W(p,q) = k(z_p, z_q)$. Using (8)

$$\begin{aligned} \|A_{\mathcal{I}_i,\mathcal{I}_j}\|_F^2 &= \sum_{p,q=1}^m \left(k(x_{\mathcal{I}_i(p)}, x_{\mathcal{I}_j(q)}) - k(z_p, z_q) \right)^2 \\ &\le C_{\mathcal{X}}^k \sum_{p,q=1}^m \left(\left\| x_{\mathcal{I}_i(p)} - z_p \right\|^2 + \left\| x_{\mathcal{I}_i(q)} - z_q \right\|^2 \right) \\ &= mC_{\mathcal{X}}^k \left(\sum_{p=1}^m \left\| x_{\mathcal{I}_i(p)} - z_p \right\|^2 + \sum_{q=1}^m \left\| x_{\mathcal{I}_j(q)} - z_q \right\|^2 \right) \\ &= 2mC_{\mathcal{X}}^k (e_{\mathcal{I}_i} + e_{\mathcal{I}_j}) \end{aligned}$$

where $e_{\mathcal{I}_i}$ is the same as that in (10) since $c(\mathcal{I}(q)) = q$. Similarly, for matrix $B_{\mathcal{I}_i,\mathcal{Z}}$, we have $\|B_{\mathcal{I}_i,\mathcal{Z}}\|_F^2 = \sum_{p,q=1}^m$

$\left(k(x_{\mathcal{I}_i(p)}, z_q) - k(z_p, z_q)\right)^2 \leq mC_{\mathcal{X}}^k \sum_{p=1}^m \left\|x_{\mathcal{I}_i(p)} - z_p\right\|^2 = mC_{\mathcal{X}}^k e_{\mathcal{I}_i}$, and for matrix $C_{\mathcal{I}_j,\mathcal{Z}}$, $\|C_{\mathcal{I}_j,\mathcal{Z}}\|_F^2 \leq mC_{\mathcal{X}}^k e_{\mathcal{I}_j}$. Finally, using (11), $\|\mathcal{E}_{\mathcal{I}_i,\mathcal{I}_j}\|_F = \|W + A_{\mathcal{I}_i,\mathcal{I}_j} - (W + B_{\mathcal{I}_i,\mathcal{Z}})W^{-1}(W + C_{\mathcal{I}_j,\mathcal{Z}})'\|_F = \|W + A_{\mathcal{I}_i,\mathcal{I}_j} - W' - C'_{\mathcal{I}_j,\mathcal{Z}} - B_{\mathcal{I}_i,\mathcal{Z}} - B_{\mathcal{I}_i,\mathcal{Z}}W^{-1}C'_{\mathcal{I}_j,\mathcal{Z}}\|_F \leq \|A_{\mathcal{I}_i,\mathcal{I}_j}\|_F + \|B_{\mathcal{I}_i,\mathcal{Z}}\|_F + \|C_{\mathcal{I}_j,\mathcal{Z}}\|_F + \|B_{\mathcal{I}_i,\mathcal{Z}}\|_F \cdot \|C_{\mathcal{I}_j,\mathcal{Z}}\|_F \cdot \|W^{-1}\|_F$. Using the bounds on $\|A_{\mathcal{I}_i,\mathcal{I}_j}\|$, $\|B_{\mathcal{I}_i,\mathcal{Z}}\|$, $\|C_{\mathcal{I}_j,\mathcal{Z}}\|$, together with the definition in (11), we obtain (9). ■

C. Approximation Error of the Complete Kernel Matrix

With the partial approximation error, we can now obtain a bound on the Nyström approximation error in (6). Note that $\mathcal{E} = \sum_{i,j=1}^T \mathcal{E}_{\mathcal{I}_i,\mathcal{I}_j}$. Hence, the basic idea is to sum up the partial errors $\mathcal{E}_{\mathcal{I}_i,\mathcal{I}_j}$ over all $i, j = 1, 2, \ldots, T$.

Proposition 4: For kernel k satisfying property (8), the error of the Nyström approximation (6) is bounded by

$$\mathcal{E} \leq 4T\sqrt{mC_{\mathcal{X}}^k eT} + mC_{\mathcal{X}}^k Te\|W^{-1}\|_F \tag{12}$$

where $T = \max_{k=1}^m |\mathcal{S}_k|$, and $e = \sum_{i=1}^n \left\|x_i - z_{c(i)}\right\|^2$ is the total quantization error of encoding each sample $x_i \in \mathcal{X}$ with the closest landmark point in $\mathcal{Z}$.

Proof: The bound on $\mathcal{E}$ is obtained by summing up the right-hand side of (9) over $i, j = 1, 2, \ldots, T$. By using the Cauchy–Schwarz inequality $\sum_{i=1}^n \sqrt{a_i} \leq \sqrt{n\left(\sum_{i=1}^n a_i\right)}$ and the fact that $e = \sum_{i=1}^T e_{\mathcal{I}_i}$, we have from the first term in (9)

$$\begin{aligned}
&\sum_{i,j=1}^T \sqrt{2mC_{\mathcal{X}}^k(e_{\mathcal{I}_i} + e_{\mathcal{I}_j})} \\
&= \sqrt{2mC_{\mathcal{X}}^k} \sum_{i=1}^T \sum_{j=1}^T \sqrt{e_{\mathcal{I}_i} + e_{\mathcal{I}_j}} \\
&\leq \sqrt{2mC_{\mathcal{X}}^k} \sum_{i=1}^T \sqrt{T\sum_{j=1}^T (e_{\mathcal{I}_i} + e_{\mathcal{I}_j})} \\
&= \sqrt{2mC_{\mathcal{X}}^k T} \sum_{i=1}^T \sqrt{Te_{\mathcal{I}_i} + e} \\
&\leq \sqrt{2mC_{\mathcal{X}}^k} \cdot T\sqrt{\sum_{i=1}^T \left(Te_{\mathcal{I}_i} + e\right)} = 2T(mC_{\mathcal{X}}^k Te)^{\frac{1}{2}}.
\end{aligned}$$

Similarly, the second (and third) term in (9) can be summed as $\sum_{i,j=1}^T \sqrt{mC_{\mathcal{X}}^k e_{\mathcal{I}_i}} = \sqrt{mC_{\mathcal{X}}^k} \sum_{j=1}^T (\sum_{i=1}^T \sqrt{e_{\mathcal{I}_i}}) \leq T\sqrt{mC_{\mathcal{X}}^k eT}$. The last term in (9) can be summed as $\sum_{i,j=1}^T mC_{\mathcal{X}}^k \sqrt{e_{\mathcal{I}_i} e_{\mathcal{I}_j}} \|W^{-1}\|_F = mC_{\mathcal{X}}^k \|W^{-1}\|_F (\sum_{i=1}^T \sqrt{e_{\mathcal{I}_i}})^2 \leq mC_{\mathcal{X}}^k \|W^{-1}\|_F Te$. By combining all these terms, we arrive at (12). ■

D. Validity of (8) for Various Kernels

In this section, we show that many commonly used kernel functions satisfy the property in (8). First, consider a stationary kernel of the form $k(x, y) = \kappa(\|x - y\|/\sigma)$. This includes, for example, the Gaussian kernel with $\kappa(\alpha) = \exp(-\alpha^2)$, the Laplacian kernel with $\kappa(\alpha) = \exp(-\alpha)$, and the inverse distance kernel with $\kappa(\alpha) = (\alpha + \epsilon)^{-1}$.

Proposition 5: Stationary kernels of the form $k(x, y) = \kappa(\|x - y\|/\sigma)$ satisfy property (8).

Proof: By using the mean value theorem, we have, for any $a, b, c, d \in \mathbb{R}^d$, $(k(a, b) - k(c, d))^2 = (\kappa(\|a - b\|/\sigma) - \kappa(\|c - d\|/\sigma))^2 \leq [\kappa'(\xi)/\sigma]^2(\|a - b\| - \|c - d\|)^2$. Let $v_1 = a - c$ and $v_2 = b - d$. By using the triangular inequality, we have $\|c - d\| \leq \|a - b\| + \|v_1\| + \|v_2\|$ and similarly $\|a - b\| \leq \|c - d\| + \|v_1\| + \|v_2\|$. So, $(\|a - b\| - \|c - d\|)^2 \leq (\|a - c\| + \|b - d\|)^2 \leq 2(\|a - c\|^2 + \|b - d\|^2)$, and thus $(k(a, b) - k(c, d))^2 \leq 2[\kappa'(\xi)/\sigma]^2(\|a - c\|^2 + \|b - d\|^2)$. Hence, $C_{\mathcal{X}}^k$ in (8) can be chosen as $2\max_\xi[\kappa'(\xi)/\sigma]^2$ which is often bounded. For example, for the Gaussian kernel, we have $\kappa'(\xi) = -2\xi\exp(-\xi^2)$, whose magnitude approaches the maximum for $\xi = 1/\sqrt{2}$, so we have $2[\kappa'(\xi)/\sigma]^2 \leq 2\left((2/\sqrt{2})\exp(-1/2)/\sigma\right)^2 = (4/e\sigma^2)$, for the Laplacian kernel, $\kappa'(\xi) = -\exp(-\xi)$ is bounded by 1, so we have $2[\kappa'(\xi)/\sigma]^2 \leq 1/\sigma^2$, for the inverse distance kernel, $k'(\xi) = -(1/(\xi + \epsilon))^2$ is bounded by $(2/\epsilon^2)$, so we have $2[\kappa'(\xi)/\sigma]^2 \leq (2/\sigma^2\epsilon^4)$. ■

Similarly, this also holds for polynomial kernels.

Proposition 6: Polynomial kernels of the form $k(x, y) = \left(x'y + \epsilon\right)^d$ satisfy property (8).

Proof: Let $\kappa(z) = z^d$. Then

$$\begin{aligned}
&(k(a, b) - k(c, d))^2 = \left((a'b + \epsilon)^d - (c'd + \epsilon)^d\right)^2 \\
&= \left(\kappa'(\xi)(a'b - c'd)\right)^2 = \kappa'(\xi)^2 \cdot \left(\left((a - c)'b + (b - d)'c\right)\right)^2 \\
&\leq 2[\kappa'(\xi)]^2 \cdot \left(((a - c)'b)^2 + ((b - d)'c)^2\right) \\
&\leq 2[\kappa'(\xi)R]^2 \cdot \left(\|a - c\|^2 + \|b - d\|^2\right)
\end{aligned}$$

where R is the maximum distance between samples and the origin. The constant $C_{\mathcal{X}}^k$ in (8) can then be chosen as $2\max[\kappa'(\xi)R]^2 = 2(d(R^2 + \epsilon)^{d-1}R)^2$. ■

E. Sampling Procedure

The error bound in Proposition 4 provides important insights on how to choose the landmark points in the Nyström method. Unlike existing works, our error bound is not restricted to a specific sampling strategy. Instead, it provides a plain description on how the approximation error $\|K - EW^{-1}E'\|_F$ depends on the choice of the landmark points z_k's. As can be seen from Proposition 4, for a number of kernels, an important factor that influences the approximation quality is e, the error of quantizing each sample in $\mathcal{X}$ with the closest landmark in $\mathcal{Z}$. If this quantization error is zero, the Nyström low-rank approximation will be exact. This agrees with the ideal case discussed in Section III-A. It indicates that the better the landmark points can encode the data, the lower the resultant low-rank approximation error. Motivated by this observation and the fact that k-means clustering can find a local minimum of the quantization error [42], we propose to use the cluster centers obtained from k-means as the landmark points in the Nyström low-rank approximation, with the reconstruction step being the same as in the original Nyström method (5). We call

this "clustered Nyström method" considering that clustering is involved in both the error analysis (the clustered data model) and the sampling scheme (the k-means clustering).

The time complexity of the k-means algorithm is $O(ndl)$, where n is the sample size, d is the dimension, and l the number of iterations. In practice, since the distortion error drops most significantly in the first few iterations, we fix the number of iterations to a small integer (e.g., $l = 5$). Therefore the complexity is *linear* in the sample size and dimension. Moreover, the space complexity of k-means is also low, only $O(mn)$. Therefore, the k-means-based sampling scheme is suitable for large problems. It is more efficient than probabilistic sampling, most of which have to handle the whole kernel matrix to compute its column/row norms. Moreover, extensive research works have been devoted to further scale up the k-means algorithm for large applications.[1] For example, Kanungo *et al*. [43] proposed to store samples in a variant of the k-d tree. In [44], the nearest neighbor queries are reduced by using a k-d tree whose nodes store the sufficient statistics of the data. In [45], redundant distance calculations in the k-means algorithm are avoided by applying the triangle inequality. Besides these algorithmic advances, practical heuristics can also be used. For example, an over-relaxed scheme [46] can lead to faster convergence. The k-means can be parallelized [47] to utilize distributed computing facilities. One can also perform random sampling, or hierarchical k-means, to further reduce the computation. In our empirical evaluations, we use a naive MATLAB implementation of the k-means. As will be seen in Section IV, the resultant algorithm already demonstrates superior performance in terms of both accuracy and efficiency.

Of course, k-means-based sampling is not new, and has been applied heuristically in various circumstances such as [48]. However, to the best of our knowledge, it has not been applied in context of low-rank approximation of kernel matrices, not to mention a theoretical justification. Note that in the density-weighted Nyström extension [39], [40], k-means-based sampling is used for approximating eigenvectors of the kernel matrix. However, there exist important differences. First, the purpose of [39] and [40] is to extrapolate the kernel eigenfunction via the weighted Nyström extension, while the goal here is to directly reconstruct the kernel matrix via the Nyström low-rank approximation (5) (without any weighting scheme). Second, the density-weighted Nyström extension [as well as the standard Nyström extension (4)] can only provide non-orthogonal approximations. On the other hand, the Nyström low-rank approximation here allows us to compute an orthogonal set of approximate eigenvectors (by Proposition 1), which is more accurate in practice. Although the former is more efficient (the computation needed is only about one-third of the latter), it is less accurate. Recently, the k-means-based sampling scheme was further extended for large-scale semi-supervised learning with encouraging performance [49], [33].

[1]For example, in [45] and [43], the k-means algorithm has been used on datasets with 262 K and 400 K samples.

TABLE I
SUMMARY OF DATASETS USED IN THE UNSUPERVISED LEARNING EXPERIMENTS (n IS THE SAMPLE SIZE AND d IS THE DIMENSIONALITY)

	n	d		n	d
wdbc	569	30	german	1000	24
splice	1000	60	adult1a	1605	123
dna	2000	180	segment	2310	19
w1a	2477	300	satimage	4435	36
ionosphere	341	34	diabetes	768	8
australian	690	14	breastcancer	683	10
uci-3v8	769	64	uci-3v5	765	64
uci-5v6	753	64	uci-6v8	757	64
uci-8v9	757	64	uci-2v7	767	64
uci-3v9	771	64	uci-4v9	769	64

IV. EXPERIMENTS

In this section, we perform extensive experiments to compare the proposed method with existing algorithms on the low-rank approximation of kernel matrix (Section IV-A), KPCA (Section IV-B), Laplacian eigenmap, and spectral clustering (Section IV-C). In order to compare with standard algorithms that are not quite scalable, we are restricted to samples smaller than 5000. In practice, we have applied our algorithm to much larger datasets[2] and some results are presented in Section IV-D. We also make comparisons on GP regression (Section IV-E).

Experiments are performed on a number of benchmark datasets from the LIBSVM archive[3] (Tables I). The methods under comparison include: 1) ICD;[4] 2) Nyström: with random sampling; 3) Drineas: the probabilistic sampling scheme in [26] with sampling schemes I and II (as discussed in Section II-C); 4) Ours: the proposed method; and 5) singular value decomposition (SVD) (or eigenvalue decomposition in our context), which provides the optimal solution in terms of Frobenius/spectral norm [35]. Note that methods Nyström, Drineas and ours involve some randomness. Hence, to reduce statistical variability, their results are based on averages over 30 repetitions. Experiments are performed on a core-duo PC with a 2.13-GHz processor, and all the codes are written in MATLAB.

A. Low-Rank Approximation of Kernel Matrix

In this section, we evaluate the performance of different algorithms by measuring their numerical approximation errors (in terms of the Frobenius norm) on the kernel matrix. We gradually increase the subset size m from 1% to 10% of the dataset size. First, we report results for the Gaussian kernel $K(x, y) = \exp(-\|x - y\|^2/\gamma)$, where γ is chosen as the average squared distance between data points and the sample mean. The approximation errors vs. the number of landmark points are plotted in Fig. 1. As can be seen, ours is only inferior to SVD on most datasets. Moreover, Drineas is only comparable to or sometimes even worse than Nyström. Indeed, similar observations are also observed in the context of

[2]Results and codes at http://www.cse.ust.hk/~twinsen/manifold.htm.

[3]http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

[4]Code from http://www.di.ens.fr/~fbach/kernel-ica/kernel-ica1_2.tar.gz.

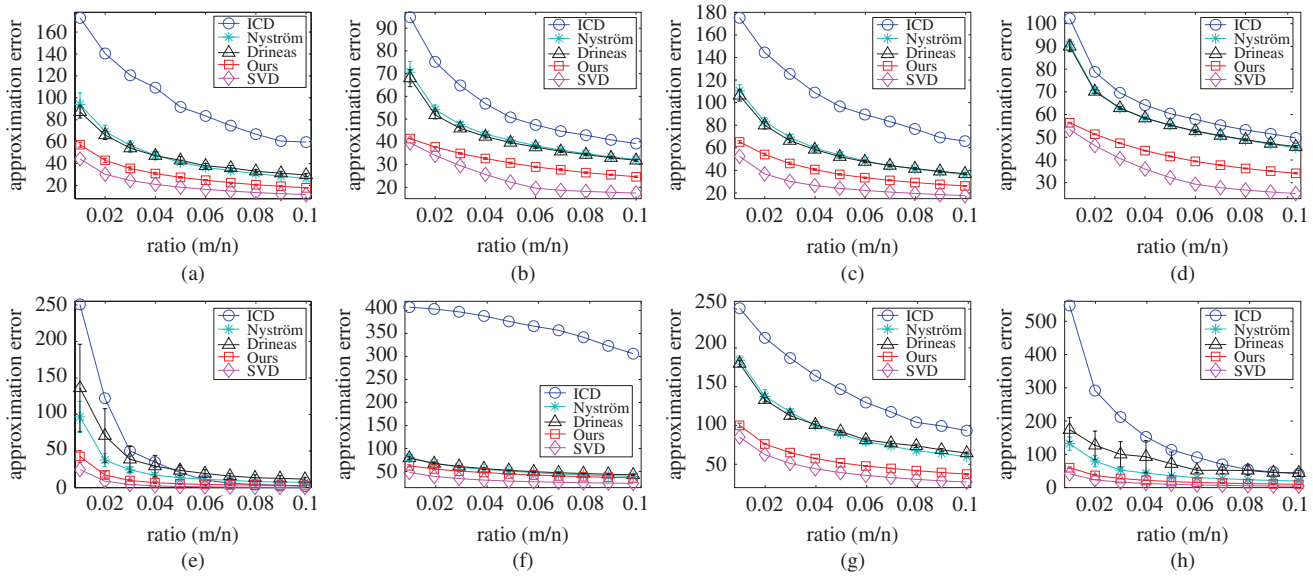

Fig. 1. Low-rank approximation errors of different algorithms using the Gaussian kernel. (a) German, (b) splice, (c) adult1a, (d) dna, (e) segment, (f) w1a, (g) uci, and (h) satimage.

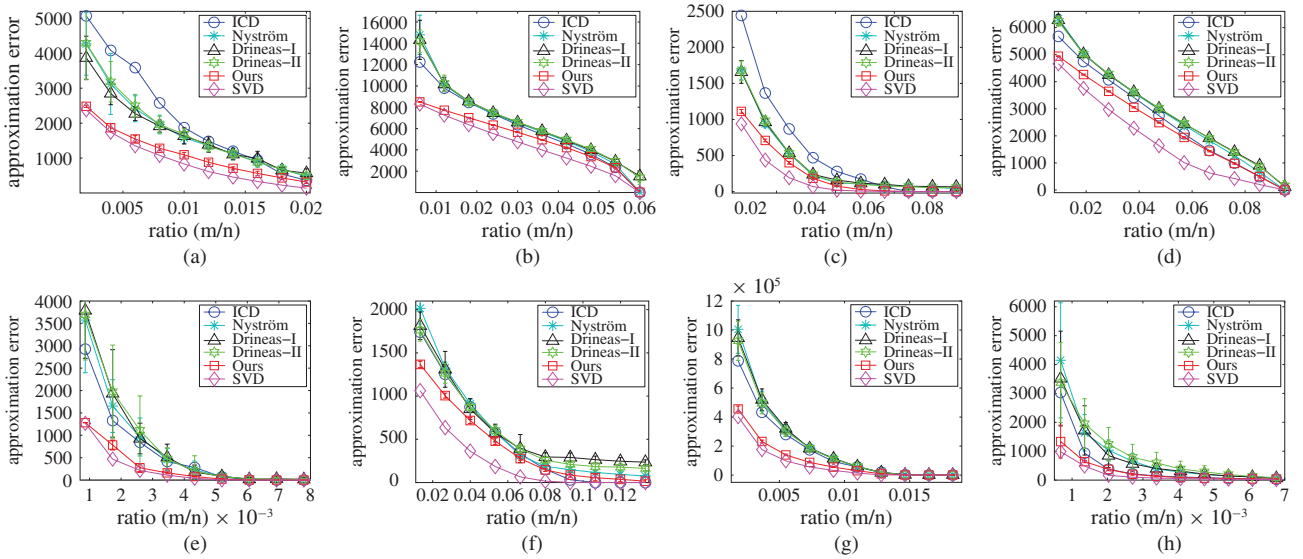

Fig. 2. Low-rank approximation errors of different algorithms using the linear kernel. (a) German, (b) splice, (c) adult1a, (d) dna, (e) segment, (f) w1a, (g) uci, and (h) satimage.

SVD [25]. ICD seems inferior on several datasets. However, for datasets whose kernel spectra decay rapidly to zero[5] (such as segment and satimage), ICD can quickly attain performance comparable to others.

We also experiment with the linear kernel and the polynomial kernel (of degree $d = 3$). As can be seen from Fig. 2, in case of the linear kernel, all the algorithms can quickly approach an approximation error that is close to zero. This is because the rank of the kernel matrix is at most the data dimensionality, which is small compared to the sample size. On average, our algorithm is very competitive compared to the other algorithms, and the same observation can be made in case of polynomial kernel in Fig. 3.

[5]The speed of decay of the kernel spectrum can be observed from the approximation curve of eigenvalue decomposition in Fig. 1. Note that the (squared) rank-m approximation error of SVD is $\sum_{i=m+1}^{n} \sigma_i^2$, where σ_i's are the singular values of K that have been sorted in a descending order [35]. Therefore, if SVDs error curve in Fig. 1 drops rapidly, so does the spectrum of K, indicating that the kernel matrix has a relatively low rank.

B. KPCA

In this section, we compare the performance of different low-rank approximation algorithms in KPCA. Given a kernel matrix K, the key step of KPCA is to perform eigenvalue decomposition on the centered kernel matrix $G = HKH$, where $H = I - (1/n)11'$ is the centering matrix, I is the $n \times n$ identity matrix, and $1 \in \mathbb{R}^n$ the vector of all 1's. With a low-rank approximation of K of the form $K \simeq LL'$, the eigen system of G can be solved efficiently by decomposing a much smaller matrix $(HL)'(HL)$ according to Proposition 1.

In this experiment, our task is to compute the top $r = 3$ features extracted by KPCA (using the Gaussian kernel). Let $U, \tilde{U} \in \mathbb{R}^{n \times r}$ be the matrices containing the

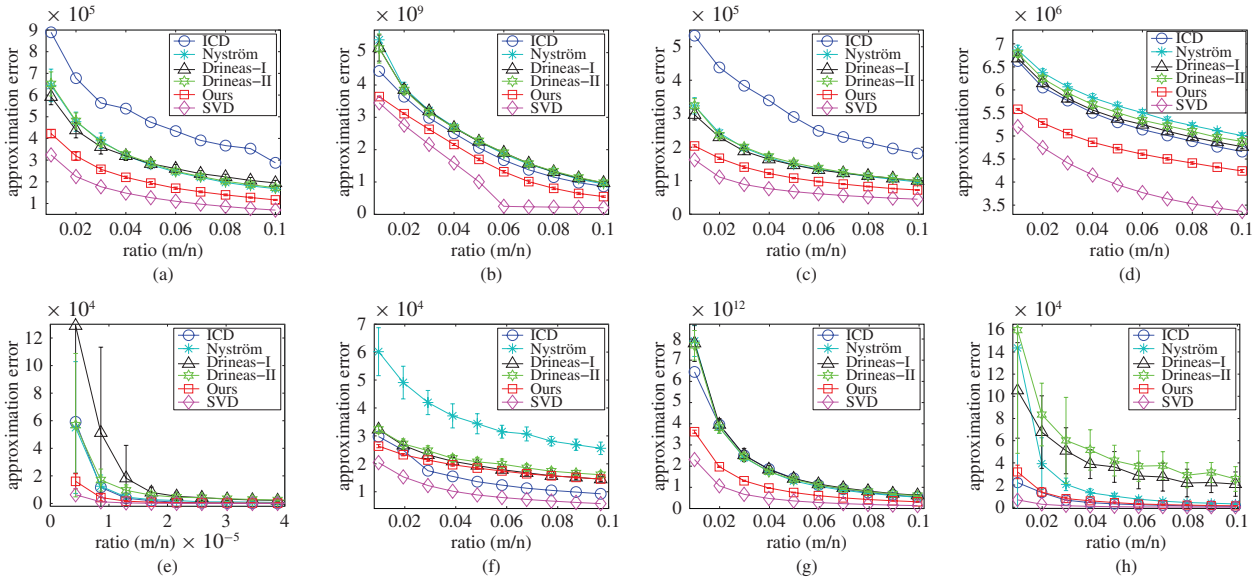

Fig. 3. Low-rank approximation errors of different algorithms on using the polynomial kernel. (a) German, (b) splice, (c) adult1a, (d) dna, (e) segment, (f) w1a, (g) uci, and (h) satimage.

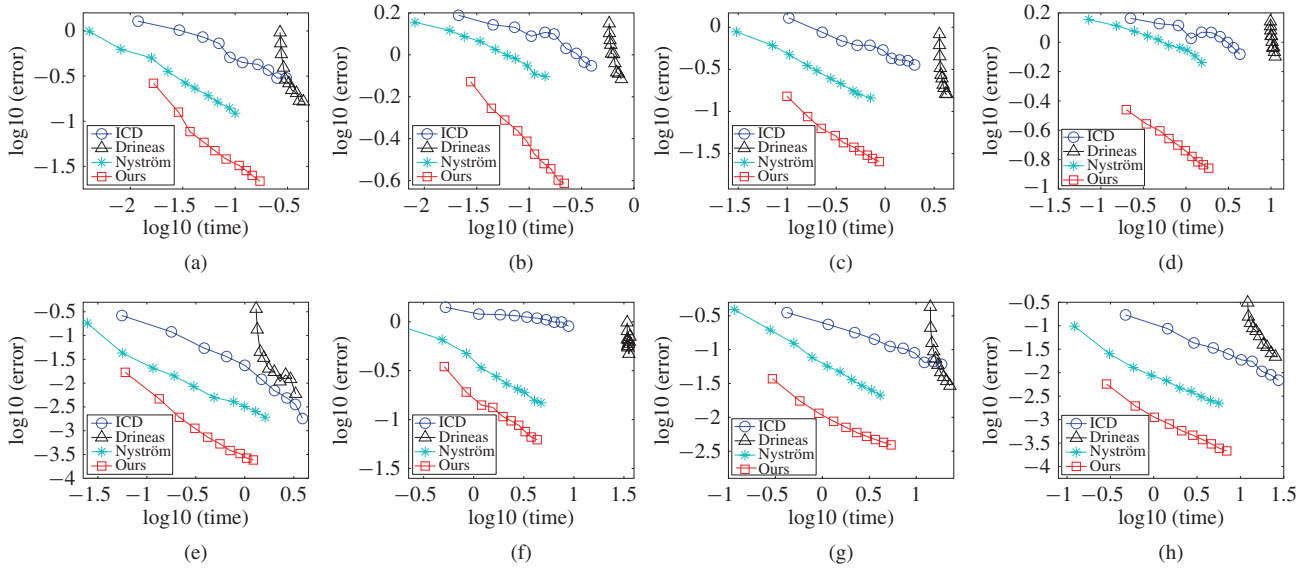

Fig. 4. Performance of different algorithms in approximating the top three features in KPCA. (a) German, (b) splice, (c) adult1a, (d) dna, (e) segment, (f) w1a, (g) uci, and (h) satimage.

top r extracted eigenvectors and the approximate ones, respectively. Following [21], we measure the misalignment $\epsilon = \min_{A \in \mathbb{R}^{r \times r}} \|U - \tilde{U}A\|_F^2$ between the spans of these two sets of eigenvectors, where $A \in \mathbb{R}^{r \times r}$ is any linear transform in the r-dimensional subspace. Fig. 4 plots the misalignment error vs. time (in log scale) for the different algorithms. Each node in the curve corresponds to one particular size (m) of the landmark set, which is gradually increased from 1% to 10% of the dataset size. As can be seen, on using the same number of landmark points (and hence the same amount of memory in encoding the matrix), our algorithm gives the lowest misalignment. Conversely, to achieve a specified accuracy, our method requires the minimum amount of time.

Since KPCA has been widely used for feature extraction, we examine the performance of the low-rank approximation schemes by applying the extracted KPCA features in a classifier. We randomly choose 80% of the patterns and use KPCA to extract $r = 3$ features. The remaining 20% of the data are then projected onto this low-dimensional space and a simple k-nearest-neighbor classifier (with $k = 20$) is used for prediction. Table II reports the classification errors averaged over 30 random repetitions. As can be seen, our method performs best on a number of datasets. Indeed, it even outperforms standard KPCA on a few datasets. We speculate that this is because our method performs k-means in selecting the landmark points, and thus can better remove the effect of noise.

C. Spectral Clustering and Embedding

Spectral clustering methods [8], such as the Laplacian eigenmap [6] and normalized cut [18], have been very popular

TABLE II

k-NN CLASSIFICATION ERRORS USING THE KPCA FEATURES EXTRACTED BY VARIOUS LOW-RANK APPROXIMATION SCHEMES

Data	Standard KPCA	Ours	Nyström	Drineas	ICD
wdbc	17.40±4.30	**7.19±2.53**	7.89±3.23	8.27±2.36	17.87±3.65
diabetes	**28.44±2.74**	29.68±2.73	28.48±3.26	28.96±2.38	28.57±2.74
splice	38.10±4.40	**22.48±2.95**	33.75±3.13	32.73±4.23	38.53±3.10
australian	16.52±2.65	**15.58±2.56**	16.64±3.28	16.74±2.85	16.55±2.66
adult1a	21.21±2.24	**20.66±2.17**	21.57±2.67	21.27±2.20	21.18±2.25
ionosphere	21.17±4.37	**13.19±4.43**	16.01±4.74	16.67±4.36	21.50±3.82
dna	37.13±6.07	**11.15±1.66**	32.45±4.48	31.24±3.50	38.06±2.51
german	30.62±3.06	31.53±3.02	29.82±3.52	**29.37±2.90**	30.50±3.17
breastcancer	3.97±1.43	**3.16±1.09**	3.89±1.39	3.89±1.48	4.01±1.47
segment	9.78±1.80	7.71±1.46	7.61±1.55	**7.34±1.60**	9.85±1.75
satimage	11.88±1.20	11.72±1.22	11.57±1.05	**11.36±1.25**	11.90±1.20

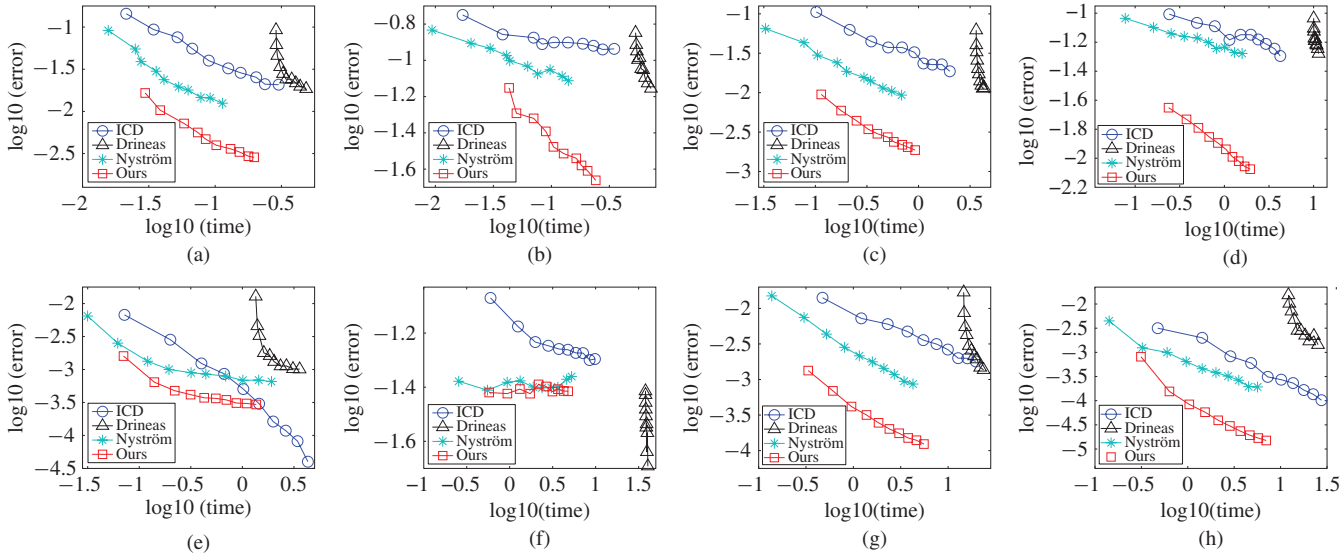

Fig. 5. Performance of different algorithms in approximating the 3-D embedding of Laplacian eigenmap. (a) German, (b) splice, (c) adult1a, (d) dna, (e) segment, (f) w1a, (g) uci, and (h) satimage.

methods in manifold learning and clustering. Common to these algorithms is the eigenvalue decomposition of the degree-normalized kernel matrix $\mathcal{S} = D^{-(1/2)}KD^{-(1/2)}$, where K is the kernel matrix and D is the diagonal degree matrix. The eigenvector corresponding to the second largest eigenvalue of $\mathcal{S}$ provides a relaxed solution to the NP-complete clustering problem [18]. Suppose that we have a low-rank approximation of the kernel matrix K of the form $K \simeq LL'$. The degree matrix can be approximated as $D \simeq \text{diag}\left(LL'1\right)$, and the eigen system of the normalized similarity matrix $\mathcal{S}$ can consequently be solved efficiently by Proposition 1.

In this section, we apply different low-rank approximation schemes to speed up the spectral methods of Laplacian eigenmap [6] and normalized cut [18]. A fully connected graph is used and the Gaussian kernel $\exp\left(-\|x\|^2/\gamma\right)$ is used to compute the similarity. The kernel width is chosen as the averaged squared distance between all the data points and the sample mean.

1) Laplacian Eigenmap: We examine the eigenvectors of the normalized Laplacian matrix corresponding to the three smallest nonzero eigenvalues. The performance criterion is the misalignment error as defined in Section IV-B. As can be seen in Fig. 5, for most of the datasets, our algorithm is very competitive with the other algorithms. Moreover, to attain the same accuracy level, our algorithm needs the least amount of computational time on most datasets.

2) Normalized Cut: We use some binary libsvm datasets and some pairs from the uci-digits. Clustering errors (averaged over 30 repetitions) are reported in Table III. For comparison, we also report the clustering errors of the standard normalized cut (without any approximation). As can be seen, on most datasets, the performance of our algorithm is very close to or slightly better than that of the original spectral clustering. In comparison, other fast approximations give inferior results. The time consumptions are not reported here because they are similar to those in the Laplacian eigenmap experiments (Fig. 5).

D. Large-Scale Examples

In this section, we perform spectral embedding (KPCA and Laplacian eigenmap) experiments on some large datasets. First, we use the isomap-face dataset which has 4000 patterns. The number of landmark points is set to $m = 100$, and the Gaussian kernel, with the kernel width chosen as the averaged squared distance between sample points and the

TABLE III
CLUSTERING PERFORMANCE OF THE DIFFERENT LOW-RANK APPROXIMATION ALGORITHMS IN THE CONTEXT OF NORMALIZED CUT

Data	NC	Ours	Nyström	Drineas	ICD
uci-3v8	3.3	**3.1±0.1**	5.6±2.3	6.4±6.9	9.4
uci-3v5	7.1	**6.5±0.3**	8.4±5.4	8.9±6.1	10.2
uci-5v6	**0.0**	**0.0±0.0**	1.7±1.4	3.2±8.4	3.7
uci-6v8	**0.8**	0.8±0.1	3.9±8.0	1.5±0.9	5.6
uci-8v9	5.6	**5.4±0.3**	11.2±8.9	10.3±6.8	5.7
uci-2v7	1.1	**1.1±0.1**	3.2±6.22	1.8±0.7	1.9
uci-3v9	19.5	**17.1±0.9**	17.6±9.8	14.6±11.5	23.8
uci-4v9	4.4	**4.3±0.2**	7.9±3.2	6.9±2.3	11.3
wdbc	**6.3**	8.3±0.3	13.1±8.5	17.9±9.1	20.3
diabetes	**34.3**	36.6±2.1	42.4±3.9	46.4±2.6	35.4
splice	36.4	**36.1±0.5**	41.9±4.2	40.1±3.6	44.2
german	**41.6**	41.8±0.6	42.7±3.6	44.3±3.7	47.3
australian	**18.2**	18.41±2.2	22.6±8.4	24.2±9.1	19.4
adult1a	30.4	**30.1±0.1**	32.0±1.6	32.4±2.2	30.6
ionosphere	**31.4**	32.0±0.9	35.5±4.5	38.9±3.2	32.8
breast	**2.6**	2.7±0.00	6.5±7.6	19.8±10.1	3.0
w1a	25.4	37.3±4.0	38.6±6.9	38.4±1.9	**23.1**
dna	**14.9**	20.1±11.4	30.9±6.2	31.5±5.9	38.6
segment	**42.1**	42.4±6.5	42.7±7.3	43.3±6.9	44.6
satimage	46.8	41.5±6.0	**40.3±5.7**	42.2±6.8	43.9

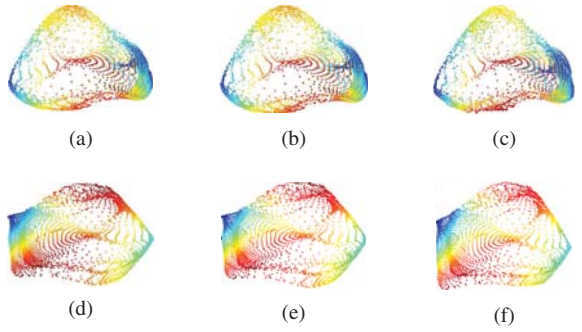

Fig. 6. Exact and approximate spectral embedding results on the isomap-face dataset. The number in brackets is the misalignment error e measured w.r.t. the exact embedding result. Top: KPCA; Bottom: Laplacian eigenmap. (a) Exact result. (b) Ours ($e = 0.047$). (c) Nystrom ($e = 0.68$). (d) Exact result. (e) Ours ($e = 5.14 \times 10^{-5}$). (f) Nystrom ($e = 1.51 \times 10^{-3}$).

sample mean, is used. Recall that for stationary kernels (such as the Gaussian kernel), sampling scheme I of [41] reduces to uniform sampling and so the Drineas method is the same as the Nyström method here. On the other hand, sampling scheme II of [41] and the ICD require $O(n^2)$ memory and so are too expensive on this dataset. The embedding results and the corresponding misalignment errors are shown in Fig. 6. To easily visualize the results, we use the two leading eigenvectors as the embedding dimensions and the third leading eigenvector for color coding. As can be seen, our results are closer to the ground truth than those of the Nyström method, both qualitatively and quantitatively.

We also experiment with two even larger datasets: connect (with 67 557 patterns) and ijcnn (with 100 000 patterns). Because of the sheer data size, the exact embedding solutions cannot be computed for reference. Nevertheless, Fig. 7 shows that there appear to exist interesting structures in these large-scale datasets, which can be seen more clearly from the results of the proposed method. These embeddings may be useful for further exploratory data analysis.

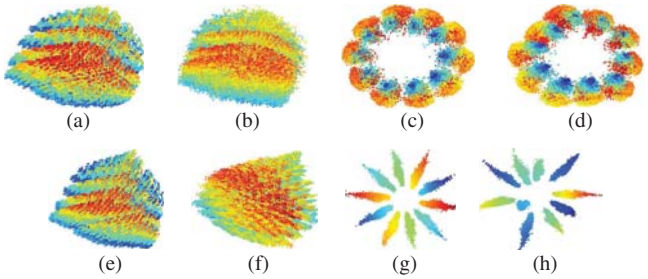

Fig. 7. Approximate spectral embedding results on some large datasets. Top: KPCA; Bottom: Laplacian eigenmap. (a) connect: ours. (b) connect: Nystrom. (c) ijcnn: ours. (d) ijcnn: Nystrom. (e) connect: ours. (f) connect: Nystrom. (g) ijcnn: ours. (h) ijcnn: Nystrom.

E. GP Regression

The Nyström-based low-rank approximation has been applied to speed up GP regression [15]. In this section, we empirically compare the performance of different low-rank approximation schemes in GP regression using the anisotropic squared exponential kernel $K(x_p, x_q) = \exp\left(-\sum_{i=1}^{D}((x_{pi} - x_{qi})^2/2\ell_i^2)\right)$, with a separate length scale for each input dimension on a data $X \in \mathbb{R}^{n\times d}$. We use two popular benchmark regression datasets, Boston housing and abalone, from the UC Irvine (UCI) Machine Learning Repository (http://archive.ics.uci.edu/ml/). The first dataset has 506 instances and 13 attributes, and the task is to predict the value of owner-occupied homes, while the second dataset has 4177 instances with 8 attributes, and the task is to predict the age of abalone from physical measurements. We randomly choose 90% of the data for training and the rest for testing. We first use GP to learn the length scales of the squared exponential kernel by maximizing the marginal likelihood. The learned length scales have an obvious anisotropic property.[6] We note that the resultant kernel widths along different dimensions demonstrate an obvious anisotropic property, which will serve as a good example to test the various low-rank approximation schemes in anisotropic situations.

We first examine the performance of different low-rank approximation schemes on this learned kernel. Again, the number of landmark points is gradually increased from 1% to 10% of the training size. Each method is repeated 30 times and the averaged approximation error is reported. Results are shown[7] in Fig. 8. As can be seen, the proposed method gives very competitive performance, and is inferior only to the optimal SVD decomposition.

Next, we perform GP regression using the full and approximate (low-rank) kernel matrices. The performance criterion is the root mean squared error $RMSE = (1/\max|y_i|)\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$ where y_i and $\hat{y}_i$ are the target and estimated output values on the ith pattern, respectively.

[6]For example, on the Boston housing data, the learned length scales are 3.2415, 21.6659, 40.0945, 66.2289, 0.8088, 3.0275, 4.1219, 1.6578, 3.0417, 0.8169, 6.9438, 6.0896, and 1.1679. Obviously, the second, third, and fourth dimensions have much larger length scales.

[7]In case of squared exponential kernel, sampling scheme I of Drineas reduces to random sampling (Section II-C of the main text) due to the diagonal entries that all equal to 1, and so Drineas-I boils down to the standard Nyström method.

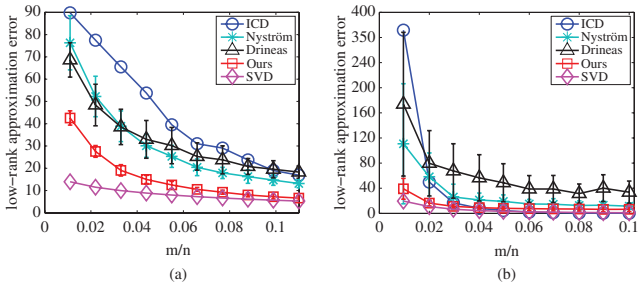

Fig. 8. Low-rank approximation errors of the different algorithms on an anisotropic kernel matrix. (a) Boston housing and (b) abalone.

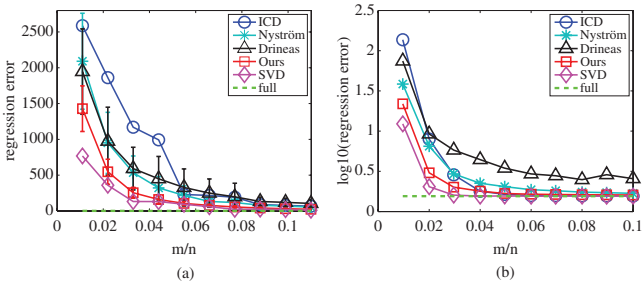

Fig. 9. Performance of the different algorithms in GP regression. To improve clarity, the error on the abalone dataset is plotted in log scale. (a) Boston housing and (b) abalone.

Regression errors on the test sets are in Fig. 9. As can be seen, the proposed method also gives very good performance. In particular, when only 5% of the data are used as landmark points, its performance on the abalone dataset is almost identical to that when the full kernel matrix is used. This clearly demonstrates the usefulness of low-rank approximation in supervised (regression) problems.

From these results, we can see that, even when the data is anisotropic, the proposed method (which uses k-means clustering for the sampling process) is still competitive. Although k-means clustering is unsuitable for non-spherical clusters, our use of the k-means algorithm here is not for clustering but for the selection of landmark points (in summarizing the data). When the data's global structure is anisotropic, the k-means algorithm simply splits it into multiple spherical clusters. This use of spherical clusters to approximate nonspherical data is similar to the use of isotropic kernels for approximating globally anisotropic distributions in Parzen window density estimation, and it is known that such an approximation converges to the ground truth asymptotically [50]–[51]. Of course, using anisotropic kernels would be more adaptive but requires more computations (e.g., k-means using Mahalanobis distance).

V. Conclusion

Low-rank matrix approximation is a useful technique for scaling up machine learning algorithms especially for dimensionality reduction and manifold learning. In this paper, we provided a novel error bound on the Nyström low-rank approximation, which relates the approximation error directly with the encoding power of the landmarks points. Our error analysis suggests the use of the k-means clustering centers as the landmark points. Empirically, our clustered Nyström low-rank approximation algorithm yields competitive performance, in terms of both time and accuracy, on a number of dimensionality reduction and manifold learning algorithms such as KPCA, spectral clustering, and Laplacian eigenmap.

The current work can be extended in several directions. For example, we will consider our algorithm in the more general context of SVD. We can also extend our algorithm to a supervised/semi-supervised learning scenario, where the choice of the landmark points will be affected by the class labels. Error analysis for nonstationary kernels (that are not of the polynomial form) will also be an interesting future direction to pursue. Finally, we are investigating the connections between the clustered Nyström method (for low-rank approximation) and the density-weighted Nyström extension (for computing kernel eigenfunctions) [40]. Results will be reported in the future.

References

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[2] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller, "Fisher discriminant analysis with kernels," in *Proc. IEEE Neural Netw. Signal Process. Workshop*, Aug. 1999, pp. 41–48.
[3] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.
[4] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
[5] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.
[6] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2002, pp. 585–591.
[7] C. Ding and X. He, "Linearized cluster assignment via spectral ordering," in *Proc. 21st Int. Conf. Mach. Learn.*, Banff, AB, Canada, Jul. 2004, pp. 30–37.
[8] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2002.
[9] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Spectral relaxation for k-means clustering," in *Advances in Neural Information Processing Systems 13*. Cambridge, MA: MIT Press, 2001, pp. 1057–1064.
[10] J. Zhang, K. Zhang, X. Xu, C. K. Tse, and M. Small, "Seeding the kernels in graphs: Toward multi-resolution community analysis," *New J. Phys.*, vol. 11, p. 113003, Nov. 2009.
[11] C. K. I. Williams and M. Seeger, "The effect of the input density distribution on kernel-based classifiers," in *Proc. 17th Int. Conf. Mach. Learn.*, Stanford, CA, Jun. 2000, pp. 1159–1166.
[12] S. Fine and K. Scheinberg, "Efficient SVM training using low-rank kernel representations," *J. Mach. Learn. Res.*, vol. 2, pp. 243–264, Dec. 2001.
[13] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2006.
[14] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, Jun. 1999.
[15] C. K. I. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems 13*. Cambridge, MA: MIT Press, 2001, pp. 682–688.
[16] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, Feb. 2004.
[17] T. Cox and M. Cox, *Multidimensional Scaling*. London, U.K.: Chapman & Hall, 1994.
[18] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[19] A. J. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning," in *Proc. 17th Int. Conf. Mach. Learn.*, Stanford, CA, Jun. 2000, pp. 911–918.
[20] S. Kumar, M. Mohri, and A. Talwalkar, "On sampling-based approximate spectral decomposition," in *Proc. Int. Conf. Mach. Learn.*, Montreal, QC, Canada, Jun. 2009, pp. 585–592.
[21] M. Ouimet and Y. Bengio, "Greedy spectral embedding," in *Proc. 10th Int. Workshop Artif. Intell. Statist.*, Barbados, Jan. 2005, pp. 253–260.
[22] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *J. Mach. Learn. Res.*, vol. 3, pp. 1–48, Jul. 2002.
[23] F. R. Bach and M. I. Jordan, "Predictive low-rank decomposition for kernel methods," in *Proc. 22nd Int. Conf. Mach. Learn.*, Bonn, Germany, Aug. 2005, pp. 33–40.
[24] D. Achlioptas and F. McSherry, "Fast computation of low rank matrix approximations," in *Proc. 33rd Annu. ACM Symp. Theory Comput.*, Hersonissos, Greece, Jul. 2001, pp. 611–618.
[25] P. Drineas, E. Drinea, and P. S. Huggins, "An experimental evaluation of a Monte-Carlo algorithm for singular value decomposition," in *Proc. 8th Panhellenic Conf. Informat.*, Nicosia, Cyprus, 2003, pp. 279–296.
[26] P. Drineas and M. W. Mahoney, "On the Nyström method for approximating a Gram matrix for improved kernel-based learning," *J. Mach. Learn. Res.*, vol. 6, pp. 2153–2175, Dec. 2005.
[27] A. Frieze, R. Kannan, and S. Vempala, "Fast Monte-Carlo algorithms for finding low-rank approximations," in *Proc. 39th Annu. Symp. Found. Comput. Sci.*, Palo Alto, CA, Nov. 1998, pp. 370–378.
[28] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang, "Parallel spectral clustering in distributed systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[29] J. C. Platt, "Fastmap, metricmap, and landmark MDS are all Nyström algorithms," in *Proc. 10th Int. Workshop Artif. Intell. Statist.*, Barbados, Jan. 2005, pp. 261–268.
[30] V. de Silva and J. B. Tenenbaum, "Global vs. local methods in nonlinear dimensionality reduction," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2003, pp. 705–712.
[31] A. Talwalkar, S. Kumar, and H. Rowley, "Large scale manifold learning," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, Jun. 2008, pp. 1–8.
[32] S. Kumar, M. Mohri, and A. Talwalkar, "Sampling techniques for the Nyström method," in *Proc. Int. Conf. Artif. Intell. Statist.*, Clearwater Beach, FL, Apr. 2009, pp. 304–311.
[33] K. Zhang, J. T. Kwok, and B. Parvin, "Prototype vector machine for large scale semi-supervised learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Montreal, QC, Canada, Jun. 2009, pp. 1233–1240.
[34] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK User Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*. Philadelphia, PA: SIAM, 1998.
[35] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.
[36] C. T. H. Baker, *The Numerical Treatment of Integral Equations*. Oxford, U.K.: Clarendon, 1977.
[37] S. Kumar, M. Mohri, and A. Talwalkar, "Ensemble Nyström method," in *Advances in Neural Information Processing Systems 22*. Cambridge, MA: MIT Press, 2009, pp. 1060–1068.
[38] M. Li, J. T. Kwok, and B.-L. Lu, "Making large-scale Nyström approximation possible," in *Proc. Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 1–8.
[39] K. Zhang and J. T. Kwok, "Block-quantized kernel matrix for fast spectral embedding," in *Proc. 23rd Int. Conf. Mach. Learn.*, Pittsburgh, PA, Jun. 2006, pp. 1097–1104.
[40] K. Zhang and J. T. Kwok, "Density-weighted Nyström method for computing large kernel eigensystems," *Neural Comput.*, vol. 21, no. 1, pp. 121–146, Jan. 2009.
[41] P. Drineas, R. Kannan, and M. W. Mahoney, "Fast Monte-Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix," *SIAM J. Comput.*, vol. 36, no. 1, pp. 158–183, 2006.
[42] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.
[43] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.
[44] D. Pelleg and A. Moore, "Accelerating exact k-means algorithms with geometric reasoning," in *Proc. 5th Int. Conf. Knowl. Discovery Data Mining*, San Diego, CA, Aug. 1999, pp. 277–281.
[45] E. Elkan, "Using the triangle inequality to accelerate k-means," in *Proc. 21th Int. Conf. Mach. Learn.*, Washington D.C., Aug. 2003, pp. 147–153.
[46] D. Lee, S. Baek, and K. Sung, "Modified k-means algorithm for vector quantizer design," *IEEE Signal Process. Lett.*, vol. 4, no. 1, pp. 2–4, Jan. 1997.
[47] I. S. Dhillon and D. S. Modha, "A data-clustering algorithm on distributed memory multiprocessors," in *Large-Scale Parallel Data Mining*. Berlin, Germany: Springer-Verlag, 2000, pp. 245–260.
[48] C. Walder, K. I. Kim, and B. Schölkopf, "Sparse multiscale Gaussian process regression," in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, Jun. 2008, pp. 1112–1119.
[49] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 1–8.
[50] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, 1962.
[51] K. Zhang, M. Tang, and J. T. Kwok, "Applying neighborhood consistency for fast clustering and kernel density estimation," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, Jun. 2005. pp. 1001–1007.

Kai Zhang received the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, China, in 2008.

He is now with the Life Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA. His current research interests include data mining applications including bioinformatics and complex networks, machine learning and pattern recognition, in particular large-scale unsupervised learning, semisupervised learning, and dimension reduction algorithms.

James T. Kwok received the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, China, in 1996.

He was with the Department of Computer Science and Engineering, Hong Kong Baptist University, Hong Kong, as an Assistant Professor. He returned to the Hong Kong University of Science and Technology in 2000, and is an Associate Professor in the Department of Computer Science and Engineering. His current research interests include kernel methods, machine learning, pattern recognition, and artificial neural networks.

Dr. Kwok received the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Award in 2004 and Paper Award in 2006. He serves as an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS and the *Neurocomputing Journal*.