

## Density-Weighted Nyström Method for Computing Large Kernel Eigensystems

**Kai Zhang**

*twinsen@cse.ust.hk*

**James T. Kwok**

*jamesk@cse.ust.hk*

*Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong*

The Nyström method is a well-known sampling-based technique for approximating the eigensystem of large kernel matrices. However, the chosen samples in the Nyström method are all assumed to be of equal importance, which deviates from the integral equation that defines the kernel eigenfunctions. Motivated by this observation, we extend the Nyström method to a more general, density-weighted version. We show that by introducing the probability density function as a natural weighting scheme, the approximation of the eigensystem can be greatly improved. An efficient algorithm is proposed to enforce such weighting in practice, which has the same complexity as the original Nyström method and hence is notably cheaper than several other alternatives. Experiments on kernel principal component analysis, spectral clustering, and image segmentation demonstrate the encouraging performance of our algorithm.

### 1 Introduction ---

Eigenvalue decomposition of the kernel matrix plays an important role in many machine learning and computer vision problems. For example, in kernel principal component analysis (KPCA) (Schölkopf, Smola, & Müller, 1998), the eigenvectors of the kernel matrix are used to extract nonlinear structures in the high-dimensional feature space. In spectral clustering (Shi & Malik, 2000; Fowlkes, Belongie, Chung, & Malik, 2004; Ng, Jordan, & Weiss, 2002), the eigenvectors of the (normalized) kernel matrix provide an approximate solution for the NP-hard clustering problem. The eigenstructures of the kernel matrix are also quite useful in kernel design (Chapelle, Weston, & Schölkopf, 2003; Smola & Kondor, 2003), semisupervised learning (Zhang & Ando, 2006), manifold learning, and embedding (Belkin & Niyogi, 2002).

However, eigenvalue decomposition of an  $n \times n$  kernel matrix takes  $O(n^3)$  time and is computationally prohibitive. This poses a big challenge in applying the technique to large-scale problems. In this letter, we focus on

a class of sampling-based approximation techniques that have been widely used in machine learning. Among them, a well-known approach is the Nyström method, which originated from the numerical treatment of the integral equation (Baker, 1977):

$$\int p(y)k(x, y)\phi(y) dy = \lambda\phi(x). \quad (1.1)$$

Here,  $k(\cdot, \cdot)$  is the kernel function, which is usually positive semidefinite;  $p(\cdot)$  is the underlying probability density function; and  $\lambda$  and  $\phi(\cdot)$  are the eigenvalue and eigenfunction of the kernel  $k$ , respectively. As will be clear in section 2, the Nyström method can be deemed as selecting a random subset of samples to approximate the integral 1.1, or, equivalently, as choosing a random subset of rows or columns of the full kernel matrix to approximate its eigensystem. It has been successfully used in speeding up kernel machines (Williams & Seeger, 2001) and spectral clustering (Fowlkes et al., 2004). Platt (2005) further showed that several forms of multidimensional scaling (Cox & Cox, 1994), including the landmark MDS (Silva & Tenenbaum, 2003), FastMap (Faloutsos & Lin, 1995), and MetricMap (Wang et al., 1999), are all variants of the Nyström method.

Besides random sampling, more sophisticated sampling schemes have also been pursued recently with the Nyström method. In Drineas and Mahoney (2005), the rows and columns of the kernel matrix are chosen based on a nonuniform, data-dependent probability distribution, which leads to a provable probabilistic bound. Ouimet and Bengio (2005) proposed a greedy sampling scheme based on the feature space geometry. However, these algorithms are usually more expensive. For example, the sampling probabilities in Drineas and Mahoney (2005) are computed based on the norms of all the rows and columns of the Gram matrix, which takes  $O(n^2)$  time. The greedy scheme in Ouimet and Bengio (2005) has a time complexity of  $O(m^2n)$  for the sampling step, where  $m$  is the number of representatives chosen. In comparison, the (random) sampling step in the Nyström method is much cheaper.

While most works on the Nyström method focus on the design of sampling schemes, we find that the basic formulation may have a key inefficiency: that the chosen samples (also called the landmark points) are all assumed to be of equal importance. This deviates from the integral equation 1.1, where a density-based weighting  $p(y)$  is imposed on the integration variable  $y$ . In the following, we illustrate the importance of the density-based weighting with a numerical example. Suppose the density is a univariate gaussian  $p(x) = (2a/\pi)^{\frac{1}{2}} \exp(-2ax^2)$ , and the gaussian kernel  $k(x, y) = \exp(-b(x - y)^2)$  is used. In this case, the eigenfunctions can be obtained analytically as  $\phi(x) = \exp(-(c - a)x^2) H_k(\sqrt{2c}x)$ , where  $c = \sqrt{a^2 + 2ab}$  and  $H_k(x) = \frac{k!}{2^k \pi^{1/2}} \int e^{-t^2 + 2tx} t^{-k-1} dt$  is the  $k$ th order Hermite

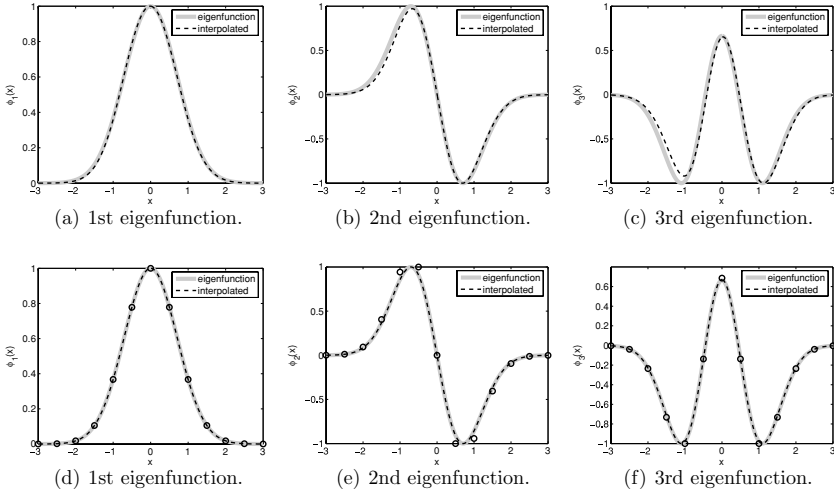


Figure 1: Approximations of the eigenfunctions using the original Nyström method with  $n = 500$  samples (top) and the weighted Nyström method using  $m = 13$  landmark points (bottom; landmark points marked by circles). The thick gray curve is the true eigenfunction.

expansion (Williams & Seeger, 2000). We choose  $a = 1/4$  (i.e.,  $p(x)$  has the unit standard deviation) and  $b = 1$ . We first apply the original Nyström method to approximate the eigenfunction, using  $n = 500$  landmark points randomly drawn from  $p(x)$ . Then we apply the density-based weighting  $p(x)$  on the landmark points in the Nyström method (details are given in section 3), but this time using only  $m = 13$  landmark points uniformly distributed in  $[-3, 3]$ . Results are plotted in Figure 1. As can be seen, the weighted version, though using many fewer landmark points, gives better performance than the original Nyström method.

To apply the weighted Nyström formulation in practice, we need knowledge of the underlying probability density function, which might be difficult to obtain. To avoid this problem, we resort to a novel block-quantization scheme of the kernel matrix. The resultant algorithm, which is called the weighted Nyström method, is computationally very efficient and demonstrates superior performance in our numerical evaluations.

The rest of the letter is organized as follows. We first briefly introduce the Nyström method in section 2. Then we extend it to the weighted version in section 3. In section 4 we propose a block-quantization scheme of kernel matrices. We show that such a quantization scheme is actually a special case of the weighted Nyström formulation, which provides important insights into how to choose the landmark points and their associated weighting coefficients. In section 5, we discuss the application of the weighted Nyström

method in speeding up the normalized cut. Section 6 provides several discussions on the weighted Nyström method. Experimental results on using our approach for KPCA, spectral clustering, and image segmentation are presented in section 7, and the last section gives concluding remarks. Our preliminary work is reported in Zhang and Kwok (2006).

## 2 The Nyström Method

---

The Nyström method was originally designed for numerical treatment of integral equations like equation 1.1. Based on a set of independent and identically distributed (i.i.d.) samples  $X = \{x_i\}_{i=1}^n$  that are supposed to be drawn from the probability density  $p(\cdot)$ , the integral can be approximated by

$$\lambda\phi(x) = \int p(y)k(x, y)\phi(y) dy \simeq \frac{1}{n} \sum_{j=1}^n k(x, x_j)\phi(x_j). \quad (2.1)$$

When  $x$  in equation 2.1 goes through all  $x_i$ 's in  $X$ , we obtain  $n$  linear equations:

$$\frac{1}{n} \sum_{j=1}^n k(x_i, x_j)\phi(x_j) = \lambda\phi(x_i), \quad i = 1, 2, \dots, n,$$

which can be written as the eigenvalue decomposition:

$$K\phi = n\lambda\phi. \quad (2.2)$$

Here  $K \in \mathbb{R}^{n \times n}$  is the kernel matrix such that  $K_{ij} = K(x_i, x_j)$ , and  $\phi \in \mathbb{R}^n$  is the corresponding eigenvector. After equation 2.2 is solved, the eigenfunction  $\phi(\cdot)$  at any point  $x$  can be approximately evaluated using equation 2.1:

$$\phi(x) \approx \frac{1}{n\lambda} \sum_{j=1}^n k(x, x_j)\phi(x_j).$$

The eigenvalue decomposition in equation 2.2 scales cubically with the sample size. To reduce the time complexity, one may use only a random subset of the available samples, which leads to a much smaller eigenvalue problem. This is commonly known as the Nyström method (Williams & Seeger, 2001; Fowlkes et al., 2004) and is summarized in algorithm 1:

## Algorithm 1: The Nyström Method

1. Given the sample set  $X = \{x_i\}_{i=1}^n$ , randomly choose a subset  $Z = \{z_k\}_{k=1}^m$ .
2. Compute the kernel submatrix  $W \in \mathbb{R}^{m \times m}$ :

$$W_{ij} = k(z_i, z_j). \quad (2.3)$$

3. Perform the eigenvalue decomposition  $W\phi_Z = m\lambda_Z\phi_Z$ , with eigenvector  $\phi_Z \in \mathbb{R}^m$  and eigenvalue  $m\lambda_Z$ .
4. Compute the extrapolation matrix  $E \in \mathbb{R}^{n \times m}$ :

$$E_{ij} = k(x_i, z_j). \quad (2.4)$$

5. Extrapolate  $\phi_Z \in \mathbb{R}^m$  to  $\phi_X \in \mathbb{R}^n$  by  $\phi_X = (m\lambda_Z)^{-1}E\phi_Z$ .

In algorithm 1, the subscript  $Z$  is to emphasize that the evaluation is with regard to the landmark point set  $Z$ , and  $\phi_X \in \mathbb{R}^{n \times 1}$  represents the (approximate) evaluations of the kernel eigenfunction  $\phi(\cdot)$  on the whole sample set  $X = \{x_i\}_{i=1}^n$ . Note that  $\phi_X$  will approximate the eigenvectors of the complete kernel matrix after some scaling (Williams & Seeger, 2001). Therefore in the sequel, we simply focus on the derivation of  $\phi_X$ .

### 3 Density-Weighted Nyström Method

---

In this section, we extend the Nyström method to a more general setting based on the integral equation, 1.1. As has been discussed in section 1, the original Nyström method assigns equal importance to all the chosen samples. This deviates from the integral equation 1.1, where the integration variable is weighted by the density function. Here, we explicitly introduce this density function  $p(\cdot)$  evaluated at the landmark points  $Z = \{z_i\}_{i=1}^m$ . Then the integral equation 1.1 can be approximated as

$$\lambda\tilde{\phi}(x) = \int p(y)k(x, y)\tilde{\phi}(y)dy \simeq \frac{1}{c} \sum_{i=1}^m p(z_i)k(x, z_i)\tilde{\phi}(z_i),$$

where  $c = \sum_{i=1}^m p(z_i)$  is the normalization factor. Here the symbol  $\tilde{\cdot}$  denotes entities corresponding to this weighted version. By choosing  $x$  at the landmark points, we have

$$\tilde{W}\tilde{\phi}_Z = c\tilde{\lambda}_Z\tilde{\phi}_Z, \quad (3.1)$$

where  $\tilde{W} \in \mathbb{R}^{m \times m}$  is the density-weighted kernel matrix evaluated at the landmark points,

$$\tilde{W}_{ij} = p(z_j)k(z_i, z_j), \quad (3.2)$$

and  $\tilde{\phi}_Z \in \mathbb{R}^m$  and  $c\tilde{\lambda}_Z$  are the corresponding eigenvector and eigenvalue, respectively. After the eigensystem 3.1 is solved, the eigenfunction can be evaluated at an arbitrary point  $x$  as

$$\phi(x) \approx \frac{1}{c\tilde{\lambda}_Z} \sum_{j=1}^m p(z_j)k(x, z_j)\tilde{\phi}(z_j)$$

or in matrix form

$$\phi_X = (c\tilde{\lambda}_Z)^{-1}\tilde{E}\tilde{\phi}_Z,$$

where  $\tilde{E} \in \mathbb{R}^{n \times m}$  is the density-weighted extrapolation matrix,

$$\tilde{E}_{ij} = p(z_j)k(x_i, z_j). \quad (3.3)$$

We can see that a major difference between the density-weighted Nyström formulation and the original one is that the probability density function appears explicitly. Both the kernel matrix to be diagonalized ( $W$  in equation 2.3), and the extrapolation matrix ( $E$  in equation 2.4) are weighted by the density  $p(\cdot)$  as shown in equations 3.2 and 3.3, respectively. In the next section, we discuss how to choose landmark points  $Z$  and the density-based weighting  $p(z_k)$ 's through a matrix quantization view.

## 4 Kernel Matrix Quantization

---

This section presents a novel block-quantization scheme of the kernel matrix, which will be shown to be a special case of the weighted Nyström formulation. It provides important insights on how to choose the density-based weighting coefficients  $p(z_k)$ 's (section 4.1) and the landmark points  $z_k$ 's (section 4.2).

**4.1 Block Quantization.** Unlike common quantization methods, which sparsify the matrix by zeroing out small entries, our quantization scheme takes advantage of the special pairwise structure of the kernel matrix. The basic idea is to partition the data set  $X$  into disjoint clusters  $S_k$ 's ( $k = 1, 2, \dots, m$ ), each with cluster size  $|S_k|$  and cluster representative  $z_k$ . Then, for any two points  $x_i \in S_p$  and  $x_j \in S_q$ , the kernel evaluation  $k(x_i, x_j)$  is approximated by using the corresponding cluster representatives as  $k(x_i, x_j) = k(z_p, z_q)$ . Without loss of generality, suppose that the data set is ordered such that the first  $|S_1|$  samples belong to cluster  $S_1$ , the following  $|S_2|$  samples belong to cluster  $S_2, \dots$ , and the last  $|S_m|$  samples belong to cluster  $S_m$ . The  $n \times n$  kernel matrix  $K$  can then be quantized into  $m \times m$

constant blocks. We use  $\bar{W}$  to denote this block-wise constant matrix, with

$$\bar{W}_{ij} = k(z_p, z_q), \quad x_i \in S_p, x_j \in S_q, \quad 1 \leq i, j \leq n, \quad 1 \leq p, q \leq m. \quad (4.1)$$

In the sequel, the symbol  $\bar{\cdot}$  denotes entities that are related to this block-wise constant matrix. The following is an illustrative example where the data are grouped into two clusters, with  $|S_1| = 2$  and  $|S_2| = 3$ :

$$\bar{W} = \begin{pmatrix} a & a & b & b & b \\ a & a & b & b & b \\ c & c & d & d & d \\ c & c & d & d & d \\ c & c & d & d & d \end{pmatrix}.$$

The following proposition shows that the eigenvectors of  $\bar{W}$  are piecewise constant and can be simply computed by decomposing a much smaller matrix  $\tilde{W}$  in  $O(m^3)$  time. As we shall see, this is the key to the computational advantage of the weighted Nyström method. The proof can be found in appendix A.

**Proposition 1.** *Define a matrix based on the cluster centers  $z_k$ 's and cluster sizes  $|S_k|$ 's,  $\tilde{W} \in \mathbb{R}^{m \times m}$ , such that*

$$\tilde{W}_{pq} = k(z_p, z_q)|S_q|, \quad p, q = 1, 2, \dots, m, \quad (4.2)$$

and let  $\tilde{\phi} \in \mathbb{R}^m$  be the corresponding eigenvector. Then the eigenvector  $\bar{\phi} \in \mathbb{R}^n$  of the block-wise constant kernel matrix  $\bar{W}_{n \times n}$  in equation 4.1 can be obtained as

$$\bar{\phi}(i) = \tilde{\phi}(k), \quad \forall x_i \in S_k,$$

that is,  $\bar{\phi}$  is obtained by repeating the  $k$ th entry of  $\tilde{\phi}$  ( $1 \leq k \leq m$ )  $|S_k|$  times and then concatenating them together.

The block-quantization procedure is actually a weighted Nyström formulation. First, note that the block quantization depends on grouping the sample set  $X$  into clusters. These clusters can be considered as the bins in a multivariate histogram, and the bin heights can be chosen as the (normalized) cluster sizes  $\frac{1}{n}|S_k|$ 's. Now, by using  $p(z_k) = \frac{1}{n}|S_k|$  and  $c = \sum_{k=1}^m p(z_k) = 1$ , the eigenvalue decomposition in the weighted Nyström formulation 3.1 can be written as  $\tilde{W}\tilde{\phi}_Z = \lambda_Z\tilde{\phi}_Z$ , with  $\tilde{W}_{pq} = K(z_p, z_q)|S_q|/n$ . This is almost identical to eigenvalue decomposition 4.2 in the block-quantization procedure except for a scaling of the eigenvalues. Therefore, the weighted Nyström formulation, when using the normalized histogram

as the density estimator, is equivalent to the block quantization of the kernel matrix. In other words, the block-quantization procedure is a special case of the weighted Nyström formulation.

**4.2 Choosing the Landmark Points.** In this section, we consider how to choose the landmark points  $z_k$ 's in the weighted Nyström method. The basic idea is to choose  $z_k$ 's such that the resultant, block-wise constant kernel matrix  $\bar{W}$  is close to the original kernel matrix  $K$ . We do so by minimizing the Frobenius norm of the difference between them. With regard to this, we have the following proposition (the proof can be found in appendix B):

**Proposition 2.** *Suppose that the data set  $X = \{x_i\}_{i=1}^n$  is partitioned into  $m$  disjoint clusters  $S_k$ 's with cluster center  $z_k$ 's,  $k = 1, 2, \dots, m$ , and the block-wise constant kernel matrix, equation 4.1 is obtained by replacing each sample  $x_i$  with its corresponding cluster center  $z_{c(i)}$ . Then, with the use of the stationary kernel  $k(x, y) = k(\|x - y\|^2/\sigma^2)$ , we have*

$$\|K - \bar{W}\|_F \leq 8 \frac{\xi^2}{\sigma^4} \left( nR^2 \overline{D^{(2)}} + nR \overline{D^{(3)}} + \frac{1}{4} n \overline{D^{(4)}} + R^2 (\overline{D^{(1)}})^2 + \frac{3}{4} (\overline{D^{(2)}})^2 + 3R \overline{D^{(2)}} \overline{D^{(1)}} + \overline{D^{(3)}} \overline{D^{(1)}} \right),$$

where  $R$  is the maximum pairwise distance between samples,  $\xi = \max_x |k'(x)|$ , and  $\overline{D^{(k)}}$ 's are distortion errors under different norms defined by

$$\overline{D^{(k)}} = \sum_{i=1}^n \|x_i - z_{c(i)}\|^k$$

for  $k = 1, 2, 3, 4$ .

From proposition 2, we can see that  $\|K - \bar{W}\|_F$  is determined by  $\overline{D^{(k)}}$ 's ( $k = 1, 2, 3, 4$ ), which are the distortion errors of quantizing each point with the corresponding cluster center. The lower the distortion error, the smaller the  $\|K - \bar{W}\|_F$ . In the limiting case where each sample is chosen as a landmark point and considered as one cluster,  $\overline{D^{(k)}}$ 's are all zero, and, hence, the Frobenius norm error is also zero.

Note that when  $k = 2$ ,  $\overline{D^{(k)}}$  is exactly the objective of the  $k$ -means clustering algorithm, which can find a local minimum of the  $\overline{D^{(k)}}$  (Gersho & Gray, 1992). Therefore, we choose the  $k$ -means algorithm to partition the data set and use the resultant cluster centers as the landmark points  $z_k$ 's. Note that the  $k$ -means algorithm is easy to implement, and the complexity is linear with the sample size and dimension. There are also several recent advances on scaling up the  $k$ -means algorithm (Pelleg & Moore, 1999;



Elkan, 2003; Kanungo et al., 2002). Therefore, the  $k$ -means-based sampling strategy is suitable for solving large-scale problems in this context. The complete weighted Nyström method for approximating the kernel eigenfunctions is shown in algorithm 2:

**Algorithm 2: The Weighted Nyström Method**

1. Given the sample set  $X = \{x_i\}_{i=1}^n$ , use  $k$ -means to group it into  $m$  clusters, with cluster centers  $z_k$ 's and cluster sizes  $|S_k|$ 's,  $k = 1, 2, \dots, m$ .
2. Compute the weighted kernel submatrix  $\tilde{W} \in \mathbb{R}^{m \times m}$ ,  $\tilde{W}_{ij} = k(z_i, z_j)|S_j|$ .
3. Perform the eigenvalue decomposition  $\tilde{W}\tilde{\phi}_Z = n\tilde{\lambda}_Z\tilde{\phi}_Z$ , with eigenvector  $\tilde{\phi}_Z \in \mathbb{R}^m$  and eigenvalue  $n\tilde{\lambda}_Z$ .
4. Compute the weighted extrapolation matrix  $\tilde{E} \in \mathbb{R}^{n \times m}$ ,  $\tilde{E}_{ij} = k(x_i, z_j)|S_j|$ .
5. Extrapolate  $\tilde{\phi}_Z \in \mathbb{R}^m$  to  $\phi_X \in \mathbb{R}^n$  by  $\phi_X = (n\tilde{\lambda}_Z)^{-1}\tilde{E}\tilde{\phi}_Z$ .

Note that the weighted Nyström method needs to decompose an asymmetric matrix  $\tilde{W}$  (step 3). However,  $\tilde{W}$  can be written as  $\tilde{W} = WP$ , where  $W \in \mathbb{R}^{m \times m}$  is a symmetric kernel matrix defined on the landmark set  $Z$  and  $P \in \mathbb{R}^{m \times m}$  is a diagonal matrix such that  $P_{kk} = p(z_k)$ . Therefore we can apply a simple transform  $u = P^{-\frac{1}{2}}\tilde{\phi}_Z$  and turn step 3 into a symmetric eigenvalue problem (Press, Teukolsky, Vetterling, & Flannery, 1992). This is numerically much easier than decomposing asymmetric matrices.

Considering that random initialization of the  $k$ -means introduces some statistical variability and that we may prefer scanning the data only once for a large data set rather than iterate many steps, we propose an efficient procedure to partition the data in one round, which is called *sequential sampling* in algorithm 4 (see appendix C). Its complexity is only  $O(mn)$  since the algorithm requires only one pass of the data, and when a hierarchical scheme is used (Feder & Greene, 1988), it can be further reduced to  $O(n \log m)$ . We can also use sequential sampling to initialize the  $k$ -means procedure.

## 5 Application in Normalized Cut

---

The normalized cut (Ncut) (Shi & Malik, 2000) is the most popular spectral method (Ng et al., 2002) and has been widely used in clustering problems such as speech separation (Bach & Jordan, 2006), image, and motion segmentation (Fowlkes et al., 2004; Shi & Malik, 2000). In this section, we discuss how to apply the weighted Nyström formulation to speed up eigenvalue decomposition in the normalized cut.

The key step of normalized cut is to solve the following eigenvalue problem:

$$D^{-1/2}KD^{-1/2}z = \lambda z. \quad (5.1)$$

Here  $K \in \mathbb{R}^{n \times n}$  is the similarity (or adjacency) matrix, and  $D \in \mathbb{R}^{n \times n}$  is the diagonal degree matrix whose  $i$ th entry is the sum of the  $i$ th row of  $K$ . After obtaining the eigenvector corresponding to the second-largest eigenvalue of equation 5.1, and transforming it by  $D^{-1/2}z$ , we can obtain a relaxed solution of the normalized cut partition (Shi & Malik, 2000). Note that the integral equation corresponding to equation 5.1 is

$$\int D^{-\frac{1}{2}}(x)k(x, y)p(y)D^{-\frac{1}{2}}(y)\phi(y) dy = \lambda\phi(x), \quad (5.2)$$

where

$$D(x) = \int k(x, y)p(y) dy. \quad (5.3)$$

We now consider how to obtain an approximate solution of equation 5.2 using the weighted Nyström formulation. Similar to section 3, we choose both  $x$  and  $y$  in equation 5.2 at a set of landmark points  $Z = \{z_i\}_{i=1}^m$  and obtain the following set of linear equations:

$$\frac{1}{c} \sum_{q=1}^m D^{-\frac{1}{2}}(z_p)k(z_p, z_q)p(z_q)D^{-\frac{1}{2}}(z_q)\phi(z_q) = \lambda\phi(z_p), \quad p=1, 2, \dots, m, \quad (5.4)$$

where  $c = \sum_{q=1}^m p(z_q)$ . Note that  $D(z_k)$  ( $k = 1, 2, \dots, m$ ) can be computed by the discrete counterpart of equation 5.3, as

$$D(z_k) = \frac{1}{c} \sum_{i=1}^m k(z_k, z_i)p(z_i). \quad (5.5)$$

Plugging equation 5.5 into equation 5.4, we have

$$\begin{aligned} & \frac{1}{c} \sum_{q=1}^m \left( \frac{1}{c} \sum_{i=1}^m k(z_p, z_i)p(z_i) \right)^{-\frac{1}{2}} k(z_p, z_q)p(z_q) \left( \frac{1}{c} \sum_{i=1}^m k(z_q, z_i)p(z_i) \right)^{-\frac{1}{2}} \\ & \times \phi(z_q) = \lambda\phi(z_p), \end{aligned}$$

or in matrix form,

$$D_Z^{-\frac{1}{2}} \tilde{W} D_Z^{-\frac{1}{2}} \tilde{\phi}_Z = \lambda \tilde{\phi}_Z,$$

where  $\tilde{W}$  is the same as defined in equation 3.2,  $\tilde{\phi}_Z \in \mathbb{R}^m$  is the eigenvector, and  $D_Z$  is the diagonal matrix whose  $p$ th diagonal entry is

$$[D_Z]_{p,p} = \sum_{k=1}^m k(z_p, z_k) p(z_k), \quad p = 1, 2, \dots, m. \quad (5.6)$$

Note that  $D_Z$  is actually the degree matrix of  $\tilde{W}$ , in that the  $k$ th diagonal entry of  $D_Z$  is the sum of the  $k$ th row of  $\tilde{W}$ .

To evaluate the eigenfunction at any point  $x$ , we use the discrete approximations of equations 5.2 and 5.3 to obtain

$$\phi(x) = \frac{1}{c\lambda} \sum_{q=1}^m D^{-\frac{1}{2}}(x) k(x, z_q) p(z_q) D^{-\frac{1}{2}}(z_q) \phi(z_q), \quad (5.7)$$

$$D(x) = \frac{1}{c} \sum_{i=1}^m k(x, z_i) p(z_i). \quad (5.8)$$

Plugging equations 5.5 and 5.8 into 5.7, we have

$$\begin{aligned} \phi(x) &= \frac{1}{c\lambda} \sum_{q=1}^m \left( \frac{1}{c} \sum_{i=1}^m k(x, z_i) p(z_i) \right)^{-\frac{1}{2}} k(x, z_q) p(z_q) \\ &\quad \times \left( \frac{1}{c} \sum_{i=1}^m k(z_q, z_i) p(z_i) \right)^{-\frac{1}{2}} \phi(z_q). \end{aligned}$$

The eigenvector of the full sample set  $X$  can thus be obtained as

$$\phi_X = \frac{1}{\lambda} D_X^{-\frac{1}{2}} \tilde{E} D_Z^{-\frac{1}{2}} \phi_Z,$$

where  $\tilde{E}$  is the same as defined in equation 3.3, and  $D_X$  is the diagonal matrix whose  $i$ th diagonal entry ( $i = 1, 2, \dots, n$ ) is

$$[D_X]_{i,i} = \sum_{k=1}^m k(x_i, z_k) p(z_k) \quad i = 1, 2, \dots, n. \quad (5.9)$$

Note that  $D_X$  is actually the degree matrix of  $\tilde{E}$ , in that the  $k$ th diagonal entry of  $D_X$  is the sum of the  $k$ th row of the matrix  $\tilde{E}$ .

The normalized cut algorithm using weighted Nyström approximation is summarized in algorithm 3 (for notational clarity, we use capital letters for matrices of (multiple) eigenvectors and eigenvalues):

Algorithm 3: Normalized Cut Through Weighted Nyström Method

1. Perform steps 1, 2, and 4 in algorithm 2.
2. Compute  $D_Z$  (see equation 5.6) and  $D_X$  (see equation 5.9).
3. Perform the eigenvalue decomposition  $D_Z^{-\frac{1}{2}} \widetilde{W} D_Z^{-\frac{1}{2}} U_1 = U_1 \Lambda_1$ , with eigenvector matrix  $U_1 \in \mathbb{R}^{m \times m}$  and diagonal eigenvalue matrix  $\Lambda_1 \in \mathbb{R}^{m \times m}$ .
4. Extrapolate  $U_1 \in \mathbb{R}^{m \times m}$  to  $\bar{U} \in \mathbb{R}^{n \times m}$  by  $\bar{U} = D_X^{-\frac{1}{2}} \widetilde{E} D_Z^{-\frac{1}{2}} U_1 \Lambda_1^{-1}$ .
5. From  $\bar{U}$ , choose the eigenvector of the second-largest eigenvalue,  $\bar{u}$ , map it to  $D_X^{-1/2} \bar{u}$ , threshold, and obtain the normalized cut.

Note that the matrix to be decomposed in algorithm 3,  $D_Z^{-\frac{1}{2}} \widetilde{W} D_Z^{-\frac{1}{2}}$ , is asymmetric, and the corresponding eigenvectors might be complex. However, as will be shown in proposition 3, the eigensystem must be real since it can be represented by that of a symmetric matrix. Proof can be found in appendix D.

**Proposition 3.** *Define the diagonal matrix  $P \in \mathbb{R}^{m \times m}$  such that  $P_{kk} = p(z_k)$ ,  $k = 1, 2, \dots, m$ . Then the eigensystems associated with the matrices  $D_Z^{-\frac{1}{2}} \widetilde{W} D_Z^{-\frac{1}{2}}$  and  $(D_Z^P)^{-\frac{1}{2}} (PWP) (D_Z^P)^{-\frac{1}{2}}$ ,*

$$D_Z^{-\frac{1}{2}} \widetilde{W} D_Z^{-\frac{1}{2}} U_1 = U_1 \Lambda_1,$$

$$(D_Z^P)^{-\frac{1}{2}} (PWP) (D_Z^P)^{-\frac{1}{2}} U_2 = U_2 \Lambda_2,$$

will satisfy  $\Lambda_1 = \Lambda_2$  and  $U_1 = P^{-\frac{1}{2}} U_2$ .

## 6 Discussion

---

**6.1 Relationship with Quadrature Method.** The quadrature rule (Davis & Robinowitz, 1984) is widely used for numerical approximations of integrals. It is of the general form

$$\int_a^b F(s) ds \simeq \sum_{i=1}^n w_i F(s_j),$$

where  $w_j$ 's are the quadrature weights and  $s_j$ 's are the quadrature points or nodes. On using the quadrature rule to approximate the integral in

equation 1.1, we have

$$\sum_{j=1}^n w_j k(x, s_j) \phi(s_j) = \lambda \phi(x).$$

By choosing  $x$  at the nodes, we obtain the linear system

$$K\mathcal{W}\phi = \lambda\phi,$$

where  $K$  is the kernel matrix defined on the nodes and  $\mathcal{W}$  is the diagonal matrix containing the weights  $w_j$ 's. Different quadrature rules will enforce different choice of weights, such as the Simpson's rule or the trapezoid rule.

Note that the standard Nyström method and the density-weighted version can also be deemed as quadrature methods in approximating the integral equation. However, unlike commonly used quadrature rules, both methods actually enforce some kind of data-dependent quadrature rules (since the integral to be approximated is data dependent). This is reflected in the way they treat the density  $p(\cdot)$ : the standard Nyström method implicitly approximates the density through the samples that are drawn from  $p(\cdot)$  while the weighted Nyström method directly estimates the density by the normalized data histogram.

**6.2 Probabilistic Sampling.** Recently a probabilistic sampling scheme has been proposed for the Nyström method in Drineas and Mahoney (2005). Given an  $n \times n$  kernel matrix  $K$ , a prespecified integer  $m \leq n$ , and a distribution  $\{p_i\}_{i=1}^n$ , the algorithm proceeds as follows:

1. Pick with replacement  $m$  i.i.d. columns of  $K$  with respect to the probabilities  $\{p_i\}_{i=1}^n$ . Let  $I$  be the set of indices of the sampled columns obtained.
2. Scale each sampled column by dividing its elements by  $\sqrt{mp_i}$ .
3. Let  $C$  be the  $n \times m$  matrix containing the sampled columns after rescaling and  $W$  be the  $m \times m$  submatrix of  $K$  whose entries are  $W_{ij} = K_{ij} / \sqrt{mp_i p_j}$  for  $i, j \in I$ .
4. Compute  $W_k$ , the best rank- $k$  approximation to  $W$ .
5. Return  $K_k = CW_k^+ C'$ .

Typically, the distribution  $\{p_i\}_{i=1}^n$  is computed based on the norms of the rows or columns of the kernel matrix  $K$  as  $p_i = \frac{\|K^{(i)}\|_2}{\|K\|_F}$ , where  $K^{(i)}$  denotes the  $i$ th row of  $K$ .

The algorithm can be deemed as approximating the eigenvectors of the kernel matrix  $K$  with  $\bar{U} = CU\Lambda^{-1}$ , where  $U$  contains the eigenvectors of  $W$  as columns and  $\Lambda$  contains the eigenvalue of  $W$  in its diagonal (i.e.,  $W = U\Lambda U'$ ). To see this, note that reconstruction of  $K$  using  $\bar{U}\Lambda\bar{U}'$  will lead to  $K \approx \bar{U}\Lambda\bar{U}' = CU\Lambda^{-1}\Lambda\Lambda^{-1}U'C' = CW^{-1}C'$ , which is quite similar to step

5 except that the matrix inverse is replaced by the pseudo-inverse there for numerical stability. Therefore, it is a variant of the Nyström method. The primary difference is that sampling of the rows or columns is based on a precomputed distribution, and  $W$  and  $C$  are also reweighted by these probabilities. It is worthwhile to note that the reweighting here is quite different from our density-based weighting, and in the experimental evaluations (see section 7), we will see that the density-based weighting will give a better approximation.

**6.3 Complexity.** Suppose the sample size is  $n$  and the number of landmark points is  $m$ . Then the original Nyström method (see algorithm 1) has a complexity  $O(m^3 + nm)$ , where the first term is for the eigenvalue decomposition of the  $m \times m$  kernel matrix  $W$  and the second term is for the Nyström extension. For the weighted Nyström method, we need to perform  $k$ -means to partition the data set, which takes an extra  $O(mnl)$  time, where  $l$  is the number of  $k$ -means iterations. Since we set  $l$  as a constant (10 in all our experiments), the overall complexity of the weighted Nyström method is still  $O(m^3 + mn)$ . This is the same as the original Nyström method and is notably cheaper than several other related algorithms mentioned in section 1.

In applying the weighted Nyström method to the NCut, some extra computations are needed for estimating the degree matrices  $D_Z$  and  $D_X$ . The former is computed based on the row sums of the  $n \times m$  matrix  $\tilde{E}$  and the latter on the  $m \times m$  matrix  $\tilde{W}$ . Therefore,  $O(mn)$  time is further needed, which, however, does not change the complexity of  $O(m^3 + mn)$ .

Note that the approximate eigenvectors obtained by the Nyström method (see algorithm 1) may not satisfy the orthogonality constraints. An orthogonalization step can therefore be applied (Fowlkes et al., 2004), which improves the approximation performance but at the cost of increasing the overall complexity from  $O(m^3 + mn)$  to  $O(m^2n)$ . For simplicity, in this letter, we make comparisons between different algorithms without requiring the orthogonality condition. Note that orthogonalization of the approximate eigenvectors in the weighted Nyström method will involve developing its matrix completion view and will be considered in the future.

## 7 Experiments

---

In this section, we evaluate the performance of the different variants of the Nyström method for eigenapproximation. Experiments include kernel principal component analysis (section 7.1), spectral clustering, and image segmentation (section 7.2).

**7.1 Eigenvalue Decomposition of the Kernel Matrix.** We first evaluate the proposed algorithms by performing eigenvalue decomposition of the kernel matrix and examining the approximation qualities of the obtained

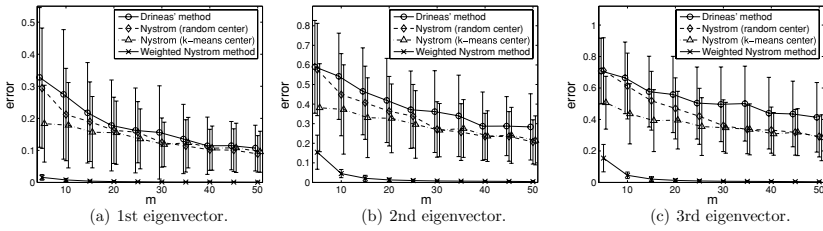


Figure 2: Approximation errors of the top three eigenvectors on the 1D gaussian data set.

eigenvectors. This is the first step of the kernel principal component analysis.

The first data set has  $n = 500$  samples drawn from the 1D normal distribution  $p(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$ . The gaussian kernel  $k(x, y) = \exp(-b(x - y)^2)$  is used with  $b = 1$ . We compare the Nyström method using (1) randomly chosen landmark points; (2)  $k$ -means-based landmark points; (3) weighted,  $k$ -means-based landmark points (our approach); and (4) probabilistic sampling by Drineas and Mahoney (2005). We gradually increase the subset size  $m$  from 5 to 50, and for each  $m$ , all the algorithms are repeated 100 times with the average performance reported.

Figure 2 shows the average approximation errors (and the 1-standard-deviation error bars<sup>1</sup>) on the first three eigenvectors of the kernel matrix  $K$ . Here the error is computed as the  $L_2$ -distance between the original and the approximate eigenvectors, both of which are scaled to have norm 1. As can be seen, the approximation errors obtained by the weighted Nyström method are much lower than those by others. Indeed, on average, the probabilistic sampling scheme in Drineas and Mahoney (2005) performs even worse than random sampling, which is also observed in the context of singular value decomposition (Drineas, Drinea, & Huggins, 2003).

The second data set is from the digits 0 and 1 of the MNIST data set.<sup>2</sup> We randomly choose 2000 images for training. Following Ouimet and Bengio (2005), we use the gaussian kernel with bandwidth  $\sigma = 31.6$ . We gradually increase  $m$  from 10 to 200, and for each  $m$ , all the algorithms are repeated 30 times. Figure 3 plots the errors of the top three eigenvectors obtained by the different algorithms. Again, we can see that our method is more accurate, and the leading eigenvectors are usually better approximated than the trailing ones. Another observation is that the direct use of  $k$ -means for choosing the landmark points in the Nyström method, although it slightly improves

<sup>1</sup>Note that the error curves are slightly offset so that the error bars can be more easily identified.

<sup>2</sup>Available online at: <http://yann.lecun.com/exdb/mnist/>.

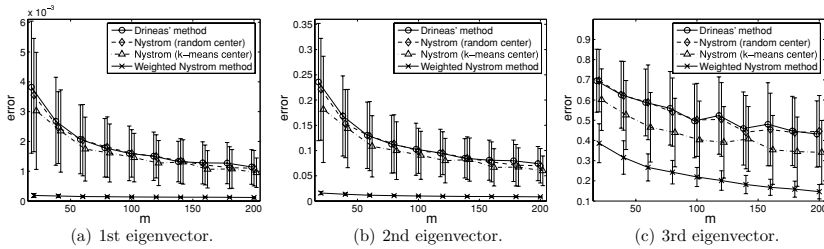


Figure 3: Approximation errors of the top three eigenvectors on the MNIST digits 0 and 1.

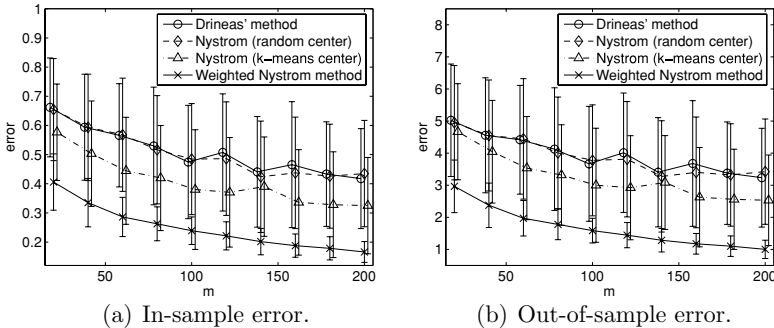


Figure 4: Embedding error (using the three leading eigenvectors) versus the number of representatives used.

the approximation performance, is still worse than the weighted Nyström method. This reflects the importance of the density-based weighting in the Nyström method.

Following Ouimet and Bengio (2005), we perform a quantitative comparison on the embedding results of the three leading eigenvectors obtained in the KPCA. We align the obtained embedding (i.e., coordinates of the data points) to the KPCA embedding through linear regression and then report the mean squared error between them. Both the in-sample error (based on the embedding of the training patterns) and out-of-sample error (based on the embedding of a test set of 2000 images) are shown in Figure 4. As can be seen, our embedding results are always superior, and the error drops rapidly as the number of landmark points increases.

We also provide a visual example by comparing the embedding results of our approach with that of standard KPCA. Figures 5a and 5b show the standard KPCA embedding results using the three leading eigenvectors, while Figures 5c and 5d show our results using only  $m = 6$  landmark points. As can be seen, by using only six representatives, our method obtains



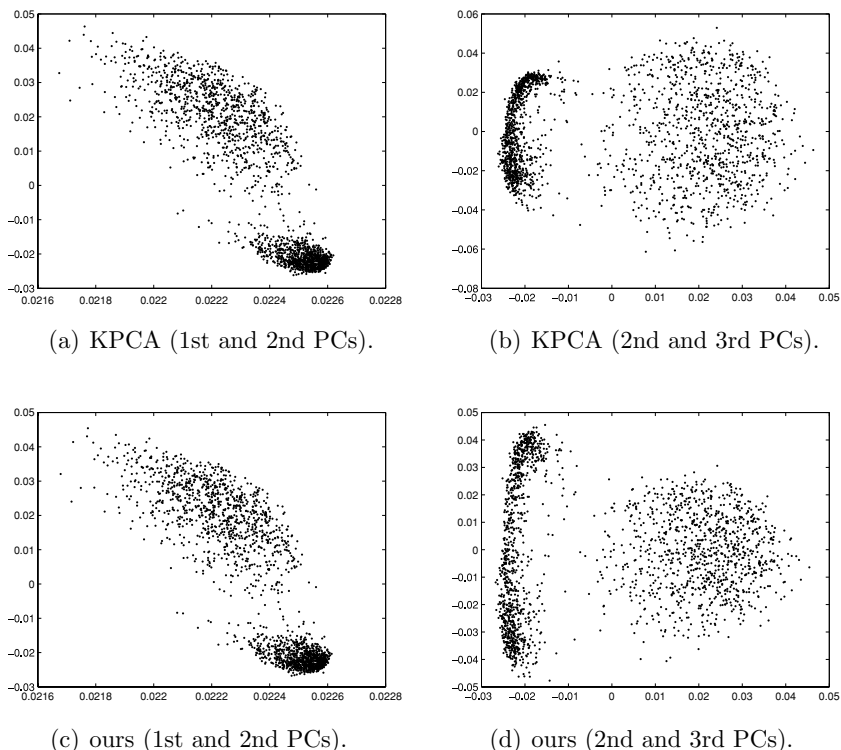


Figure 5: Embedding of the digits 0 and 1 obtained by KPCA (using the three leading eigenvectors) and our method (using only six representatives).

comparable results to those of KPCA. In other words, the eigenvectors of the  $2000 \times 2000$  kernel matrix have been well approximated by those of a  $6 \times 6$  matrix. This demonstrates the effectiveness of our approach in extracting the eigenstructures of large kernel matrices with highly compact models.

**7.2 Spectral Clustering.** In this section, we perform spectral clustering on two-digit data set, UCI and MNIST.<sup>3</sup> The UCI digits are of size  $8 \times 8$ , and each digit has about 350 training instances. The MNIST digits are of size  $28 \times 28$ , and each digit has 6000 training instances. For each data set, we perform clustering between digit 3 and each of the nine remaining digits (and thus a total of nine clustering tasks for each data set). Considering that the standard normalized cut can be quite time-consuming on the MNIST

<sup>3</sup>UCI is from <http://mllearn.ics.uci.edu/databases/>, and MNIST is from <http://yann.lecun.com/exdb/mnist/>.

Table 1: Clustering Errors (%) on the UCI Digits.

Digit	Ncut	Drineas	Nyström (Random)	Nyström(k-Means)	Ours
3-0	0.00	14.47 ± 19.43	5.31 ± 14.89	0.17 ± 0.18	0.05 ± 0.06
3-1	1.28	11.31 ± 10.06	20.18 ± 15.62	4.78 ± 7.57	1.63 ± 0.22
3-2	0.78	13.79 ± 14.05	17.53 ± 13.94	5.57 ± 10.23	1.54 ± 1.09
3-4	0.00	7.77 ± 14.28	6.78 ± 11.34	0.41 ± 0.69	0.36 ± 0.60
3-5	4.57	18.29 ± 10.42	20.62 ± 14.29	12.00 ± 8.07	5.19 ± 1.04
3-6	0.00	4.34 ± 9.49	6.48 ± 14.83	0.12 ± 0.14	0.08 ± 0.06
3-7	0.90	7.10 ± 13.36	14.18 ± 15.23	1.49 ± 0.82	1.03 ± 0.28
3-8	2.99	20.16 ± 12.93	19.25 ± 12.20	4.35 ± 4.04	2.32 ± 0.45
3-9	24.38	28.15 ± 12.26	36.52 ± 7.62	30.53 ± 3.08	25.21 ± 4.09

Table 2: Clustering Errors (%) on the MNIST Digits.

Digit	Ncut	Drineas	Nyström (Random)	Nyström(k-Means)	Ours
3-0	3.80	20.05 ± 12.13	22.17 ± 10.34	7.14 ± 4.63	3.55 ± 0.34
3-1	3.10	27.27 ± 15.06	21.93 ± 12.15	5.35 ± 3.61	3.32 ± 0.34
3-2	13.1	30.80 ± 11.55	25.20 ± 9.20	14.17 ± 4.23	12.64 ± 1.07
3-4	3.00	18.33 ± 12.13	12.00 ± 7.90	4.74 ± 2.26	2.59 ± 0.12
3-5	31.90	39.04 ± 6.41	36.53 ± 7.66	35.17 ± 4.81	32.25 ± 1.34
3-6	2.10	20.92 ± 13.37	18.73 ± 13.87	5.06 ± 4.18	1.87 ± 0.22
3-7	3.30	15.97 ± 9.64	16.38 ± 11.53	5.78 ± 2.84	3.35 ± 0.22
3-8	23.60	33.52 ± 9.46	35.13 ± 8.77	27.55 ± 3.85	25.53 ± 2.44
3-9	6.00	22.45 ± 12.91	22.24 ± 11.83	9.83 ± 3.91	6.50 ± 0.45

digits, each with 6000 samples, we choose only 500 samples from each MNIST digit. We use the gaussian kernel  $k(x, y) = \exp(-\|x - y\|^2/\sigma^2)$ , and  $\sigma$  is chosen such that the standard normalized cut gives the best performance on the nine clustering tasks. To make the problem more challenging, we choose the subset size as  $m = 5$ . All algorithms are repeated 30 times, and the averaged performance is reported in Tables 1 and 2.<sup>4</sup> As can be seen, the clustering performance of our approach is very close to that of the standard normalized cut. In comparison, other variants of the Nyström method are much inferior. A significance test (paired student-t test) shows that on all the tasks, our approach is better than the other three approaches, with a confidence level of at least 97.5%.

Next we use a specific example to demonstrate the efficiency of our approach. We choose digits 6 and 8 from the MNIST data, gradually increase the sample size of both digits, and examine the performance (clustering error and time consumption) of the proposed method (again using  $m = 5$ ).

<sup>4</sup>For binary clustering problems, the clustering error is defined as  $\frac{1}{2n} \min(\|\ell_1 + \ell_2\|_1, \|\ell_1 - \ell_2\|_1)$ , where  $\ell_1$  and  $\ell_2$  are the true label and computed label vectors of  $\{\pm 1\}$ , respectively.

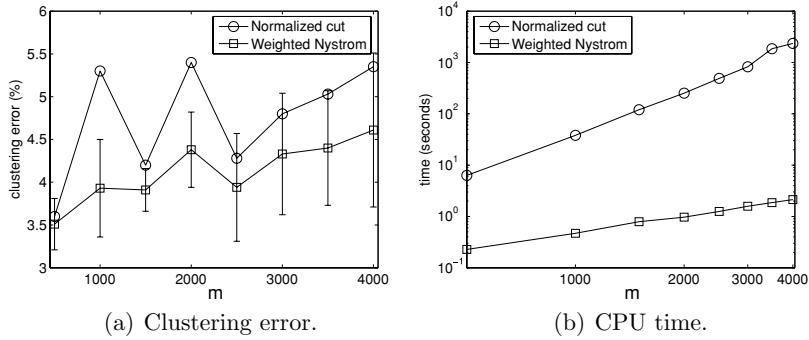


Figure 6: Comparison of clustering performance for spectral clustering and the proposed method on MNIST digits 6 and 8.

For comparison, we also report the performance of standard NCut. As can be seen from Figure 6a, when the sample size increases, the performance of our approach is very close to or even better than the standard NCut. From Figure 6b, our algorithm is faster than standard NCut by at least one to two orders of magnitude. In particular, the larger the sample size, the more obvious the improvement.

Finally, we perform image segmentation using the NCut by applying a  $k$ -means procedure in the subspace of the top three eigenvectors of the normalized kernel matrix, equation 5.1. In order to use both local coherence and global similarity information, we use the RGB colors concatenated with the  $(x, y)$  pixel positions as features. All the features are normalized to  $[0, 255]$ ; that is, we assume that the color and spatial features take equal weight. The gaussian kernel is adopted with bandwidth  $\sigma$  in the range  $[20, 40]$ . The sequential sampling procedure with  $r = 25$  is used to initialize the  $k$ -means partitioning. The implementations here are in VC7.0 and are run on a Pentium-III 2.26 GHz machine.

Segmentation results on some  $481 \times 321$  images (from the Berkeley image data set) are shown in Figure 7.<sup>5</sup> The CPU time is shown in Table 3. As can be seen, our method produces competitive segmentation results with very high speed. In particular, note that standard spectral clustering algorithms cannot be run with this large data set on our machine.

## 8 Conclusion

In this letter, we identify a primary source of inefficiency in the Nyström approximation of large kernel eigensystems, namely, that the chosen landmark

<sup>5</sup>Available online at: <http://www.cs.berkeley.edu/projects/vision/grouping>.

Table 3: CPU Time (in Seconds) and Number of Representatives  $m$  in Segmentation Tasks.

	HILL	MAN	HOUSE	FLOWER
$m$	114	175	162	86
TIME	0.45	0.66	0.91	0.51



Figure 7: Image segmentation results obtained by normalized cut with the proposed method.

points are treated with equal importance in approximating a distribution-dependent integral. We therefore extend the Nyström method to a more general, density-weighted version, called the weighted Nyström method. It uses the normalized data histogram to set the weighting coefficients, and the landmark points are carefully chosen as the cluster centers. It has the same computational complexity as the original Nyström method. Superior empirical results are obtained in its applications to kernel principal component analysis, spectral clustering, and normalized-cut-based image segmentation.

Several problems remain to be investigated in the future. For example, our current analysis and methodology can be extended to different kernels, such as linear or polynomial kernel. One interesting direction is to generalize our algorithm to more complicated, domain-specific kernels such as the string kernel. In this case, the data might be nonvectorial, and kernel evaluations will no longer take analytic forms. As a result the block quantization of kernel matrix will be more difficult. We will investigate efficient quantization schemes under specific kernel evaluation rules. Another future work is to study the matrix completion view of the density-weighted Nyström method. Such interpretation will allow us to correctly orthogonalize the approximate eigenvectors obtained with the weighted Nyström method (using more computations). It also makes possible application of the weighted Nyström method in low-rank approximation problem, which is the key to scaling up a number of popular kernel methods.

## Appendix A: Proof of Proposition 1

---

The eigensystem of  $\bar{W}$  can be written as  $n$  linear equations:

$$\sum_{j=1}^n \bar{W}_{ij} \bar{\phi}_j = \bar{\lambda} \bar{\phi}_i, \quad i = 1, 2, \dots, n.$$

The left-hand sides of the first  $|S_1|$  equations ( $i = 1, 2, \dots, |S_1|$ ) are all the same. Therefore, the corresponding right-hand sides,  $\bar{\phi}_i$ 's, are also the same. Similarly, the next  $|S_2|$   $\bar{\phi}_i$ 's (i.e.,  $i = |S_1| + 1, |S_1| + 2, \dots, |S_1| + |S_2|$ ) are also the same, and so on. Thus, it is easy to see that the eigensystem  $\bar{W}\bar{\phi} = \bar{\lambda}\bar{\phi}$  has only  $m$  independent equations,

$$\sum_{q=1}^m \tilde{W}_{pq} \tilde{\phi}_q = \tilde{\lambda} \tilde{\phi}_p, \quad p = 1, 2, \dots, m,$$

or, in matrix form,  $\tilde{W}\tilde{\phi} = \tilde{\lambda}\tilde{\phi}$ , where  $\tilde{W} \in \mathbb{R}^{m \times m}$  with elements  $\tilde{W}_{pq} = k(z_p, z_q) |S_q|$ , and  $\tilde{\lambda}, \tilde{\phi} \in \mathbb{R}^m$  are the corresponding eigenvalue and eigenvector. Hence,  $\bar{\phi}_i$  can be easily obtained by first performing eigenvalue decomposition of  $\tilde{W}$  and then extending its eigenvector  $\tilde{\phi} \in \mathbb{R}^m$  to  $\bar{\phi} \in \mathbb{R}^n$  by repeating the  $k$ th entry of  $\tilde{\phi}$   $n_k$  times ( $k = 1, 2, \dots, m$ ).

## Appendix B: Proof of Proposition 2

---

For each entry  $K_{ij}$ , the corresponding entry  $\bar{W}_{ij}$  is  $K(z_p, z_q)$ , where  $p = c(i)$  and  $q = c(j)$ . Here  $c(i)$  maps the sample index  $i$  to the corresponding cluster index  $c(i)$ . So we have

$$\begin{aligned} (K_{ij} - \bar{W}_{ij})^2 &= (K(x_i, x_j) - K(z_p, z_q))^2 \\ &= (k(\|x_i - x_j\|^2/\sigma^2) - k(\|z_p - z_q\|^2/\sigma^2))^2. \end{aligned}$$

By denoting  $d_{ij} = \|x_i - x_j\|$ ,  $D_{pq} = \|z_p - z_q\|$ , and using the mean value theorem, we have

$$\begin{aligned} (K_{ij} - \bar{W}_{ij})^2 &= (k(d_{ij}^2/\sigma^2) - k(D_{pq}^2/\sigma^2))^2 \\ &= \sigma^{-4} \xi^2 (d_{ij} + D_{pq})^2 (d_{ij} - D_{pq})^2, \end{aligned} \tag{B.1}$$

where  $\xi = \max_x |k'(x)|$ . Now we try to find bounds for  $d_{ij} + D_{pq}$  and  $|d_{ij} - D_{pq}|$ . Using the triangle inequality, we have

$$\begin{aligned} d_{ij} &= \|x_i - x_j\| \\ &\leq \|x_i - z_p\| + \|z_p - z_q\| + \|z_q - x_j\| \\ &= r_i + D_{pq} + r_j. \end{aligned} \tag{B.2}$$

Here  $r_i$  denotes the distance between sample  $x_i$  and its corresponding cluster center  $z_{c(i)}$ — $r_i = \|x_i - z_{c(i)}\| = \|x_i - z_p\|$ . By using equation B.2, we have

$$d_{ij} + D_{pq} \leq 2D_{pq} + r_i + r_j.$$

Next we derive the bound on  $|D_{pq} - d_{ij}|$ . Note that from equation B.2, we have  $d_{ij} - D_{pq} \leq r_i + r_j$ . On the other hand, through the triangular inequality,

$$\begin{aligned} D_{pq} &= \|z_p - z_q\| \\ &\leq \|z_p - x_i\| + \|x_i - x_j\| + \|x_j - z_q\| \\ &= r_i + d_{ij} + r_j, \end{aligned}$$

we have  $D_{pq} - d_{ij} \leq r_i + r_j$ . So  $|d_{ij} - D_{pq}|$  is always bounded by  $r_i + r_j$ .

With the bounds on  $|d_{ij} - D_{pq}|$  and  $d_{ij} + D_{pq}$ , equation B.1 can be further bounded by

$$\begin{aligned} (K_{ij} - \bar{W}_{ij})^2 &\leq \sigma^{-4} \xi^2 (2D_{pq} + r_i + r_j)^2 (r_i + r_j)^2 \\ &= \sigma^{-4} \xi^2 (4D_{pq}^2 (r_i + r_j)^2 + 4D_{pq} (r_i + r_j)^3 + (r_i + r_j)^4). \end{aligned}$$

Note that  $D_{pq}$  is the distance between cluster center  $z_p$  and  $z_q$ . Here we assume that  $D_{pq}$  is bounded by  $R$ , the maximum pairwise distance between sample points. Then, by summing up all the pairwise distances in the Frobenius norm, we have

$$\begin{aligned} \sum_{i,j=1}^n (K_{ij} - \bar{W}_{ij})^2 &\leq \sigma^{-4} \xi^2 \sum_{i,j=1}^n (4R^2 (r_i + r_j)^2 + 4R (r_i + r_j)^3 + (r_i + r_j)^4) \\ &\leq \sigma^{-4} \xi^2 \sum_{i,j=1}^n (4R^2 (r_i^2 + r_j^2) + 4R (r_i^3 + r_j^3) + (r_i^4 + r_j^4) \\ &\quad + 8R^2 r_i r_j + 6r_i^2 r_j^2 + 12R r_i r_j (r_i + r_j) + 4r_i r_j (r_i^2 + r_j^2)). \end{aligned}$$

Recall that  $r_i$  is the distance between sample  $x_i$  and its cluster center  $z_{c(i)}$ . By breaking the summation over index  $(i, j)$  into separate summations over  $i$  and  $j$ , we arrive at the bound in proposition 2 using the  $k$ th-norm distortion error defined there.

### Appendix C: Sequential Sampling Procedure

---

This section gives the pseudocode of sequential sampling (see algorithm 4):

Algorithm 4: Sequential Sampling

1. **Input:** data set  $X = \{x_i\}_{i=1}^n$ , threshold  $r$ .
2. **Initialization:** randomly pick one  $x_i$  as  $t_1$  and set  $C = \{t_1\}$ ,  $|C| = 1$ .
3. **for**  $i = 1$  **to**  $n$  **do**
4.    $new\_center = true$ ;
5.   **for**  $t_j \in C$  **do**
6.     **if**  $\|x_i - t_j\| \leq r$  **then**
7.       Assign  $x_i$  to  $S_j$ ;
8.        $new\_center = false$ ;
9.       **break**;
10.    **end if**
11.    **end for**
12.    **if**  $new\_center == true$  **then**
13.      $|C| = |C| + 1$ ;
14.      $t_{|C|} = x_i$ ;
15.      $C \leftarrow C \cup \{t_{|C|}\}$ ;
16.    **end if**
17. **end for**

The number of clusters obtained by sequential sampling is controlled by the threshold  $r$ . In case the user specifies the number of clusters  $m$ , a binary search procedure can be performed that requires  $O(\log m)$  repeats of sequential sampling to choose a suitable  $r$ . A similar search procedure is also applied in the greedy sampling approach (Ouimet & Bengio, 2005).

### Appendix D: Proof of Proposition 3

---

We first introduce two lemmas that will be used in the proposition.

**Lemma 1.** *A square matrix  $A$  and a nonsingular diagonal matrix  $D$  are given. For the eigensystems*

$$D^{-1}A\phi_1 = \lambda_1\phi_1,$$

$$D^{-\frac{1}{2}}AD^{-\frac{1}{2}}\phi_2 = \lambda_2\phi_2,$$

*the two eigenpairs are related as  $\lambda_1 = \lambda_2$ ,  $D^{\frac{1}{2}}\phi_1 = \phi_2$ .*

**Proof.** This can be easily verified by substitution.

**Lemma 2.** Denote the degree matrix of  $PWP$  by  $D_Z^P$ . Then,  $D_Z^P = D_Z P = P D_Z$ , where  $D_Z$  (see equation 5.6) is the degree matrix of  $\tilde{W}$ .

**Proof.** It is easy to see that  $\tilde{W} = WP$ . So  $D_Z$  is the degree matrix of  $WP$ , and the  $k$ th diagonal entry of  $D_Z$  is the sum of the  $k$ th row of  $WP$ . Similarly,  $D_Z^P$  is the degree matrix of  $PWP$ , and the  $k$ th diagonal entry of  $D_Z^P$  is the sum of the  $k$ th row of  $PWP$ . Note that the  $k$ th row of  $PWP$  is  $P_{kk}$  times that of  $WP$ . Therefore,  $[D_Z^P]_{k,k} = P_{kk}[D_Z]_{k,k}$ , or, in matrix form,  $D_Z^P = D_Z P = P D_Z$ , since both  $D_Z$  and  $P$  are diagonal.

Now we can prove proposition 3:

**Proof.** Lemma 1 shows that the eigensystem of  $D_Z^{-\frac{1}{2}}(WP)D_Z^{-\frac{1}{2}}$  can be obtained from that of  $D_Z^{-1}(WP)$ ; similarly, the eigensystem of  $(D_Z^P)^{-\frac{1}{2}}(PWP)(D_Z^P)^{-\frac{1}{2}}$  can be obtained from  $(D_Z^P)^{-1}(PWP)$ :

$$D_Z^{-1}(WP)V_1 = V_1\Lambda_1, \quad (\text{D.1})$$

$$(D_Z^P)^{-1}(PWP)V_2 = V_2\Lambda_2. \quad (\text{D.2})$$

Here  $(V_1, \Lambda_1)$  is the eigenpair of  $D_Z^{-1}(WP)$ , and  $(V_2, \Lambda_2)$  is the eigenpair of  $(D_Z^P)^{-1}(PWP)$ . According to lemma 2,  $D_Z^{-1} = (D_Z^P)^{-1}P$ . By plugging it into equation D.2, we obtain

$$\Lambda_1 = \Lambda_2, \quad \text{and} \quad V_1 = V_2. \quad (\text{D.3})$$

Namely, the two matrices have the same eigensystem. Now use lemma 1 to map the eigenvectors of  $D_Z^{-1}(WP)$  and  $(D_Z^P)^{-1}(PWP)$  back to those of  $D_Z^{-\frac{1}{2}}(WP)D_Z^{-\frac{1}{2}}$  and  $(D_Z^P)^{-\frac{1}{2}}(PWP)(D_Z^P)^{-\frac{1}{2}}$ :

$$D_Z^{\frac{1}{2}}V_1 = U_1, \quad (\text{D.4})$$

$$(D_Z^P)^{\frac{1}{2}}V_2 = U_2. \quad (\text{D.5})$$

By using equations D.3 to D.5, we have

$$U_1 = D_Z^{\frac{1}{2}}(D_Z^P)^{-\frac{1}{2}}U_2 = D_Z^{\frac{1}{2}}(P D_Z)^{-\frac{1}{2}}U_2 = P^{-\frac{1}{2}}U_2.$$

## References

---

Bach, F.R., & Jordan, M.I. (2006). Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, 7, 1963–2001.



- Baker, C. (1977). *The numerical treatment of integral equations*. Oxford: Clarendon Press.
- Belkin, M., & Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems, 14*. Cambridge, MA: MIT Press.
- Chapelle, O., Weston, J., & Schölkopf, B. (2003). Cluster kernels for semi-supervised learning. In S. Becker, S. Thrün, & K. Obermayer (Eds.), *Advances in neural information processing systems, 15* (pp. 585–592). Cambridge, MA: MIT Press.
- Cox, T., & Cox, M. (1994). *Multidimensional scaling*. London: Chapman & Hall.
- Davis, P. J., & Robinowitz, P. (1984). *Methods of numerical integration (computer science and applied mathematics)*. New York: Academic Press.
- Drineas, P., Drinea, E., & Huggins, P. (2003). An experimental evaluation of a Monte-Carlo algorithm for singular value decomposition. In *Proceedings of 8th Panhellenic Conference on Informatics* (pp. 279–296). Berlin: Springer.
- Drineas, P., & Mahoney, M. W. (2005). On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *Journal of Machine Learning Research, 6*, 2153–2175.
- Elkan, C. (2003). Using the triangular inequality to accelerate k-means. In *Proceedings of the 21th International Conference on Machine Learning* (pp. 147–153). New York: ACM Press.
- Faloutsos, C., & Lin, K.-I. (1995). FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1st ACM SIGMOD International Conference on Management of Data* (pp. 163–174). New York: ACM Press.
- Feder, T., & Greene, D. (1988). Optimal algorithms for approximate clustering. In *Proceedings of the 20th ACM Symposium on Theory of Computing* (pp. 434–444). New York: ACM Press.
- Fowlkes, C., Belongie, S., Chung, F., & Malik, J. (2004). Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*, 214–225.
- Gersho, A., & Gray, R. (1992). *Vector quantization and signal compression*. Norwell, MA: Kluwer Academic Press.
- Kanungo, T., Mount, D. M., Netanyahu, N., Piatko, C., Silverman, R., & Wu, A. Y. (2002). An efficient  $k$ -means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*, 881–892.
- Ng, A.Y., Jordan, M.I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems, 14*. Cambridge, MA: MIT Press.
- Ouimet, M., & Bengio, Y. (2005). Greedy spectral embedding. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics* (pp. 253–260). Madison, WI: Omni Press.
- Pelleg, D., & Moore, A. (1999). Accelerating exact  $k$ -means algorithms with geometric reasoning. In *Knowledge Discovery and Data Mining* (pp. 277–281). New York: ACM Press.
- Platt, J. C. (2005). Fastmap, MetricMap, and Landmark MDS are all Nyström algorithms. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics* (pp. 261–268). Madison, WI: Omni Press.

- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). *Numerical recipes in C* (2nd ed.). Cambridge: Cambridge University Press.
- Schölkopf, B., Smola, A. J., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*, 1299–1319.
- Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 888–905.
- Silva, V., & Tenenbaum, J. (2003). Global versus local methods in nonlinear dimensionality reduction. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems*, *15* (pp. 721–728). Cambridge, MA: MIT Press.
- Smola, A. J., & Kondor, R. (2003). Kernels and regularization on graphs. In *Proceedings of 16th Annual Conference on Learning Theory*. Berlin: Springer-Verlag.
- Wang, L., Wang, X., Lin, K., Shasha, D., Shapiro, B., & Zhang, K. (1999). Evaluating a class of distance-mapping algorithms for data mining and clustering. In *Proceedings of 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 307–311). New York: ACM Press.
- Williams, C., & Seeger, M. (2000). The effect of the input density distribution on kernel-based classifiers. In *Proceedings of the 17th International Conference on Machine Learning* (pp. 1159–1166). New York: ACM Press.
- Williams, C., & Seeger, M. (2001). Using the Nyström method to speed up kernel machines. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems*, *13* (pp. 682–688). Cambridge, MA: MIT Press.
- Zhang, K., & Kwok, J. (2006). Block-quantized kernel matrix for fast spectral embedding. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 1097–1104). New York: ACM Press.
- Zhang, T., & Ando, R. (2006). Analysis of spectral kernel design based semi-supervised learning. In Y. Weiss, B. Shölkopf, & J. Platt (Eds.), *Advances in neural information processing systems*, *18* (pp. 1601–1608). Cambridge, MA: MIT Press.