Block-Quantized Kernel Matrix for Fast Spectral Embedding

Kai Zhang TWINSEN@CS.UST.HK
James T. Kwok JAMESK@CS.UST.HK
Department of Computer Science, The Hong Kong University of Science and Technology, Kowloon, Hong Kong

Abstract

Eigendecomposition of kernel matrix is an indispensable procedure in many learning and vision tasks. However, the cubic complexity $O(N^3)$ is impractical for large problem, where N is the data size. In this paper, we propose an efficient approach to solve the eigendecomposition of the kernel matrix W. The idea is to approximate W with $\overline{W}$ that is composed of m^2 constant blocks. The eigenvectors of $\overline{W}$, which can be solved in $O(m^3)$ time, is then used to recover the eigenvectors of the original kernel matrix. The complexity of our method is only $O(mN + m^3)$, which scales more favorably than state-of-the-art low rank approximation and sampling based approaches ($O(m^2N + m^3)$), and the approximation quality can be controlled conveniently. Our method demonstrates encouraging scaling behaviors in experiments of image segmentation (by spectral clustering) and kernel principal component analysis.

1. Introduction

Eigendecomposition of the kernel matrix plays an important role in many machine learning and vision problems. For example, in kernel principal component analysis (KPCA) (Schölkopf et al., 1998), the eigensystem of the kernel matrix is used to extract nonlinear structures in the high-dimensional feature space. In spectral clustering (Shi & Malik, 2000; Fowlkes et al., 2004), the eigenvectors of the (normalized) Gram matrix provide an approximate clustering solution. Many manifold learning and embedding algorithms also use eigendecomposition of an affinity matrix to capture the low-dimensional structure of the input patterns.

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

However, given a set of N points, the eigendecomposition of the $N \times N$ kernel matrix scales as $O(N^3)$. This may be prohibitive for large data sets in practice. To circumvent this problem, many methods often make use of the rapidly decaying spectrum of the kernel matrix (Williams & Seeger, 2000), with prominent examples including the low-rank approximations and sampling-based methods.

As its name suggests, a low-rank approximation is an approximation of the form $L = GG'$, where $G \in \mathbb{R}^{N\times m}$ and the rank m is generally much smaller than N. A well-known example is the incomplete Cholesky decomposition (Bach & Jordan, 2002; Fine & Scheinberg, 2001) in linear algebra. Other sparse greedy kernel methods, such as (Lawrence & Herbrich, 2003; Smola & Bartlett, 2000), also compute similar approximations, and most of them scale as $O(m^2N)$.

Recently, there has been a lot of interest in sampling-based approaches. For example, the Nyström method (Baker, 1977), which selects a random subset of columns to approximate the full kernel matrix, has been used to speed up kernel machines (Williams & Seeger, 2001; Lawrence & Herbrich, 2003). More sophisticated sampling approaches have also been proposed along this line. In (Drineas & Mahoney, 2005), the columns are chosen based on a nonuniform, data-dependent probability distribution p. However, this p is constructed based on the L2 norms of all the columns in the kernel matrix, which is a relatively expensive operation. Ouimet and Bengio (2005) studies a greedy sampling scheme based on the feature space distance between a candidate example and the span of previously chosen examples. This greedy scheme outperforms the Nyström method with random sampling. However, its complexity still scales as $O(m^2N)$.

In this paper, we propose an efficient approach to compute the eigendecomposition of a large $N \times N$ kernel (affinity) matrix W. The idea is to approximate W by a matrix $\overline{W}$ that is composed of m^2 constant submatrices, where $m \ll N$. This special structure allows the eigendecomposition of $\overline{W}$ to be computed very ef-

ficiently, which then serves as an approximate solution to the eigendecomposition of the original W. The approximation quality can be controlled by the Frobenius norm $\|W - \overline{W}\|_F$. Moreover, it also allows the use of Nyström formula (Williams & Seeger, 2001) to further refine the approximate eigenvectors obtained.

The proposed approach has two desirable properties. First, its complexity only scales as $O(mN + m^3)$. This is more efficient than most existing methods, whose complexities are at least $O(m^2N + m^3)$. Second, instead of picking a subset of rows/columns from the kernel matrix, our approach computes the "representative" rows/columns based on the structure of the training data. These representatives are further weighted by using information on the data distribution. Therefore, it is more powerful than sampling-based approaches with uniform weighting. Experimentally, our approach can afford the use of very few representatives to maintain the eigen-structure of the kernel matrix. For example, in one experiment involving a $2,000 \times 2,000$ kernel matrix computed on real-world data, only 3 representatives are needed to produce a faithful embedding.

The rest of the paper is organized as follows. In Section 2, we show that matrices of the form $\overline{W}$ (i.e., composed of m^2 constant blocks, where $m \ll N$) can be eigendecomposed very efficiently in $O(m^3)$ time. This lays the foundation for the scaling behavior of the proposed approach. In Section 3, we show how the difference $\|W - \overline{W}\|_F$ between matrices W and $\overline{W}$ can be used to bound the difference of their eigen-spectra and eigenvectors. Now, the problem boils down to how to construct a blockwise-constant matrix $\overline{W}$ such that $\|W - \overline{W}\|_F$ is minimized. In Section 4, we propose two methods to optimize this objective, together with some error analysis and complexity analysis. Moreover, the Nyström extension can also be used to further refine the piecewise-constant eigenvectors of $\overline{W}$. Experimental results on using our approach for KPCA and spectral clustering are presented in Section 5, and the last section gives some concluding remarks.

2. Gram Matrix of Special Forms

In this Section, we consider a $N \times N$ matrix $\overline{W}$ with blockwise-constant structure. To be more precise, $\overline{W}$ is composed of m^2 sub-matrices, each having the constant value β_{ij}. The following shows an example:

$$\overline{W} = \begin{pmatrix} a & a & b & b & b \\ a & a & b & b & b \\ c & c & d & d & d \\ c & c & d & d & d \\ c & c & d & d & d \end{pmatrix}. \qquad (1)$$

Here, $\beta_{11} = a$, $\beta_{12} = b$, $\beta_{21} = c$, $\beta_{22} = d$. Note that the "quantization" in (1) is different from the commonly used quantization procedures (Achlioptas & McSherry, 2001). Here, our quantization has a regular "block" structure. This can also be interpreted as partitioning the data set into m local clusters $(S_1, S_2, \ldots, S_m)$, and then set the pairwise similarity between any point in S_i and any point in S_j at the constant value β_{ij}. In the following, we shall see that such a block quantization makes the eigendecomposition of $\overline{W}$ particularly easy. In comparison, other quantization schemes do not have this advantage[1].

Let n_i be the number of samples in the ith cluster $(i = 1, 2, \ldots, m)$. For example, in (1), $n_1 = 2$, and $n_2 = 3$. We now show that the eigen-system of $\overline{W}$ can be obtained efficiently. First, the eigen-system of $\overline{W}$ can be written in scalar form as N equations:

$$\sum_{j=1}^{N} \overline{W}_{ij}\overline{\phi}_j = \lambda\overline{\phi_i}, \;\; i = 1, 2, \ldots, N, \qquad (2)$$

where $\overline{\phi}$ is the $N \times 1$ eigenvector of $\overline{W}$. For the first n_1 equations $(i = 1, 2, \ldots, n_1)$, their left-hand-sides are all the same. Therefore, the corresponding right-hand-sides, $\overline{\phi}_i$s, are also the same. Similarly, the next n_2 $\overline{\phi}_i$s are also the same $(i = n_1+1, n_1+2, \ldots, n_1+n_2)$, and so on. Thus, it is easy to see that the eigen-system $\overline{W}\overline{\phi} = \lambda\overline{\phi}$ has only m independent equations

$$\sum_{j=1}^{m} n_j\beta_{ij}\widetilde{\phi}_j = \lambda\widetilde{\phi}_i, \;\; i = 1, 2, \ldots, m, \qquad (3)$$

or, in matrix form, $\widetilde{W}\widetilde{\phi} = \lambda\widetilde{\phi}$, where $\widetilde{W}$ is a $m \times m$ matrix with elements $\widetilde{W}_{ij} = \beta_{ij}n_j$, and $\widetilde{\phi}$ is the $m \times 1$ eigenvector of $\widetilde{W}$. In summary, the eigenvectors of the $N \times N$ blockwise-constant matrix $\overline{W}$ can be easily obtained by first solving the eigendecomposition of the $m \times m$ matrix $\widetilde{W}$, and then extending its eigenvector from $\widetilde{\phi}$ $(m\times1)$ to $\overline{\phi}$ $(N\times1)$ by repeating the kth entries of $\widetilde{\phi}$ a total of n_k times $(k = 1, 2, \ldots, m)$. Note that the eigenvalues of $\overline{W}$ and $\widetilde{W}$ are exactly the same.

3. Quality of the Approximation

In this Section, we give some background on the approximation of matrices and their associated eigen-systems.

[1] In principle, the columns and rows of W may have to be permuted before it can be quantized as "blocks". However, this operation can be avoided in practice by re-arranging the "sample points" instead.

3.1. Bounding the Difference in Eigenvalues

From the perturbation theory of matrices, it is known that the size of the difference between two matrices can be used to bound the difference between their singular value spectra (Bhatia, 1997). In particular, given two matrices $A, E \in \mathbb{R}^{m\times n}$, let $\sigma_k(A)$ denote the kth singular value of A, then

$$\begin{aligned} \max_{1\le t\le n} |\sigma_t(A+E)-\sigma_t(A)| &\le \|E\|_2, \\ \sum_{k=1}(\sigma_k(A+E)-\sigma_k(A))^2 &\le \|E\|_F^2. \end{aligned} \quad (4)$$

(4) is also known as the Hoffman-Wielandt inequality.

3.2. Bounding the Difference in Eigenvectors

Besides the eigenvalues, the eigenvectors are also of great importance in tasks such as KPCA and spectral clustering. In this Section, we derive an error bound on the eigenvectors using the Frobenius norm of the difference matrix $\|E\|_F$.

Denote the original Gram matrix and its blockwise-constant approximation by W and $\overline{W}$, respectively. Let their eigen-systems be $W\mu = \alpha\mu$ and $\overline{W}\nu = \beta\nu$, where μ (or ν) is an (normalized) eigenvector of W (or $\overline{W}$). In the following, we use $\|\mu-\nu\|$ to measure how well the eigenvectors of $\overline{W}$ approximate those of W. Let $E = W - \overline{W}$, then

$$\begin{aligned} \|\mu-\nu\| &= \left\|\frac{1}{\alpha}W\mu - \frac{1}{\beta}\overline{W}\nu\right\| \\ &= \left\|\frac{1}{\alpha}W\mu - \frac{1}{\beta}(W-E)\nu\right\| \\ &= \left\|W\left(\frac{1}{\alpha}\mu - \frac{1}{\beta}\nu\right) + \frac{1}{\beta}E\nu\right\|. \end{aligned} \quad (5)$$

Note that $\frac{1}{\alpha}\mu - \frac{1}{\beta}\nu = \left(\frac{1}{\alpha}-\frac{1}{\beta}\right)\mu + \frac{1}{\beta}(\mu-\nu)$. Hence, (5) can be written as

$$\begin{aligned} \|\mu-\nu\| &= \left\|W\left(\frac{1}{\alpha}-\frac{1}{\beta}\right)\mu + \frac{1}{\beta}W(\mu-\nu) + \frac{1}{\beta}E\nu\right\| \\ &\le \frac{1}{|\beta|}\|W(\mu-\nu)\| + \left(\left|\frac{1}{\alpha}-\frac{1}{\beta}\right|\right)\|W\mu\| \\ &\quad + \frac{1}{|\beta|}\|E\nu\|. \end{aligned} \quad (6)$$

As $\|\mu\| = \|\nu\| = 1$, therefore $\|W\mu\| \le \|W\|_2$ and $\|E\nu\| \le \|E\|_2$. Moreover, $\|W(\mu-\nu)\| \le \|W\|_2 \cdot \|\mu-\nu\| \le \|W\|_2(\|\mu\|+\|\nu\|) = 2\|W\|_2$. Plugging these back into (6), we obtain

$$\|\mu-\nu\| \le \frac{2}{|\beta|}\|W\|_2 + \left(\left|\frac{1}{\alpha}-\frac{1}{\beta}\right|\right)\|W\|_2 + \frac{1}{|\beta|}\|E\|_2.$$

For positive eigenvalues α and β, we have

$$\|\mu-\nu\| \le \begin{cases} \left(\frac{1}{\alpha}+\frac{1}{\beta}\right)\|W\|_2 + \frac{1}{\beta}\|E\|_2, & \alpha \le \beta, \\ \left(\frac{3}{\beta}-\frac{1}{\alpha}\right)\|W\|_2 + \frac{1}{\beta}\|E\|_2, & \alpha > \beta. \end{cases} \quad (7)$$

Note that α and $\|W\|_2$ are fixed constants, and $\|E\|_2$ is bounded by $\|E\|_F$. Therefore the error $\|\mu-\nu\|$ is controlled by the eigenvalue β and the Frobenius norm of the difference matrix E. This suggests: 1) The larger the β, the smaller is the approximation error. In other words, the leading eigenvectors of W have a better approximation than its trailing eigenvectors. 2) By minimizing $\|E\|_F$, the quality of the approximate eigenvectors can be improved.

4. The Proposed Method

The basic idea of our approach is described as follows. In order to compute the eigendecomposition of the kernel matrix W, we first find a blockwise-constant matrix $\overline{W}$ to approximate W. The approximation criteria is $\|W-\overline{W}\|_F$, which (as shown in Section 3) can be used to bound the difference between the eigensystems of the two matrices. At the same time, the blockwise-constant structure of $\overline{W}$ makes its eigendecomposition particularly easy. Therefore, we can also obtain the eigensystem of W easily.

Let f be the objective $\|W-\overline{W}\|_F$, then

$$f = \sum_{i,j=1}^{N}\left(W_{ij}-\overline{W}_{ij}\right)^2 = \sum_{i,j=1}^{m}\sum_{x_p\in S_i, x_q\in S_j}\left(W_{pq}-\beta_{ij}\right)^2.$$

It can be minimized directly by setting $\frac{\partial f}{\partial \beta_{ij}} = 0$ to obtain

$$\beta_{ij} = \frac{1}{n_i n_j}\sum_{x_p\in S_i, x_q\in S_j} W_{pq} = \frac{1}{n_i n_j}\sum_{p,q} K(x_p, x_q). \quad (8)$$

However, this takes $O(N^2)$ time for computing β_{ij}'s.

In the following, we employ a different strategy. As discussed in Section 2, the blockwise-constant structure of $\overline{W}$ implies that the data set is divided into clusters. The constant β_{ij} can be equivalently expressed as $\beta_{ij} = K(t_i, t_j)$, where t_i's are "representatives" of the local clusters. We will first analyze how the data partitioning step, i.e., configuration of the local clusters, affects the approximation error $\|W-\overline{W}\|_F$ (Section 4.1). Then, we propose two methods to obtain the cluster representatives (Sections 4.2 and 4.3).

4.1. Effect of Partitioning on the Approximation Error

To simplify notations, let $f = \sum_{i=1}^{m}\sum_{j=1}^{m} f_{ij}$, where f_{ij} denotes the component of f associated with the

local clusters S_i and S_j. Suppose the use of stationary kernels of the form $K(x,y) = k\left(\frac{\|x-y\|^2}{\sigma^2}\right)$. Denote $d_{pq} = \|x_p - x_q\|$, and $D_{ij} = \|t_i - t_j\|$. Then, by the mean value theorem,

$$|K(x_p,x_q)-K(t_i,t_j)| = \left|k(\frac{d_{pq}^2}{\sigma^2})-k(\frac{D_{ij}^2}{\sigma^2})\right| \leq \xi\frac{\left|d_{pq}^2 - D_{ij}^2\right|}{\sigma^2},$$

where $\xi = \max_x |k'(x)|$. Without loss of generality, suppose that the local cluster S_i is enclosed by a minimum enclosing ball of radius r_i. We are interested in the case where the representatives t_i's also fall inside the enclosing ball of the local clusters S_i's. Then, it is easy to see that $|d_{pq} + D_{ij}| \leq 2(D_{ij} + r_i + r_j)$ and $|d_{pq} - D_{ij}| \leq 2(r_i + r_j)$. Consequently,

$$\begin{aligned} f_{ij} &= \sum_{p\in S_i, q\in S_j} \left(K(x_p,x_q) - K(t_i,t_j)\right)^2 \\ &\leq \xi^2 \sum_{p,q} \frac{16(D_{ij}+r_i+r_j)^2(r_i+r_j)^2}{\sigma^4} \\ &\leq \xi^2 n_i n_j \frac{16(2R)^2(D_{ij}+2R)^2}{\sigma^4}, \end{aligned}$$

where $R = \max_i r_i$. The overall objective function can be bounded by

$$\begin{aligned} \sum_{i,j=1}^{m} f_{ij} &\leq \xi^2 \sum_{i=1}^{m}\sum_{j=1}^{m} n_i n_j \frac{64R^2(D_{ij}+2R)^2}{\sigma^4} \\ &= 64N^2\xi^2\frac{R^2}{\sigma^4}\left(\overline{D^2} + 4R^2 + 4\overline{D}R\right), \quad (9) \end{aligned}$$

where $\overline{D} = \frac{1}{N^2}\sum_{ij} n_i n_j D_{ij}$ and $\overline{D^2} = \frac{1}{N^2}\sum_{ij} n_i n_j D_{ij}^2$ can be regarded as proxies for the average pairwise distance and average squared pairwise distance of the training data. Note that both $\overline{D}$ and $\overline{D^2}$ are fixed given the data. Therefore, to obtain a low error, we should partition the data set into compact local clusters such that every point is close to its cluster center and the value of R is small. In the extreme case where every point is a cluster, all the enclosing balls have zero radius, and the error f is zero. Note that the larger the kernel parameter σ, the lower is the error.

4.2. Sequential Sampling

As discussed in Section 4.1, we are interested in efficient algorithms that partition the data into local compact clusters. Popular methods include vector quantization (VQ) and spatial data structures like the kd-tree (Moore, 1998). However, VQ is sensitive to the initial choice of code vectors, while the efficiency of spatial data structures degrades rapidly as the input dimension increases. Here, we propose a simple but very efficient procedure, called *sequential sampling*, to partition the data and obtain the cluster representatives t_i's:

1: Randomly select a sample as t_1 and initialize the cluster center set $C = \{t_1\}$. Then, for $i = 1, 2, \ldots, N$, do the following.
2: Calculate the distance between x_i and each t_j in C. Once we have a t_j such that $\|x_i - t_j\| \leq r$, assign x_i to S_j, let $i = i + 1$, and go to the next iteration.
3: If $\|x_i - t_j\| > r, \forall t_j \in C$, add x_i to C as a new cluster center and assign x_i to this new cluster. Let $i = i + 1$ and go to the next iteration.
4: On termination, count the number of samples, n_j, in S_j, and update each $t_j \in C$ as $t_j = \frac{1}{n_j}\sum_{x_i\in S_j} x_i$.

The complexity of sequential sampling is only $O(mN)$, as each data point is related to at most m centers, and the algorithm requires only one pass of the data. By using a hierarchical scheme (Feder & Greene, 1988), this can be further reduced to $N \log m$. At the same time, the volume of the resultant local clusters will never exceed the hypercube of side length $2r$, where r is the partitioning threshold. Combining this with the error bound in (9), we can have a guidance on the choice of r for a certain level of accuracy.

4.3. Gradient Optimization

From optimization point of view, note that the approximation error $f = \|W - \overline{W}\|_F$ can also be written as a function of the cluster representatives t_i's,

$$\begin{aligned} f &= \sum_{i,j=1}^{m} \sum_{p\in S_i, q\in S_j} \left(K(x_p,x_q) - K(t_i,t_j)\right)^2 \\ &= \sum_{i,j=1}^{m} \left(\sum_{p,q} K^2(x_p,x_q) + \sum_{p,q} K^2(t_i,t_j) \right. \\ &\quad \left. -2\sum_{p,q} K(x_p,x_q)K(t_i,t_j)\right) \quad (10) \\ &= \sum_{i,j=1}^{m} \left(\Omega_{ij} + A_{ij}K^2(t_i,t_j) - 2B_{ij}K(t_i,t_j)\right), \end{aligned}$$

where $\Omega_{ij} = \sum_{p,q} K^2(x_p,x_q)$ is constant with regard to t_i's, $A_{ij} = |S_i| \cdot |S_j|$, and $B_{ij} = \sum_{p\in S_i, q\in S_j} K(x_p,x_q)$. Therefore, standard optimization techniques, such as gradient descent, can be used to find the local optima of f. Suppose the use of the Gaussian kernel $K(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)$, then by setting the derivative of f with regard to t_k $(k = 1, 2, ..., m)$

to zero, we have the iteration formula

$$t_k = \frac{\sum_{j\neq k} t_j \left(B_{kj} e^{-\frac{(t_k-t_j)^2}{2\sigma^2}} - A_{kj} e^{-\frac{(t_k-t_j)^2}{\sigma^2}} \right)}{\sum_{j\neq k} B_{kj} e^{-\frac{(t_k-t_j)^2}{2\sigma^2}} - A_{kj} e^{-\frac{(t_k-t_j)^2}{\sigma^2}}}. \quad (11)$$

The gradient optimization procedure can be used to further improve the cluster representatives as follows:

1: Initialize the cluster centers t_is by using the sequential sampling procedure in Section 4.2.
2: Apply the update equation in (11) to the cluster centers/representatives.
3: Re-partition the data set by allocating every point to its closest center. Then compute the mean of each cluster as their representatives.
4: Go to step 2 until convergence.

There are several practical issues here. First, a direct computation of the B_{ik}'s is expensive. This can be avoided by making use of the partition in sequential sampling (details are in the Appendix). Second, care should be taken during the iterations, and the update procedure should be terminated before the denominator of (11) is close to zero. In our experiments, the number of iterations required is very small. Third, when the number of partitions m is reasonably large (say, larger than $0.05N$ in our experiments), the improvement brought by the gradient method is limited. In other words, the gradient method is particularly useful when the data partitioning is coarse.

4.4. Refining the Eigenvectors

Recall that the eigenvectors of $\overline{W}$ are piecewise-constant, i.e., there are only m different values in the $N \times 1$ eigenvector $\overline{\phi}$. These m values are associated with the m cluster representatives t_i's. For all the x_i's in the same cluster, their $\overline{\phi}_i$'s are also the same. This means we can use the Nyström extension to further refine the piecewise-constant eigenvectors $\overline{\phi_k}$'s as:

$$\overline{\phi_k}(x) = \frac{1}{N\lambda_k} \sum_{i=1}^{m} n_i \overline{\phi}_{ik} K(x, t_i), \quad (12)$$

where the local cluster sizes (n_i's) are very important weighting factors that carry distribution information.

The refinement procedure in (12) can further improve the accuracy of the obtained eigenvectors. However, empirical experience on clustering and image segmentation tasks shows that the piecewise-constant eigenvectors obtained are already good enough. Thus, the $O(mN)$ operation in (12) can be avoided, making our algorithm much more efficient than the Nyström method. On the other hand, for the KPCA experiment in Section 5, since we are directly measuring the quantitative error of the obtained eigenvectors, we will first use the refinement procedure (12) and then conduct our evaluation.

4.5. Complexity Analysis

Partitioning the data into a number of m local clusters takes $O(\log mN)$ time for the sequential sampling procedure. If gradient optimization is used to further refine the cluster representatives, the complexity is $O(m^2)$ by using the approximate procedure in the Appendix. With the cluster representatives, entries of $\widetilde{W}$ can be computed in $O(m^2)$ time, and decomposing the $m \times m$ matrix $\widetilde{W}$ takes $O(m^3)$ time. Finally, using the Nyström method to reconstruct the complete eigenvectors (and thus the embedding) of the training data takes $O(mN)$ time.

Data-dependent kernels often involve additive normalization (e.g., for KPCA), $K_A(x,y) = K(x,y) - E_u[K(u,y)] - E_v[K(x,v)] + E_{uv}[K(u,v)]$, or divisive normalization (e.g., for spectral clustering) $K_D(x,y) = K(x,y)(E_u[K(u,y)]E_v[K(x,v)])^{-1/2}$. It is easy to see that the blockwise-constant structure of $\overline{W}$ allows us to (approximately) compute both of them in $O(m^2)$ time. So the overall complexity is $O(m^2 + \log mN + m^3) = O(mN + m^3)$. Note that for most existing methods, selection of the representatives already takes $O(m^2N)$ time. Moreover, the memory requirement of our approach is only $O(mN)$ because we do not need to store the full Gram matrix.

5. Experiments

In this Section, we study the efficiency of the proposed method in solving large eigen-systems by performing experiments on both kernel principal component analysis and spectral clustering. Implementations are in VC7.0, and all experiments are run on a 2.26GHz Pentium–3 PC.

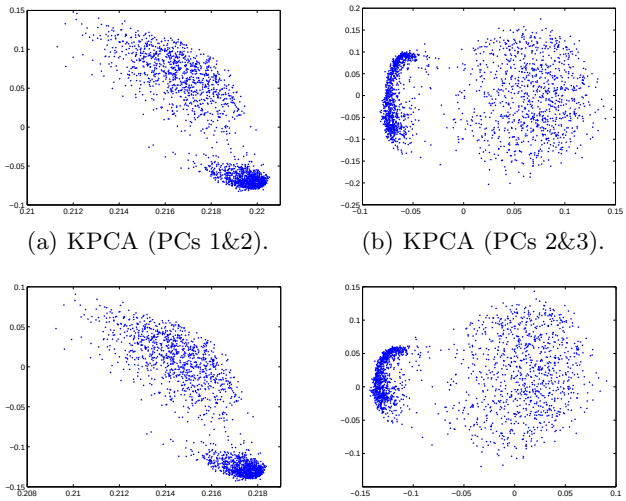
5.1. Kernel Principal Component Analysis

In this Section, we perform KPCA experiments on the digits 0 and 1 from the MNIST data set[2]. Each image is of size 28×28. We use 2,000 images for training and another 2,000 for testing. The Gaussian kernel is used. For our method, we use sequential sampling with threshold $r = \sqrt{140}$, which divides the data set into $m = 3$ clusters. Gradient optimization is then used to further improve the cluster centers.

Figures 1(a) and 1(b) show the KPCA embedding

[2] http://yann.lecun.com/exdb/mnist/

results using the 3 leading eigenvectors, while Figures 1(c) and 1(d) show our results using the refined eigenvectors in (12). As can be seen, by using only three representatives, our method obtains comparable result as that of KPCA. In other words, the eigenvectors of the $2,000 \times 2,000$ kernel matrix has been well-approximated by those of a 3×3 matrix. This demonstrates the effectiveness of our approach in extracting the eigen-structures of large kernel matrices with highly compact models.



(a) KPCA (PCs 1&2). (b) KPCA (PCs 2&3).

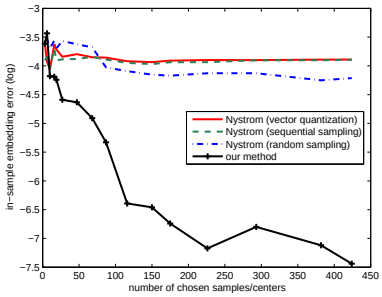
(c) ours (PCs 1&2). (d) ours (PCs 2&3).

Figure 1. Embedding of the digits 0 and 1 obtained by KPCA (using the three leading eigenvectors) and our method (using only 3 representatives).

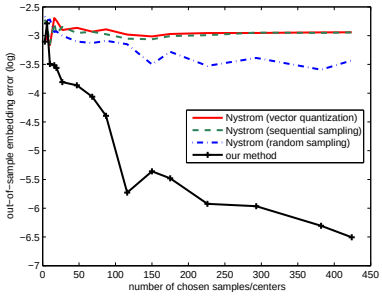
Following (Ouimet & Bengio, 2005), we perform a quantitative comparison of the embedding results on the 3 leading eigenvectors. We align the obtained embedding (i.e., coordinates of the data points) to the KPCA embedding through linear regression, and then report the mean squared error between them. Our method (without using gradient optimization) is compared to the Nyström algorithm with different ways of selecting the samples: 1) random subset[3]; 2) using sequential sampling; 3) using VQ. Both the in-sample error (embedding of the training patterns) and out-of-sample error (embedding of the test patterns) are shown.

Figure 2 shows that our embedding results are always superior, and the error drops rapidly as the number of representatives used increases. Another interesting observation is that the direct use of sequential sampling or VQ in the Nyström algorithm does not help, as the sampling procedure misrepresents the data distribution. On the other hand, the weighting scheme in (3) and (12) plays a central role in representing the input density and rectifies this problem when sampling procedures are used with our method.

[3]Experiments with different random subsets are repeated ten times, and the average performance reported.



(a) In-sample error.



(b) Out-of-sample error.

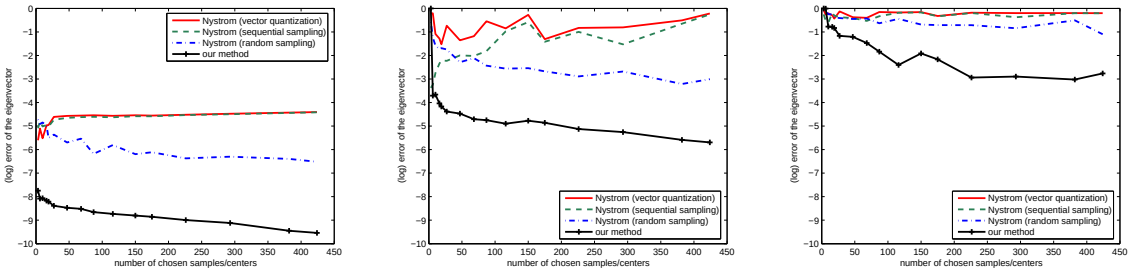
Figure 2. Embedding error (using the 3 leading eigenvectors) versus the number of representatives used.

Figure 3 compares the approximation errors of the eigenvectors obtained ($\|\phi - \overline{\phi}\|$). Again, our method is more accurate than the others. Moreover, as discussed in Section 3.2, the leading eigenvectors have a better approximation than the trailing ones.

Table 1 shows the total time taken by our method on using different partitioning thresholds r in the sequential sampling procedure. The number of representatives/clusters (m) obtained is also shown. Empirically, good performance can usually be obtained when m is around 50. While standard KPCA takes 87.45 seconds, our method can be hundreds of times faster. Moreover, in general, the larger the training set, the higher is the speedup.

5.2. Spectral Clustering

In this experiment, we perform image segmentation using the normalized cuts (Shi & Malik, 2000), which is one of the most widely used spectral clustering techniques. 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 $[0, 255]$, i.e., we assume that the color and spatial features take equal weight.



(a) 1st eigenvector. (b) 2nd eigenvector. (c) 3rd eigenvector.

Figure 3. Approximation errors of the three leading eigenvectors.

The Gaussian kernel is adopted, with bandwidth σ varying in $[20, 40]$, and the partitioning threshold is chosen as $r = 25$.

Segmentation results on some 481×321 images[4] and a large 1024×768 image[5] are shown in Figures 4 and 5. The time consumption is shown in Table 2. As can be seen, our method produces competitive segmentation results with very high speed. In particular, the segmentation of the 1024×768 image takes only about 4 seconds, while standard spectral clustering algorithms cannot be run with this large data set on our machine.

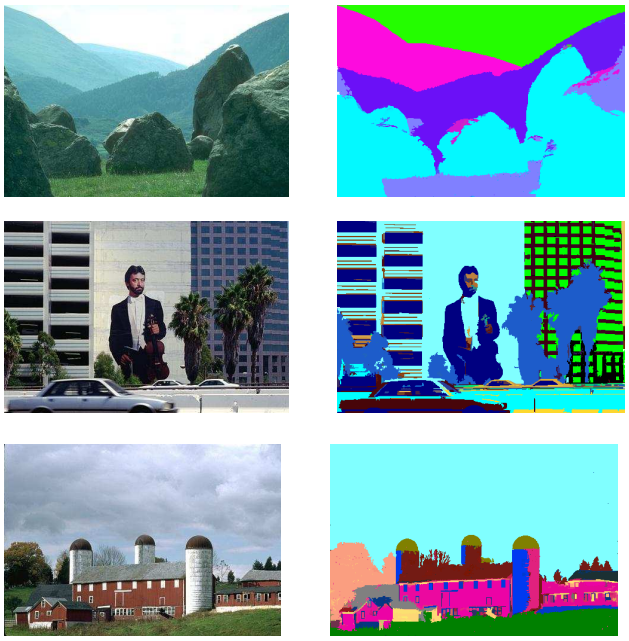


Figure 4. Image segmentation results obtained by spectral clustering with the proposed method.

[4]These images are taken from the Berkeley image data set (http://www.cs.berkeley.edu/projects/vision/grouping).

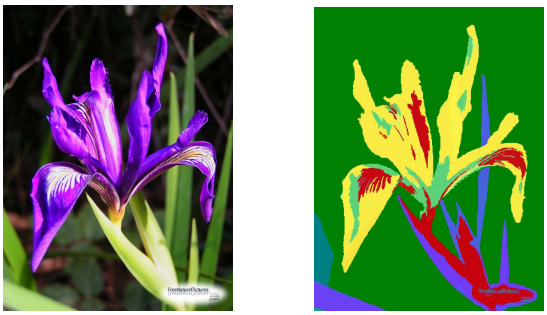
[5]http://www.freenaturepictures.com

Table 2. Total time (in seconds) and number of representatives (m) in segmentation tasks.

	HILL	MAN	HOUSE	FLOWER
m	114	175	162	240
TIME	0.45	0.66	0.91	4.27

6. Conclusion

In this paper, we propose an efficient approach for eigendecomposition of kernel matrices. It scales more favorably than most existing low-rank approximations and sampling-based approaches. Moreover, the method greatly improves the convergence behavior of the Nyström method by introducing density-based coefficients as weighting factors for the data representatives. While it is usually difficult to make use of the distributional information in the Nyström method, our method provides an effective solution to this problem. Experiments demonstrate the success of our method in extracting the eigen-structures of large kernel matrices using very few representatives.



(a) Original image. (b) Result.

Figure 5. Segmentation result on an $1,024 \times 768$ image.

Our approach can be easily extended to the multiscale framework (Cour et al., 2005). The sequential sampling procedure (which is implemented hierarchically) can naturally follow the multiscale methodologies when working in the concatenated spatial-range domain. Moreover, it will be interesting to compare

Table 1. Total time (in seconds) and number of representatives m obtained at different partitioning thresholds (r) used in the sequential sampling procedure.

r^2	140	120	110	100	90	80	70	65	60	56	52	48	44	40	36
m	3	6	10	16	19	27	47	68	87	116	150	175	226	293	382
TIME	0.04	0.06	0.09	0.14	0.17	0.26	0.48	0.72	0.95	1.32	1.82	2.31	3.57	5.35	8.55

the matrix approximation obtained with other low-rank approximation techniques.

Acknowledgments

This research has been partially supported by the Research Grants Council of the Hong Kong Special Administrative Region under grant 615005.

Appendix: Computation of B_{ij}

Note that the data set has been divided into clusters S_i's by sequential sampling (with threshold r). For each cluster, we further apply the sequential sampling procedure with a smaller threshold $r' < r$. Suppose that each cluster S_i is divided into k_i smaller groups, $Q_{i1}, Q_{i2}, \ldots, Q_{ik_i}$, with centers $o_{i1}, o_{i2}, \ldots, o_{ik_i}$, and group size $l_{i1}, l_{i2}, \ldots, l_{ik_i}$. Then B_{ij} can be approximated by $B_{ij} = \sum_{o_{ia} \in S_i} \sum_{o_{jb} \in S_j} l_{ia} l_{jb} K(o_{ia}, o_{jb})$. The complexity for computing B_{ij}'s becomes $O((m')^2)$, where $m' = \sum_{i=1}^{m} k_i$. Note that this partition is just a deeper hierarchy of the original partition obtained. So, $m < m' \ll N$, and m' is roughly a constant times larger than m. Computing B_{ij}'s then also takes $O(m^2)$ time.

References

Achlioptas, D., & McSherry, F. (2001). Fast computation of low rank matrix approximations. *Proceedings of the 23th Annual ACM Symposium on Theory of Computing* (pp. 611 – 618).

Bach, F., & Jordan, M. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, *3*, 1–48.

Baker, C. (1977). *The numerical treatment of integral equations*. Oxford: Clarendon Press.

Bhatia, R. (1997). *Matrix analysis*. New York: Springer-Verlag.

Cour, T., Bénézit, F., & Shi, J. (2005). Spectral segmentation with multiscale graph decomposition. *International Conference on Computer Vision and Pattern Recognition* (pp. 1124–1131).

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.

Feder, T., & Greene, D. (1988). Optimal algorithms for approximate clustering. *Proceedings of the 20th ACM Symposium on Theory of Computing* (pp. 434–444).

Fine, S., & Scheinberg, K. (2001). Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, *2*, 243–264.

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.

Lawrence, N. Seeger, M., & Herbrich, R. (2003). Fast sparse Gaussian process methods: The informative vector machine. *Advances in Neural Information Processing Systems.* (pp. 625–632). MIT Press.

Moore, A. (1998). Very fast EM-based mixture model clustering using multiresolution kd-trees. *Advances in Neural Information Processing Systems* (pp. 543–549). San Mateo, CA: Morgan Kaufmann.

Ouimet, M., & Bengio, Y. (2005). Greedy spectral embedding. *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics* (pp. 253–260).

Schölkopf, B., Smola, A., & 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.

Smola, A. J., & Bartlett, P. L. (2000). Sparse greedy Gaussian process regression. *Advances in Neural Information Processing System* (pp. 619–625).

Williams, C., & Seeger, M. (2000). The effect of the input density distribution on kernel-based classifiers. *Proceedings of the 17th International Conference on Machine Learning* (pp. 1159–1166).

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