Journal of Machine Learning Research 13 (2012) 981-1006 Submitted 3/11; Revised 12/11; Published 4/12

# Sampling Methods for the Nyström Method

**Sanjiv Kumar** SANJIVK@GOOGLE.COM
*Google Research*
*76 Ninth Avenue*
*New York, NY 10011*

**Mehryar Mohri** MOHRI@CS.NYU.EDU
*Courant Institute and Google Research*
*251 Mercer Street*
*New York, NY 10012*

**Ameet Talwalkar** AMEET@CS.BERKELEY.EDU
*University of California, Berkeley*
*Division of Computer Science*
*465 Soda Hall*
*Berkeley, CA 94720*

**Editor:** Inderjit Dhillon

## Abstract

The Nyström method is an efficient technique to generate low-rank matrix approximations and is used in several large-scale learning applications. A key aspect of this method is the procedure according to which columns are sampled from the original matrix. In this work, we explore the efficacy of a variety of *fixed* and *adaptive* sampling schemes. We also propose a family of *ensemble*-based sampling algorithms for the Nyström method. We report results of extensive experiments that provide a detailed comparison of various fixed and adaptive sampling techniques, and demonstrate the performance improvement associated with the ensemble Nyström method when used in conjunction with either fixed or adaptive sampling schemes. Corroborating these empirical findings, we present a theoretical analysis of the Nyström method, providing novel error bounds guaranteeing a better convergence rate of the ensemble Nyström method in comparison to the standard Nyström method.

**Keywords:** low-rank approximation, nyström method, ensemble methods, large-scale learning

## 1. Introduction

A common problem in many areas of large-scale machine learning involves deriving a useful and efficient approximation of a large matrix. This matrix may be a kernel matrix used with support vector machines (Cortes and Vapnik, 1995; Boser et al., 1992), kernel principal component analysis (Schölkopf et al., 1998) or manifold learning (Platt, 2004; Talwalkar et al., 2008). Large matrices also naturally arise in other applications, for example, clustering, collaborative filtering, matrix completion, robust PCA, etc. For these large-scale problems, the number of matrix entries can be in the order of tens of thousands to millions, leading to difficulty in operating on, or even storing the matrix. An attractive solution to this problem involves using the Nyström method to generate a low-rank approximation of the original matrix from a subset of its columns (Williams and Seeger, 2000). A key aspect of the Nyström method is the procedure according to which the columns are

sampled. This paper presents an analysis of different sampling techniques for the Nyström method both empirically and theoretically.[1]

In the first part of this work, we focus on various *fixed* sampling methods. The Nyström method was first introduced to the machine learning community (Williams and Seeger, 2000) using uniform sampling without replacement, and this remains the sampling method most commonly used in practice (Talwalkar et al., 2008; Fowlkes et al., 2004; de Silva and Tenenbaum, 2003; Platt, 2004). More recently, the Nyström method has been theoretically analyzed assuming sampling from fixed, non-uniform distributions over the columns (Drineas and Mahoney, 2005; Belabbas and Wolfe, 2009; Mahoney and Drineas, 2009). In this work, we present novel experiments with several real-world data sets comparing the performance of the Nyström method when used with uniform versus non-uniform sampling distributions. Although previous studies have compared uniform and non-uniform distributions in a more restrictive setting (Drineas et al., 2001; Zhang et al., 2008), our results are the first to compare uniform sampling with the sampling technique for which the Nyström method has theoretical guarantees. Our results suggest that uniform sampling, in addition to being more efficient both in time and space, produces more effective approximations. We further show the benefits of sampling without replacement. These empirical findings help motivate subsequent theoretical analyses.

The Nyström method has also been studied empirically and theoretically assuming more sophisticated iterative selection techniques (Smola and Schölkopf, 2000; Fine and Scheinberg, 2002; Bach and Jordan, 2002). In the second part of this work, we provide a survey of adaptive techniques that have been suggested for use with the Nyström method, and present an empirical comparison across these algorithms. As part of this work, we build upon ideas of Deshpande et al. (2006), in which an adaptive, error-driven sampling technique with relative error bounds was introduced for the related problem of matrix projection (see Kumar et al. 2009b for details). However, this technique requires the full matrix to be available at each step, and is impractical for large matrices. Hence, we propose a simple and efficient algorithm that extends the ideas of Deshpande et al. (2006) for adaptive sampling and uses only a small submatrix at each step. Our empirical results suggest a trade-off between time and space requirements, as adaptive techniques spend more time to find a concise subset of informative columns but provide improved approximation accuracy.

Next, we show that a new family of algorithms based on mixtures of Nyström approximations, *ensemble Nyström algorithms*, yields more accurate low-rank approximations than the standard Nyström method. Moreover, these ensemble algorithms naturally fit within distributed computing environments, where their computational costs are roughly the same as that of the standard Nyström method. This issue is of great practical significance given the prevalence of distributed computing frameworks to handle large-scale learning problems. We describe several variants of these algorithms, including one based on simple averaging of $p$ Nyström solutions, an exponential weighting method, and a regression method which consists of estimating the mixture parameters of the ensemble using a few columns sampled from the matrix. We also report the results of extensive experiments with these algorithms on several data sets comparing different variants of the ensemble Nyström algorithms and demonstrating the performance improvements gained over the standard Nyström method.

1. Portions of this work have previously appeared in the Conference on Artificial Intelligence and Statistics (Kumar et al., 2009a), the International Conference on Machine Learning (Kumar et al., 2009b) and Advances in Neural Information Processing Systems (Kumar et al., 2009c).

Finally, we present a theoretical analysis of the Nyström method, namely bounds on the reconstruction error for both the Frobenius norm and the spectral norm. We first present a novel bound for the Nyström method as it is often used in practice, that is, using uniform sampling without replacement. We next extend this bound to the ensemble Nyström algorithms, and show these novel generalization bounds guarantee a better convergence rate for these algorithms in comparison to the standard Nyström method.

The remainder of the paper is organized as follows. Section 2 introduces basic definitions, provides a short survey on related work and gives a brief presentation of the Nyström method. In Section 3, we study various fixed sampling schemes used with the Nyström method. In Section 4, we provide a survey of various adaptive techniques used for sampling-based low-rank approximation and introduce a novel adaptive sampling algorithm. Section 5 describes a family of ensemble Nyström algorithms and presents extensive experimental results. We present novel theoretical analysis in Section 6.

## 2. Preliminaries

Let $\mathbf{T} \in \mathbb{R}^{a\times b}$ be an arbitrary matrix. We define $\mathbf{T}^{(j)}$, $j = 1\dots b$, as the $j$th column vector of $\mathbf{T}$, $\mathbf{T}_{(i)}$, $i = 1\dots a$, as the $i$th row vector of $\mathbf{T}$ and $\|\cdot\|$ the $l_2$ norm of a vector. Furthermore, $\mathbf{T}^{(i:j)}$ refers to the $i$th through $j$th columns of $\mathbf{T}$ and $\mathbf{T}_{(i:j)}$ refers to the $i$th through $j$th rows of $\mathbf{T}$. If $\mathrm{rank}(\mathbf{T}) = r$, we can write the thin Singular Value Decomposition (SVD) of this matrix as $\mathbf{T} = \mathbf{U}_T\mathbf{\Sigma}_T\mathbf{V}_T^\top$ where $\mathbf{\Sigma}_T$ is diagonal and contains the singular values of $\mathbf{T}$ sorted in decreasing order and $\mathbf{U}_T \in \mathbb{R}^{a\times r}$ and $\mathbf{V}_T \in \mathbb{R}^{b\times r}$ have orthogonal columns that contain the left and right singular vectors of $\mathbf{T}$ corresponding to its singular values. We denote by $\mathbf{T}_k$ the 'best' rank-$k$ approximation to $\mathbf{T}$, that is, $\mathbf{T}_k = \mathrm{argmin}_{\mathbf{V}\in\mathbb{R}^{a\times b},\mathrm{rank}(\mathbf{V})=k}\|\mathbf{T}-\mathbf{V}\|_\xi$, where $\xi \in \{2, F\}$ and $\|\cdot\|_2$ denotes the spectral norm and $\|\cdot\|_F$ the Frobenius norm of a matrix. We can describe this matrix in terms of its SVD as $\mathbf{T}_k = \mathbf{U}_{T,k}\mathbf{\Sigma}_{T,k}\mathbf{V}_{T,k}^\top$ where $\mathbf{\Sigma}_{T,k}$ is a diagonal matrix of the top $k$ singular values of $\mathbf{T}$ and $\mathbf{U}_{T,k}$ and $\mathbf{V}_{T,k}$ are the matrices formed by the associated left and right singular vectors.

Now let $\mathbf{K} \in \mathbb{R}^{n\times n}$ be a symmetric positive semidefinite (SPSD) kernel or Gram matrix with $\mathrm{rank}(\mathbf{K}) = r \le n$, that is, a symmetric matrix for which there exists an $\mathbf{X} \in \mathbb{R}^{N\times n}$ such that $\mathbf{K} = \mathbf{X}^\top\mathbf{X}$. We will write the SVD of $\mathbf{K}$ as $\mathbf{K} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$, where the columns of $\mathbf{U}$ are orthogonal and $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1,\dots,\sigma_r)$ is diagonal. The pseudo-inverse of $\mathbf{K}$ is defined as $\mathbf{K}^+ = \sum_{t=1}^r \sigma_t^{-1}\mathbf{U}^{(t)}{\mathbf{U}^{(t)}}^\top$, and $\mathbf{K}^+ = \mathbf{K}^{-1}$ when $\mathbf{K}$ is full rank. For $k < r$, $\mathbf{K}_k = \sum_{t=1}^k \sigma_t\mathbf{U}^{(t)}{\mathbf{U}^{(t)}}^\top = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{U}_k^\top$ is the 'best' rank-$k$ approximation to $\mathbf{K}$, that is, $\mathbf{K}_k = \mathrm{argmin}_{\mathbf{K}'\in\mathbb{R}^{n\times n},\mathrm{rank}(\mathbf{K}')=k}\|\mathbf{K}-\mathbf{K}'\|_{\xi\in\{2,F\}}$, with $\|\mathbf{K}-\mathbf{K}_k\|_2 = \sigma_{k+1}$ and $\|\mathbf{K}-\mathbf{K}_k\|_F = \sqrt{\sum_{t=k+1}^r \sigma_t^2}$ (Golub and Loan, 1983).

We will be focusing on generating an approximation $\widetilde{\mathbf{K}}$ of $\mathbf{K}$ based on a sample of $l \ll n$ of its columns. For now, we assume that the sample of $l$ columns is given to us, though the focus of this paper will be on various methods for selecting columns. Let $\mathbf{C}$ denote the $n \times l$ matrix formed by these columns and $\mathbf{W}$ the $l \times l$ matrix consisting of the intersection of these $l$ columns with the corresponding $l$ rows of $\mathbf{K}$. Note that $\mathbf{W}$ is SPSD since $\mathbf{K}$ is SPSD. Without loss of generality, the columns and rows of $\mathbf{K}$ can be rearranged based on this sampling so that $\mathbf{K}$ and $\mathbf{C}$ be written as follows:

$$\mathbf{K} = \begin{bmatrix} \mathbf{W} & \mathbf{K}_{21}^\top \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} \mathbf{W} \\ \mathbf{K}_{21} \end{bmatrix}. \tag{1}$$

### 2.1 Nyström Method

The Nyström method uses $\mathbf{W}$ and $\mathbf{C}$ from (1) to approximate $\mathbf{K}$. Assuming a uniform sampling of the columns, the Nyström method generates a rank-$k$ approximation $\widetilde{\mathbf{K}}$ of $\mathbf{K}$ for $k < n$ defined by:

$$\widetilde{\mathbf{K}}_k^{nys} = \mathbf{C}\mathbf{W}_k^+\mathbf{C}^\top \approx \mathbf{K},$$

where $\mathbf{W}_k$ is the best $k$-rank approximation of $\mathbf{W}$ with respect to the spectral or Frobenius norm and $\mathbf{W}_k^+$ denotes the pseudo-inverse of $\mathbf{W}_k$. The Nyström method thus approximates the top $k$ singular values ($\mathbf{\Sigma}_k$) and singular vectors ($\mathbf{U}_k$) of $\mathbf{K}$ as:

$$\widetilde{\mathbf{\Sigma}}_k^{nys} = \left(\frac{n}{l}\right)\mathbf{\Sigma}_{W,k} \quad \text{and} \quad \widetilde{\mathbf{U}}_k^{nys} = \sqrt{\frac{l}{n}}\mathbf{C}\mathbf{U}_{W,k}\mathbf{\Sigma}_{W,k}^+. \tag{2}$$

When $k = l$ (or more generally, whenever $k \geq \text{rank}(\mathbf{C})$), this approximation perfectly reconstructs three blocks of $\mathbf{K}$, and $\mathbf{K}_{22}$ is approximated by the Schur Complement of $\mathbf{W}$ in $\mathbf{K}$:

$$\widetilde{\mathbf{K}}_l^{nys} = \mathbf{C}\mathbf{W}^+\mathbf{C}^\top = \begin{bmatrix} \mathbf{W} & \mathbf{K}_{21}^\top \\ \mathbf{K}_{21} & \mathbf{K}_{21}\mathbf{W}^+\mathbf{K}_{21} \end{bmatrix}. \tag{3}$$

Since the running time complexity of SVD on $\mathbf{W}$ is in $O(kl^2)$ and matrix multiplication with $\mathbf{C}$ takes $O(kln)$, the total complexity of the Nyström approximation computation is in $O(kln)$.

### 2.2 Related Work

There has been a wide array of work on low-rank matrix approximation within the numerical linear algebra and computer science communities, much of which has been inspired by the celebrated result of Johnson and Lindenstrauss (1984), which showed that random low-dimensional embeddings preserve Euclidean geometry. This result has led to a family of random projection algorithms, which involves projecting the original matrix onto a random low-dimensional subspace (Papadimitriou et al., 1998; Indyk, 2006; Liberty, 2009). Alternatively, SVD can be used to generate 'optimal' low-rank matrix approximations, as mentioned earlier. However, both the random projection and the SVD algorithms involve storage and operating on the entire input matrix. SVD is more computationally expensive than random projection methods, though neither are linear in $n$ in terms of time and space complexity. When dealing with sparse matrices, there exist less computationally intensive techniques such as Jacobi, Arnoldi, Hebbian and more recent randomized methods (Golub and Loan, 1983; Gorrell, 2006; Rokhlin et al., 2009; Halko et al., 2009) for generating low-rank approximations. These methods require computation of matrix-vector products and thus require operating on every non-zero entry of the matrix, which may not be suitable for large, dense matrices. Matrix sparsification algorithms (Achlioptas and Mcsherry, 2007; Arora et al., 2006), as the name suggests, attempt to sparsify dense matrices to speed up future storage and computational burdens, though they too require storage of the input matrix and exhibit superlinear processing time.

Alternatively, sampling-based approaches can be used to generate low-rank approximations. Research in this area dates back to classical theoretical results that show, for any arbitrary matrix, the existence of a subset of $k$ columns for which the error in matrix projection (as defined in Kumar et al., 2009b) can be bounded relative to the optimal rank-$k$ approximation of the matrix (Ruston, 1962). Deterministic algorithms such as rank-revealing QR (Gu and Eisenstat, 1996) can achieve nearly optimal matrix projection errors. More recently, research in the theoretical computer science

community has been aimed at deriving bounds on matrix projection error using sampling-based approximations, including additive error bounds using sampling distributions based on the squared $L_2$ norms of the columns (Frieze et al., 1998; Drineas et al., 2006; Rudelson and Vershynin, 2007); relative error bounds using adaptive sampling techniques (Deshpande et al., 2006; Har-peled, 2006); and, relative error bounds based on distributions derived from the singular vectors of the input matrix, in work related to the column-subset selection problem (Drineas et al., 2008; Boutsidis et al., 2009). These sampling-based approximations all require visiting every entry of the matrix in order to get good performance guarantees for any matrix. However, as discussed in Kumar et al. (2009b), the task of matrix projection involves projecting the input matrix onto a low-rank subspace, which requires superlinear time and space with respect to $n$ and is not always feasible for large-scale matrices.

There does exist, however, another class of sampling-based approximation algorithms that only store and operate on a subset of the original matrix. For arbitrary rectangular matrices, these algorithms are known as 'CUR' approximations (the name 'CUR' corresponds to the three low-rank matrices whose product is an approximation to the original matrix). The theoretical performance of CUR approximations has been analyzed using a variety of sampling schemes, although the column-selection processes associated with these analyses often require operating on the entire input matrix (Goreinov et al., 1997; Stewart, 1999; Drineas et al., 2008; Mahoney and Drineas, 2009).

In the context of symmetric positive semidefinite matrices, the Nyström method is a commonly used algorithm to efficiently generate low-rank approximations. The Nyström method was initially introduced as a quadrature method for numerical integration, used to approximate eigenfunction solutions (Nyström, 1928; Baker, 1977). More recently, it was presented in Williams and Seeger (2000) to speed up kernel algorithms and has been studied theoretically using a variety of sampling schemes (Smola and Schölkopf, 2000; Drineas and Mahoney, 2005; Zhang et al., 2008; Zhang and Kwok, 2009; Kumar et al., 2009a,b,c; Belabbas and Wolfe, 2009; Belabbas and Wolfe, 2009; Cortes et al., 2010; Talwalkar and Rostamizadeh, 2010). It has also been used for a variety of machine learning tasks ranging from manifold learning to image segmentation (Platt, 2004; Fowlkes et al., 2004; Talwalkar et al., 2008). A closely related algorithm, known as the Incomplete Cholesky Decomposition (Fine and Scheinberg, 2002; Bach and Jordan, 2002, 2005), can also be viewed as a specific sampling technique associated with the Nyström method (Bach and Jordan, 2005). As noted by Candès and Recht (2009) and Talwalkar and Rostamizadeh (2010), the Nyström approximation is related to the problem of matrix completion (Candès and Recht, 2009; Candès and Tao, 2009), which attempts to complete a low-rank matrix from a random sample of its entries. However, the matrix completion attempts to impute a low-rank matrix from a subset of (possibly perturbed) matrix entries, rather than a subset of matrix columns. This problem is related to, yet distinct from the Nyström method and sampling-based low-rank approximation algorithms in general, that deal with full-rank matrices that are amenable to low-rank approximation. Furthermore, when we have access to the underlying kernel function that generates the kernel matrix of interest, we can generate matrix entries on-the-fly as desired, providing us with more flexibility accessing the original matrix.

## 3. Fixed Sampling

Since the Nyström method operates on a small subset of $\mathbf{K}$, that is, $\mathbf{C}$, the selection of columns can significantly influence the accuracy of the approximation. In the remainder of the paper, we will discuss various sampling options that aim to select informative columns from $\mathbf{K}$. We begin with the

most common class of sampling techniques that select columns using a fixed probability distribution. The most basic sampling technique involves *uniform* sampling of the columns. Alternatively, the $i$th column can be sampled non-uniformly with weight proportional to either its corresponding diagonal element $\mathbf{K}_{ii}$ (*diagonal sampling*) or the $L_2$ norm of the column (*column-norm sampling*) (Drineas et al., 2006; Drineas and Mahoney, 2005). There are additional computational costs associated with these non-uniform sampling methods: $O(n)$ time and space requirements for diagonal sampling and $O(n^2)$ time and space for column-norm sampling. These non-uniform sampling techniques are often presented using sampling with replacement to simplify theoretical analysis. Column-norm sampling has been used to analyze a general SVD approximation algorithm. Further, diagonal sampling with replacement was used by Drineas and Mahoney (2005) and Belabbas and Wolfe (2009) to bound the reconstruction error of the Nyström method.[2] In Drineas and Mahoney (2005) however, the authors suggest that column-norm sampling would be a better sampling assumption for the analysis of the Nyström method. We also note that Belabbas and Wolfe (2009) proposed a family of 'annealed determinantal' distributions for which multiplicative bounds on reconstruction error were derived. However, in practice, these distributions cannot be efficiently computed except for special cases coinciding with uniform and column-norm sampling. Similarly, although Mahoney and Drineas (2009) present multiplicative bounds for the CUR decomposition (which is quite similar to the Nyström method) when sampling from a distribution over the columns based on 'leverage scores,' these scores cannot be efficiently computed in practice for large-scale applications.

In the remainder of this section we present novel experimental results comparing the performance of these fixed sampling methods on several data sets. Previous studies have compared uniform and non-uniform in a more restrictive setting, using fewer types of kernels and focusing only on column-norm sampling (Drineas et al., 2001; Zhang et al., 2008). However, in this work, we provide the first comparison that includes diagonal sampling, which is the non-uniform distribution that is most scalable for large-scale applications and which has been used in some theoretical analyses of the Nyström method.

### 3.1 Data Sets

We used 5 data sets from a variety of applications, for example, computer vision and biology, as described in Table 1. SPSD kernel matrices were generated by mean centering the data sets and applying either a linear kernel or RBF kernel. The diagonals (respectively column norms) of these kernel matrices were used to calculate diagonal (respectively column-norm) distributions. Note that the diagonal distribution equals the uniform distribution for RBF kernels since diagonal entries of RBF kernel matrices always equal one.

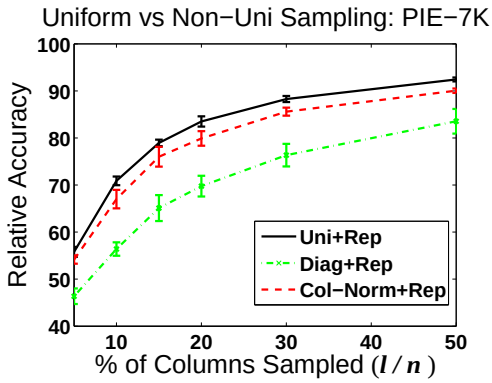
### 3.2 Experiments

We used the data sets described in the previous section to test the approximation accuracy for each sampling method. Low-rank approximations of $\mathbf{K}$ were generated using the Nyström method along with these sampling methods, and we measured the accuracy of reconstruction relative to the optimal

2. Although Drineas and Mahoney (2005) claim to weight each column proportionally to $\mathbf{K}_{ii}^2$, they in fact use the diagonal sampling we present in this work, that is, weights proportional to $\mathbf{K}_{ii}$ (Drineas, 2008).

| Name | Type | $n$ | $d$ | Kernel |
|---|---|---|---|---|
| PIE-2.7K | faces (profile) | 2731 | 2304 | linear |
| PIE-7K | faces (front) | 7412 | 2304 | linear |
| MNIST | digit images | 4000 | 784 | linear |
| ESS | proteins | 4728 | 16 | RBF |
| ABN | abalones | 4177 | 8 | RBF |

Table 1: Description of the data sets and kernels used in fixed and adaptive sampling experiments (Sim et al., 2002; LeCun and Cortes, 1998; Gustafson et al., 2006; Asuncion and Newman, 2007). '$d$' denotes the number of features in input space.




(a)

| $l/n$ | Data Set | Uniform+Rep | Diag+Rep | Col-Norm+Rep |
|---|---|---|---|---|
| 5% | PIE-2.7K | **38.8** (±**1.5**) | 38.3 (±0.9) | 37.0 (±0.9) |
| | PIE-7K | **55.8** (±**1.1**) | 46.4 (±1.7) | 54.2 (±0.9) |
| | MNIST | **47.4** (±**0.8**) | 46.9 (±0.7) | 45.6 (±1.0) |
| | ESS | **45.1** (±**2.3**) | - | 41.0 (±2.2) |
| | ABN | **47.3** (±**3.9**) | - | 44.2 (±1.2) |
| 20% | PIE-2.7K | **72.3** (±**0.9**) | 65.0 (±0.9) | 63.4 (±1.4) |
| | PIE-7K | **83.5** (±**1.1**) | 69.8 (±2.2) | 79.9 (±1.6) |
| | MNIST | **80.8** (±**0.5**) | 79.4 (±0.5) | 78.1 (±0.5) |
| | ESS | **80.1** (±**0.7**) | - | 75.5 (±1.1) |
| | ABN | **77.1** (±**3.0**) | - | 66.3 (±4.0) |

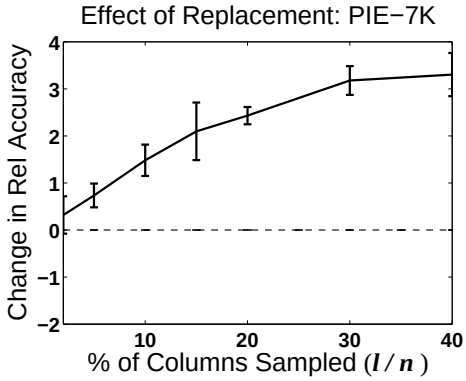
(b)

Figure 1: (a) Nyström relative accuracy for various sampling techniques on PIE-7K. (b) Nyström relative accuracy for various sampling methods for two values of $l/n$ with $k = 100$. Values in parentheses show standard deviations for 10 different runs for a fixed $l$. '+Rep' denotes sampling with replacement. No error ('-') is reported for diagonal sampling with RBF kernels since diagonal sampling is equivalent to uniform sampling in this case.

rank-$k$ approximation, $\mathbf{K}_k$, as:

$$\text{relative accuracy} = \frac{\|\mathbf{K} - \mathbf{K}_k\|_F}{\|\mathbf{K} - \widetilde{\mathbf{K}}_k\|_F} \times 100. \tag{4}$$

Note that the relative accuracy is lower bounded by zero and will approach one for good approximations. We fixed $k = 100$ for all experiments, a value that captures more than 90% of the spectral energy for each data set. We first compared the effectiveness of the three sampling techniques using sampling with replacement. The results for PIE-7K are presented in Figure 1(a) and summarized for all data sets in Figure 1(b). The results across all data sets show that uniform sampling outperforms all other methods, while being much cheaper computationally and space-wise. Thus, while non-uniform sampling techniques may be effective in extreme cases where a few columns of $\mathbf{K}$ dominate in terms of $\|\cdot\|_2$, this situation does not tend to arise with real-world data, where uniform sampling is most effective.




(a)

| Data Set | 5% | 10% | 15% | 30% |
|---|---|---|---|---|
| PIE-2.7K | 0.8 (±.6) | 1.7 (±.3) | 2.3 (±.9) | 4.4 (±.4) |
| PIE-7K | 0.7 (±.3) | 1.5 (±.3) | 2.1 (±.6) | 3.2 (±.3) |
| MNIST | 1.0 (±.5) | 1.9 (±.6) | 2.3 (±.4) | 3.4 (±.4) |
| ESS | 0.9 (±.9) | 1.8 (±.9) | 2.2 (±.6) | 3.7 (±.7) |
| ABN | 0.7 (±1.2) | 1.3 (±1.8) | 2.6 (±1.4) | 4.5 (±1.1) |

(b)

Figure 2: Comparison of uniform sampling with and without replacement measured by the difference in relative accuracy. (a) Improvement in relative accuracy for PIE-7K when sampling without replacement. (b) Improvement in relative accuracy when sampling without replacement across all data sets for various $l/n$ percentages.

Next, we compared the performance of uniform sampling with and without replacement. Figure 2(a) illustrates the effect of replacement for the PIE-7K data set for different $l/n$ ratios. Similar results for the remaining data sets are summarized in Figure 2(b). The results show that uniform sampling without replacement improves the accuracy of the Nyström method over sampling with replacement, even when sampling less than 5% of the total columns. In summary, these experimental

show that uniform sampling without replacement is the cheapest and most efficient sampling technique across several data sets (it is also the most commonly used method in practice). In Section 6, we present a theoretical analysis of the Nyström method using precisely this type of sampling.

## 4. Adaptive Sampling

In Section 3, we focused on fixed sampling schemes to create low-rank approximations. In this section, we discuss various sampling options that aim to select more informative columns from $\mathbf{K}$, while storing and operating on only $O(ln)$ entries of $\mathbf{K}$. The Sparse Matrix Greedy Approximation (SMGA) (Smola and Schölkopf, 2000) and the Incomplete Cholesky Decomposition (ICL) (Fine and Scheinberg, 2002; Bach and Jordan, 2002) were the first such adaptive schemes suggested for the Nyström method. SMGA is a matching-pursuit algorithm that randomly selects a new sample at each round from a random subset of $s \ll n$ samples, with $s = 59$ in practice as per the suggestion of Smola and Schölkopf (2000). The runtime to select $l$ columns is $O(sl^2n)$, which is of the same order as the Nyström method itself when $s$ is a constant and $k = l$ (see Section 2.1 for details).

Whereas SMGA was proposed as a sampling scheme to be used in conjunction with the Nyström method, ICL generates a low-rank factorization of $\mathbf{K}$ on-the-fly as it adaptively selects columns based on potential pivots of the Incomplete Cholesky Decomposition. ICL is a greedy, deterministic selection process that generates an approximation of the form $\widetilde{\mathbf{K}}^{icl} = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^\top$ where $\widetilde{\mathbf{X}} \in \mathbb{R}^{n\times l}$ is low-rank. The runtime of ICL is $O(l^2n)$. Although ICL does not generate an approximate SVD of $\mathbf{K}$, it does yield a low-rank approximation of $\mathbf{K}$ that can be used with the Woodbury approximation. Moreover, when $k = l$, the Nyström approximation generated from the $l$ columns of $\mathbf{K}$ associated with the pivots selected by ICL is identical to $\widetilde{\mathbf{K}}^{icl}$ (Bach and Jordan, 2005). Related greedy adaptive sampling techniques were proposed by Ouimet and Bengio (2005) and Liu et al. (2006) in the contexts of spectral embedding and spectral mesh processing, respectively.

More recently, Zhang et al. (2008) and Zhang and Kwok (2009) proposed a technique to generate informative columns using centroids resulting from $K$-means clustering, with $K = l$. This algorithm, which uses out-of-sample extensions to generate a set of $l$ representative columns of $\mathbf{K}$, has been shown to give good empirical accuracy (Zhang et al., 2008). Finally, an adaptive sampling technique with strong theoretical foundations (*adaptive-full*) was proposed in Deshpande et al. (2006). It requires a full pass through $\mathbf{K}$ in each iteration and is thus inefficient for large $\mathbf{K}$. In the remainder of this section, we first propose a novel adaptive technique that extends the ideas of Deshpande et al. (2006) and then present empirical results comparing the performance of this new algorithm with uniform sampling as well as SMGA, ICL, $K$-means and the *adaptive-full* techniques.

### 4.1 Adaptive Nyström Sampling

Instead of sampling all $l$ columns from a fixed distribution, adaptive sampling alternates between selecting a set of columns and updating the distribution over all the columns. Starting with an initial distribution over the columns, $s < l$ columns are chosen to form a submatrix $\mathbf{C}'$. The probabilities are then updated as a function of previously chosen columns and $s$ new columns are sampled and incorporated in $\mathbf{C}'$. This process is repeated until $l$ columns have been selected. The adaptive sampling scheme in Deshpande et al. (2006) is detailed in Figure 3. Note that the sampling step, UPDATE-PROBABILITY-FULL, requires a full pass over $\mathbf{K}$ at each step, and hence $O(n^2)$ time and space.

**Input**: $n \times n$ SPSD matrix ($\mathbf{K}$), number columns to be chosen ($l$), initial distribution over columns ($P_0$), number columns selected at each iteration ($s$)
**Output**: $l$ indices corresponding to columns of $\mathbf{K}$

SAMPLE-ADAPTIVE$(\mathbf{K}, n, l, P_0, s)$

$R \leftarrow$ set of $s$ indices sampled according to $P_0$
$t \leftarrow \frac{l}{s} - 1$ $\triangleright$ number of iterations
**for** $i \in [1 \ldots t]$ **do**
    $P_i \leftarrow$ UPDATE-PROBABILITY-FULL$(R)$
    $R_i \leftarrow$ set of $s$ indices sampled according to $P_i$
    $R \leftarrow R \cup R_i$
**return** $R$

UPDATE-PROBABILITY-FULL$(R)$

$\mathbf{C}' \leftarrow$ columns of $\mathbf{K}$ corresponding to indices in $R$
$\mathbf{U}_{C'} \leftarrow$ left singular vectors of $\mathbf{C}'$
$\mathbf{E} \leftarrow \mathbf{K} - \mathbf{U}_{C'}\mathbf{U}_{C'}^{\top}\mathbf{K}$
**for** $j \in [1 \ldots n]$ **do**
    **if** $j \in R$ **then**
        $P_j \leftarrow 0$
    **else** $P_j \leftarrow ||E_j||_2^2$
$P \leftarrow \frac{P}{||P||_2}$
**return** $P$

Figure 3: The adaptive sampling technique (Deshpande et al., 2006) that operates on the entire matrix $\mathbf{K}$ to compute the probability distribution over columns at each adaptive step.

We propose a simple sampling technique (*adaptive-partial*) that incorporates the advantages of adaptive sampling while avoiding the computational and storage burdens of the *adaptive-full* technique. At each iterative step, we measure the reconstruction error for each *row* of $\mathbf{C}'$ and the distribution over corresponding *columns* of $\mathbf{K}$ is updated proportional to this error. We compute the error for $\mathbf{C}'$, which is much smaller than $\mathbf{K}$, thus avoiding the $O(n^2)$ computation. As described in (3), if $k'$ is fixed to be the number of columns in $\mathbf{C}'$, it will lead to $\mathbf{C}'_{nys} = \mathbf{C}'$ resulting in perfect reconstruction of $\mathbf{C}'$. So, one must choose a smaller $k'$ to generate non-zero reconstruction errors from which probabilities can be updated (we used $k' = (\#\text{ columns in } \mathbf{C}')/2$ in our experiments). One artifact of using a $k'$ smaller than the rank of $\mathbf{C}'$ is that all the columns of $\mathbf{K}$ will have a non-zero probability of being selected, which could lead to the selection of previously selected columns in the next iteration. However, sampling *without* replacement strategy alleviates this problem. Working with $\mathbf{C}'$ instead of $\mathbf{K}$ to iteratively compute errors makes this algorithm significantly more efficient than that of Deshpande et al. (2006), as each iteration takes $O(nlk' + l^3)$ time and requires at most the storage of $l$ columns of $\mathbf{K}$. The details of the proposed sampling technique are outlined in Figure 4.

UPDATE-PROBABILITY-PARTIAL$(R)$

$\mathbf{C}' \leftarrow$ columns of $\mathbf{K}$ corresponding to indices in $R$
$k' \leftarrow$ CHOOSE-RANK() $\triangleright$ low-rank ($k$) or $\frac{|R|}{2}$
$\widetilde{\mathbf{\Sigma}}_{k'}^{nys}, \widetilde{\mathbf{U}}_{k'}^{nys} \leftarrow$ DO-NYSTRÖM $(\mathbf{C}', k')$ $\triangleright$ see Equation (2)
$\mathbf{C}'_{nys} \leftarrow$ Spectral reconstruction using $\widetilde{\mathbf{\Sigma}}_{k'}^{nys}, \widetilde{\mathbf{U}}_{k'}^{nys}$
$\mathbf{E} \leftarrow \mathbf{C}' - \mathbf{C}'_{nys}$
**for** $j \in [1 \ldots n]$ **do**
    **if** $j \in R$ **then**
        $P_j \leftarrow 0$ $\triangleright$ sample without replacement
    **else** $P_j \leftarrow ||\mathbf{E}_{(j)}||_2^2$
$P \leftarrow \frac{P}{||P||_2}$
**return** $P$

Figure 4: The proposed adaptive sampling technique that uses a small subset of the original matrix $\mathbf{K}$ to adaptively choose columns. It does not need to store or operate on $\mathbf{K}$.

| $l/n$% | Data Set | Uniform | ICL | SMGA | Adapt-Part | $K$-means | Adapt-Full |
|---|---|---|---|---|---|---|---|
| 5% | PIE-2.7K | 39.7 (0.7) | 41.6 (0.0) | 54.4 (0.6) | 42.6 (0.8) | **61.3** (0.5) | 44.2 (0.9) |
| | PIE-7K | 58.6 (1.0) | 50.1 (0.0) | 68.1 (0.9) | 61.4 (1.1) | **71.0** (0.7) | - |
| | MNIST | 47.5 (0.9) | 41.5 (0.0) | 59.2 (0.5) | 49.7 (0.9) | **72.9** (0.9) | 50.3 (0.7) |
| | ESS | 45.7 (2.6) | 25.2 (0.0) | 61.9 (0.5) | 49.3 (1.5) | **64.2** (1.6) | - |
| | ABN | 47.4 (5.5) | 15.6 (0.0) | 64.9 (1.8) | 23.0 (2.8) | **65.7** (5.8) | 50.7 (2.4) |
| 10% | PIE-2.7K | 58.2 (1.0) | 61.1 (0.0) | 72.7 (0.2) | 60.8 (1.0) | **73.0** (1.1) | 63.0 (0.3) |
| | PIE-7K | 72.4 (0.7) | 60.8 (0.0) | 74.5 (0.6) | 77.0 (0.6) | **82.8** (0.7) | - |
| | MNIST | 66.8 (1.4) | 58.3 (0.0) | 72.2 (0.8) | 69.3 (0.6) | **81.6** (0.6) | 68.5 (0.5) |
| | ESS | 66.8 (2.0) | 39.1 (0.0) | 74.7 (0.5) | 70.0 (1.0) | **81.6** (1.0) | - |
| | ABN | 61.0 (1.1) | 25.8 (0.0) | 67.1 (0.9) | 33.6 (6.7) | **79.8** (0.9) | 57.9 (3.9) |
| 20% | PIE-2.7K | 75.2 (1.0) | 80.5 (0.0) | **86.1** (0.2) | 78.7 (0.5) | 85.5 (0.5) | 80.6 (0.4) |
| | PIE-7K | 85.6 (0.9) | 69.5 (0.0) | 79.4 (0.5) | 86.2 (0.3) | **91.9** (0.3) | - |
| | MNIST | 83.6 (0.4) | 77.9 (0.0) | 78.7 (0.2) | 84.0 (0.6) | **88.4** (0.5) | 80.4 (0.5) |
| | ESS | 81.4 (2.1) | 55.3 (0.0) | 79.4 (0.7) | 83.4 (0.3) | **90.0** (0.6) | - |
| | ABN | 80.8 (1.7) | 41.2 (0.0) | 67.2 (2.2) | 44.4 (6.7) | **85.1** (1.6) | 62.4 (3.6) |

Table 2: Nyström spectral reconstruction accuracy for various sampling methods for all data sets for $k = 100$ and three $l/n$ percentages. Numbers in parenthesis indicate the standard deviations for 10 different runs for each $l$. Numbers in bold indicate the best performance on each data set, that is, each row of the table. Dashes ('-') indicate experiments that were too costly to run on the larger data sets (ESS, PIE-7K).

### 4.2 Experiments

We used the data sets in Table 1, and compared the effect of different sampling techniques on the relative accuracy of Nyström spectral reconstruction for $k = 100$. All experiments were conducted

| $l/n$% | Data Set | Uniform | ICL | SMGA | Adapt-Part | $K$-means | Adapt-Full |
|---|---|---|---|---|---|---|---|
| 5% | PIE-2.7K | 0.03 | 0.56 | 2.30 | 0.43 | 2.44 | 22.54 |
| | PIE-7K | 0.63 | 44.04 | 59.02 | 6.56 | 15.18 | - |
| | MNIST | 0.04 | 1.71 | 7.57 | 0.71 | 1.26 | 20.56 |
| | ESS | 0.07 | 2.87 | 62.42 | 0.85 | 3.48 | - |
| | ABN | 0.06 | 3.28 | 9.26 | 0.66 | 2.44 | 28.49 |
| 10% | PIE-2.7K | 0.08 | 2.81 | 8.44 | 0.97 | 3.25 | 23.13 |
| | PIE-7K | 0.63 | 44.04 | 244.33 | 6.56 | 15.18 | - |
| | MNIST | 0.20 | 7.38 | 28.79 | 1.51 | 1.82 | 21.77 |
| | ESS | 0.29 | 11.01 | 152.30 | 2.04 | 7.16 | - |
| | ABN | 0.23 | 10.92 | 33.30 | 1.74 | 4.94 | 35.91 |
| 20% | PIE-2.7K | 0.28 | 8.36 | 38.19 | 2.63 | 5.91 | 27.72 |
| | PIE-7K | 0.81 | 141.13 | 1107.32 | 13.80 | 12.08 | - |
| | MNIST | 0.46 | 16.99 | 51.96 | 4.03 | 2.91 | 26.53 |
| | ESS | 0.52 | 34.28 | 458.23 | 5.90 | 14.68 | - |
| | ABN | 1.01 | 38.36 | 199.43 | 8.54 | 12.56 | 97.39 |

Table 3: Run times (in seconds) corresponding to Nyström spectral reconstruction results in Table 2. Dashes ('-') indicate experiments that were too costly to run on the larger data sets (ESS, PIE-7K).

in Matlab on an x86 – 64 architecture using a single 2.4 Ghz core and 30GB of main memory. We used an implementation of ICL from Cawley and Talbot (2004) and an implementation of SMGA code from Smola (2000), using default parameters as set by these implementations. We wrote our own implementation of the $K$-means method using 5 iterations of $K$-means and employing an efficient (vectorized) function to compute $L_2$ distances between points and centroids at each iteration (Bunschoten, 1999).[3] Moreover, we used a random projection SVD solver to compute truncated SVD, using code by Tygert (2009).

The relative accuracy results across data sets for varying values of $l$ are presented in Table 2, while the corresponding timing results are detailed in Table 3. The $K$-means algorithm was clearly the best performing adaptive algorithm, generating the most accurate approximations in almost all settings in roughly the same amount of time (or less) as other adaptive algorithms. Moreover, the proposed Nyström adaptive technique, which is a natural extension of an important algorithm introduced in the theory community, has performance similar to this original algorithm at a fraction of the cost, but it is nonetheless outperformed by the $K$-means algorithm. We further note that ICL performs the worst of all the adaptive techniques, and it is often worse than random sampling (this observation is also noted by Zhang et al. 2008).

The empirical results also suggest that the performance gain due to adaptive sampling is inversely proportional to the percentage of sampled columns—random sampling actually outperforms many of the adaptive approaches when sampling 20% of the columns. These empirical results suggest a trade-off between time and space requirements, as noted by Schölkopf and Smola (2002)[Chapter 10.2]. Adaptive techniques spend more time to find a concise subset of informative columns, but as in the case of the $K$-means algorithm, can provide improved approximation accuracy.

3. Note that Matlab's built-in $K$-means function is quite inefficient.

## 5. Ensemble Sampling

In this section, we slightly shift focus, and discuss a meta algorithm called the *ensemble Nyström algorithm*. We treat each approximation generated by the Nyström method for a sample of $l$ columns as an *expert* and combine $p \geq 1$ such experts to derive an improved hypothesis, typically more accurate than any of the original experts.

The learning set-up is defined as follows. We assume a fixed kernel function $K\colon \mathcal{X}\times\mathcal{X}\to\mathbb{R}$ that can be used to generate the entries of a kernel matrix $\mathbf{K}$. The learner receives a set $S$ of $lp$ columns randomly selected from matrix $\mathbf{K}$ uniformly without replacement. $S$ is decomposed into $p$ subsets $S_1,\ldots,S_p$. Each subset $S_r$, $r\in[1,p]$, contains $l$ columns and is used to define a rank-$k$ Nyström approximation $\widetilde{\mathbf{K}}_r$.[4] Dropping the rank subscript $k$ in favor of the sample index $r$, $\widetilde{\mathbf{K}}_r$ can be written as $\widetilde{\mathbf{K}}_r = \mathbf{C}_r\mathbf{W}_r^+\mathbf{C}_r^\top$, where $\mathbf{C}_r$ and $\mathbf{W}_r$ denote the matrices formed from the columns of $S_r$ and $\mathbf{W}_r^+$ is the pseudo-inverse of the rank-$k$ approximation of $\mathbf{W}_r$. The learner further receives a sample $V$ of $s$ columns used to determine the weight $\mu_r\in\mathbb{R}$ attributed to each expert $\widetilde{\mathbf{K}}_r$. Thus, the general form of the approximation, $\mathbf{K}^{ens}$, generated by the ensemble Nyström algorithm, with $k\leq \operatorname{rank}(\mathbf{K}^{ens})\leq pk$, is

$$\begin{aligned}\widetilde{\mathbf{K}}^{ens} &= \sum_{r=1}^{p}\mu_r\widetilde{\mathbf{K}}_r \\ &= \begin{bmatrix}\mathbf{C}_1 & & \\ & \ddots & \\ & & \mathbf{C}_p\end{bmatrix}\begin{bmatrix}\mu_1\mathbf{W}_1^+ & & \\ & \ddots & \\ & & \mu_p\mathbf{W}_p^+\end{bmatrix}\begin{bmatrix}\mathbf{C}_1 & & \\ & \ddots & \\ & & \mathbf{C}_p\end{bmatrix}^\top. \end{aligned} \tag{5}$$

As noted by Li et al. (2010), (5) provides an alternative description of the ensemble Nyström method as a block diagonal approximation of $\mathbf{W}_{ens}^+$, where $\mathbf{W}_{ens}$ is the $lp\times lp$ SPSD matrix associated with the $lp$ sampled columns. Moreover, Li et al. (2010) further argues that computing $\mathbf{W}_{ens}^+$ would be preferable to making this block diagonal approximation and subsequently uses a random projection SVD solver to speed up computation of $\mathbf{W}_{ens}^+$ (Halko et al., 2009). However, this analysis is misleading as these two orthogonal approaches should not be viewed as competing methods. Rather, one can always use the ensemble based approach *along with* fast SVD solvers. This approach is most natural to improve performance on large-scale problems, and is precisely the approach we adopt in our experiments.

The mixture weights $\mu_r$ can be defined in many ways. The most straightforward choice consists of assigning equal weight to each expert, $\mu_r = 1/p$, $r\in[1,p]$. This choice does not require the additional sample $V$, but it ignores the relative quality of each Nyström approximation. Nevertheless, this simple *uniform method* already generates a solution superior to any one of the approximations $\widetilde{\mathbf{K}}_r$ used in the combination, as we shall see in the experimental section.

Another method, the *exponential weight method*, consists of measuring the reconstruction error $\hat{\varepsilon}_r$ of each expert $\widetilde{\mathbf{K}}_r$ over the validation sample $V$ and defining the mixture weight as $\mu_r = \exp(-\eta\hat{\varepsilon}_r)/Z$, where $\eta>0$ is a parameter of the algorithm and $Z$ a normalization factor ensuring that the vector $\mu=(\mu_1,\ldots,\mu_p)$ belongs to the unit simplex $\Delta$ of $\mathbb{R}^p$: $\Delta=\{\mu\in\mathbb{R}^p\colon \mu\geq 0\wedge\sum_{r=1}^p\mu_r=1\}$. The choice of the mixture weights here is similar to those used in the Weighted Majority algorithm

4. In this study, we focus on the class of base learners generated from Nyström approximation with uniform sampling of columns or from the adaptive $K$-means method. Alternatively, these base learners could be generated using other (or a combination of) sampling schemes discussed in Sections 3 and 4.

(Littlestone and Warmuth, 1994). Let $\mathbf{K}_V$ denote the matrix formed by using the samples from $V$ as its columns and let $\widetilde{\mathbf{K}}_r^V$ denote the submatrix of $\widetilde{\mathbf{K}}_r$ containing the columns corresponding to the columns in $V$. The reconstruction error $\hat{\varepsilon}_r = \|\widetilde{\mathbf{K}}_r^V - \mathbf{K}_V\|$ can be directly computed from these matrices.

A more general class of methods consists of using the sample $V$ to train the mixture weights $\mu_r$ to optimize a regression objective function such as the following:

$$\min_{\mu} \lambda\|\mu\|_2^2 + \|\sum_{r=1}^{p} \mu_r \widetilde{\mathbf{K}}_r^V - \mathbf{K}_V\|_F^2,$$

where $\lambda > 0$. This can be viewed as a ridge regression objective function and admits a closed form solution. We will refer to this method as the *ridge regression method*. Note that to ensure that the resulting matrix is SPSD for use in subsequent kernel-based algorithms, the optimization problem must be augmented with standard non-negativity constraints. This is not necessary however for reducing the reconstruction error, as in our experiments. Also, clearly, a variety of other regression algorithms such as Lasso can be used here instead.

The total complexity of the ensemble Nyström algorithm is $O(pl^3 + plkn + C_\mu)$, where $C_\mu$ is the cost of computing the mixture weights, $\mu$, used to combine the $p$ Nyström approximations. The mixture weights can be computed in constant time for the uniform method, in $O(psn)$ for the exponential weight method, or in $O(p^3 + p^2ns)$ for the ridge regression method where $O(p^2ns)$ time is required to compute a $p \times p$ matrix and $O(p^3)$ time is required for inverting this matrix. Furthermore, although the ensemble Nyström algorithm requires $p$ times more space and CPU cycles than the standard Nyström method, these additional requirements are quite reasonable in practice. The space requirement is still manageable for even large-scale applications given that $p$ is typically O(1) and $l$ is usually a very small percentage of $n$ (see Section 5.2 for further details). In terms of CPU requirements, we note that the algorithm can be easily parallelized, as all $p$ experts can be computed simultaneously. Thus, with a cluster of $p$ machines, the running time complexity of this algorithm is nearly equal to that of the standard Nyström algorithm with $l$ samples.

### 5.1 Ensemble Woodbury Approximation

The Woodbury approximation is a useful tool to use alongside low-rank approximations to efficiently (and approximately) invert kernel matrices. We are able to apply the Woodbury approximation since the Nyström method represents $\widetilde{\mathbf{K}}$ as the product of low-rank matrices. This is clear from the definition of the Woodbury approximation:

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}, \tag{6}$$

where $\mathbf{A} = \lambda\mathbf{I}$ and $\widetilde{\mathbf{K}} = \mathbf{BCD}$ in the context of the Nyström method. In contrast, the ensemble Nyström method represents $\widetilde{\mathbf{K}}$ as the sum of products of low-rank matrices, where each of the $p$ terms corresponds to a base learner. Hence, we cannot directly apply the Woodbury approximation as presented above. There is however, a natural extension of the Woodbury approximation in this setting, which at the simplest level involves running the approximation $p$ times. Starting with $p$ base learners with their associated weights, that is, $\widetilde{\mathbf{K}}_r$ and $\mu_r$ for $r \in [1, p]$, and defining $\mathbf{T}_0 = \lambda\mathbf{I}$, we

perform the following series of calculations:

$$\begin{aligned}
\mathbf{T}_1^{-1} &= (\mathbf{T}_0 + \mu_1 \widetilde{\mathbf{K}}_1)^{-1}, \\
\mathbf{T}_2^{-1} &= (\mathbf{T}_1 + \mu_2 \widetilde{\mathbf{K}}_2)^{-1}, \\
&\cdots \\
\mathbf{T}_p^{-1} &= (\mathbf{T}_{p-1} + \mu_p \widetilde{\mathbf{K}}_p)^{-1}.
\end{aligned}$$

To compute $\mathbf{T}_1^{-1}$, notice that we can use Woodbury approximation as stated in (6) since we can express $\mu_1 \widetilde{\mathbf{K}}_1$ as the product of low-rank matrices and we know that $T_0^{-1} = \frac{1}{\lambda}\mathbf{I}$. More generally, for $1 \leq i \leq p$, given an expression of $T_{i-1}^{-1}$ as a product of low-rank matrices, we can efficiently compute $T_i^{-1}$ using the Woodbury approximation (we use the low-rank structure to avoid ever computing or storing a full $n \times n$ matrix). Hence, after performing this series of $p$ calculations, we are left with the inverse of $\mathbf{T}_p$, which is exactly the quantity of interest since $\mathbf{T}_p = \lambda \mathbf{I} + \sum_{r=1}^{p} \mu_r \widetilde{\mathbf{K}}_r$. Although this algorithm requires $p$ iterations of the Woodbury approximation, these iterations can be parallelized in a tree-like fashion. Hence, when working on a cluster, using an ensemble Nyström approximation along with the Woodbury approximation requires only a $\log_2(p)$ factor more time than using the standard Nyström method.[5]

### 5.2 Experiments

In this section, we present experimental results that illustrate the performance of the ensemble Nyström method. We again work with the data sets listed in Table 1, and compare the performance of various methods for calculating the mixture weights ($\mu_r$). Throughout our experiments, we measure performance via relative accuracy (defined in (4)). For all experiments, we fixed the reduced rank to $k = 100$, and set the number of sampled columns to $l = 3\% \times n$.[6]

#### 5.2.1 ENSEMBLE NYSTRÖM WITH VARIOUS MIXTURE WEIGHTS

We first show results for the ensemble Nyström method using different techniques to choose the mixture weights, as previously discussed. In these experiments, we focused on base learners generated via the Nyström method with uniform sampling of columns. Furthermore, for the exponential and the ridge regression variants, we sampled a set of $s = 20$ columns and used an additional 20 columns ($s'$) as a hold-out set for selecting the optimal values of $\eta$ and $\lambda$. The number of approximations, $p$, was varied from 2 to 25. As a baseline, we also measured the maximum relative accuracy across the $p$ Nyström approximations used to construct $\widetilde{\mathbf{K}}^{ens}$. We also calculated the performance when using the optimal $\mu$, that is, we used least-square regression to find the best possible choice of combination weights for a fixed set of $p$ approximations by setting $s = n$. The results of these experiments are presented in Figure 5.[7] These results clearly show that the ensemble Nyström performance is significantly better than any of the individual Nyström approximations. We further note that the ensemble Nyström method tends to converge very quickly, and the most significant gain in performance occurs as $p$ increases from 2 to 10.

5. Note that we can also efficiently obtain singular values and singular vectors of the low-rank matrix $\mathbf{K}^{ens}$ using coherence-based arguments, as in Talwalkar and Rostamizadeh (2010).

6. Similar results (not reported here) were observed for other values of $k$ and $l$ as well.

7. Similar results (not reported here) were observed when measuring relative accuracy using the spectral norm instead of the Frobenium norm.

| Base Learner | Method | PIE-2.7K | PIE-7K | MNIST | ESS | ABN |
|---|---|---|---|---|---|---|
| Uniform | Average Base Learner | 26.9 | 46.3 | 34.2 | 30.0 | 38.1 |
| | Best Base Learner | 29.2 | 48.3 | 36.1 | 34.5 | 43.6 |
| | Ensemble Uniform | 33.0 | 57.5 | 47.3 | 43.9 | 49.8 |
| | Ensemble Exponential | 33.0 | 57.5 | 47.4 | 43.9 | 49.8 |
| | Ensemble Ridge | 35.0 | 58.5 | 54.0 | 44.5 | 53.6 |
| $K$-means | Average Base Learner | 47.6 | 62.9 | 62.5 | 42.2 | 60.6 |
| | Best Base Learner | 48.4 | 66.4 | 63.9 | 47.1 | 72.0 |
| | Ensemble Uniform | **54.9** | 71.3 | 76.9 | 52.2 | 76.4 |
| | Ensemble Exponential | **54.9** | 71.4 | 77.0 | 52.2 | 78.3 |
| | Ensemble Ridge | **54.9** | **71.6** | **77.2** | **52.7** | **79.0** |

Table 4: Relative accuracy for ensemble Nyström method with Nyström base learners generated with uniform sampling of columns or via the $K$-means algorithm.

#### 5.2.2 EFFECT OF RANK

As mentioned earlier, the rank of the ensemble approximations can be $p$ times greater than the rank of each of the base learners. Hence, to validate the results in Figure 5, we performed a simple experiment in which we compared the performance of the best base learner to the best rank $k$ approximation of the uniform ensemble approximation (obtained via SVD of the uniform ensemble approximation). We again used base learners generated via the Nyström method with uniform sampling of columns. The results of this experiment, presented in Figure 6, suggest that the performance gain of the ensemble methods is not due to this increased rank.

#### 5.2.3 EFFECT OF RIDGE

Figure 5 also shows that the ridge regression technique is the best of the proposed techniques, and generates nearly the optimal solution in terms of relative accuracy using the Frobenius norm. We also observed that when $s$ is increased to approximately 5% to 10% of $n$, linear regression without any regularization performs about as well as ridge regression for both the Frobenius and spectral norm. Figure 7 shows this comparison between linear regression and ridge regression for varying values of $s$ using a fixed number of experts ($p = 10$). In these experiments, we again used base learners generated via the Nyström method with uniform sampling of columns.

#### 5.2.4 ENSEMBLE $K$-MEANS NYSTRÖM

In the previous experiments, we focused on base learners generated via the Nyström method with uniform sampling of columns. In light of the performance of the $K$-means algorithm in Section 4, we next explored the performance of this algorithm when used in conjunction with the ensemble Nyström method. We fixed the number of base learners to $p = 10$ and when using ridge regression to learn weights, we set $s = s' = 20$. As shown in Table 4, similar performance gains in comparison to the average or best base learner can be seen when using an ensemble of base learners derived from the $K$-means algorithm. Consistent with the experimental results of Section 4, the accuracy values are higher for $K$-means relative to uniform sampling, though as noted in the previous section, this increased performance comes with an added cost, as the $K$-means step is more expensive than random sampling.

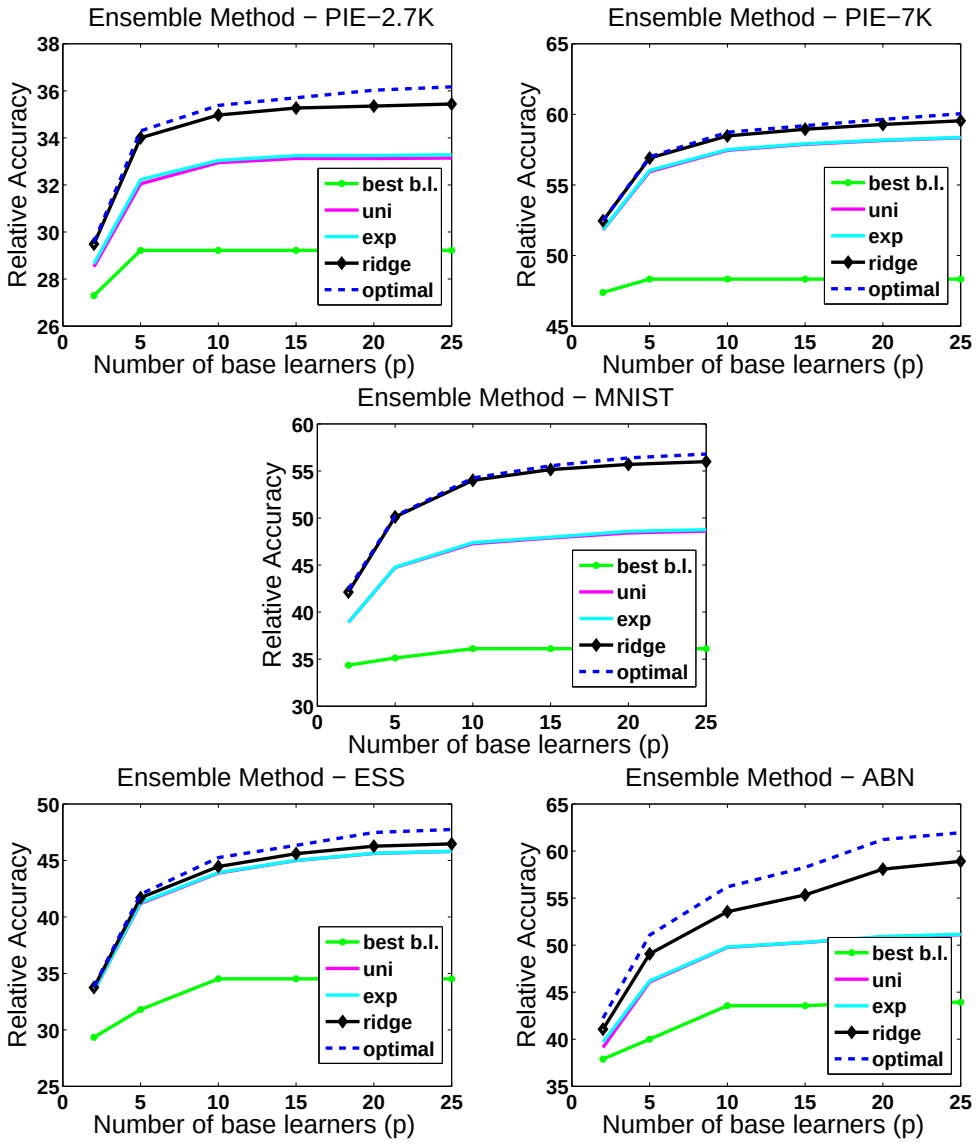



Figure 5: Relative accuracy for ensemble Nyström method using uniform ('uni'), exponential ('exp'), ridge ('ridge') and optimal ('optimal') mixture weights as well as the best ('best b.l.') of the $p$ base learners used to create the ensemble approximations.

## 6. Theoretical Analysis

We now present theoretical results that compare the quality of the Nyström approximation to the 'best' low-rank approximation, that is, the approximation constructed from the top singular values and singular vectors of $\mathbf{K}$. This work, related to work by Drineas and Mahoney (2005), provides performance bounds for the Nyström method as it is often used in practice, that is, using uniform

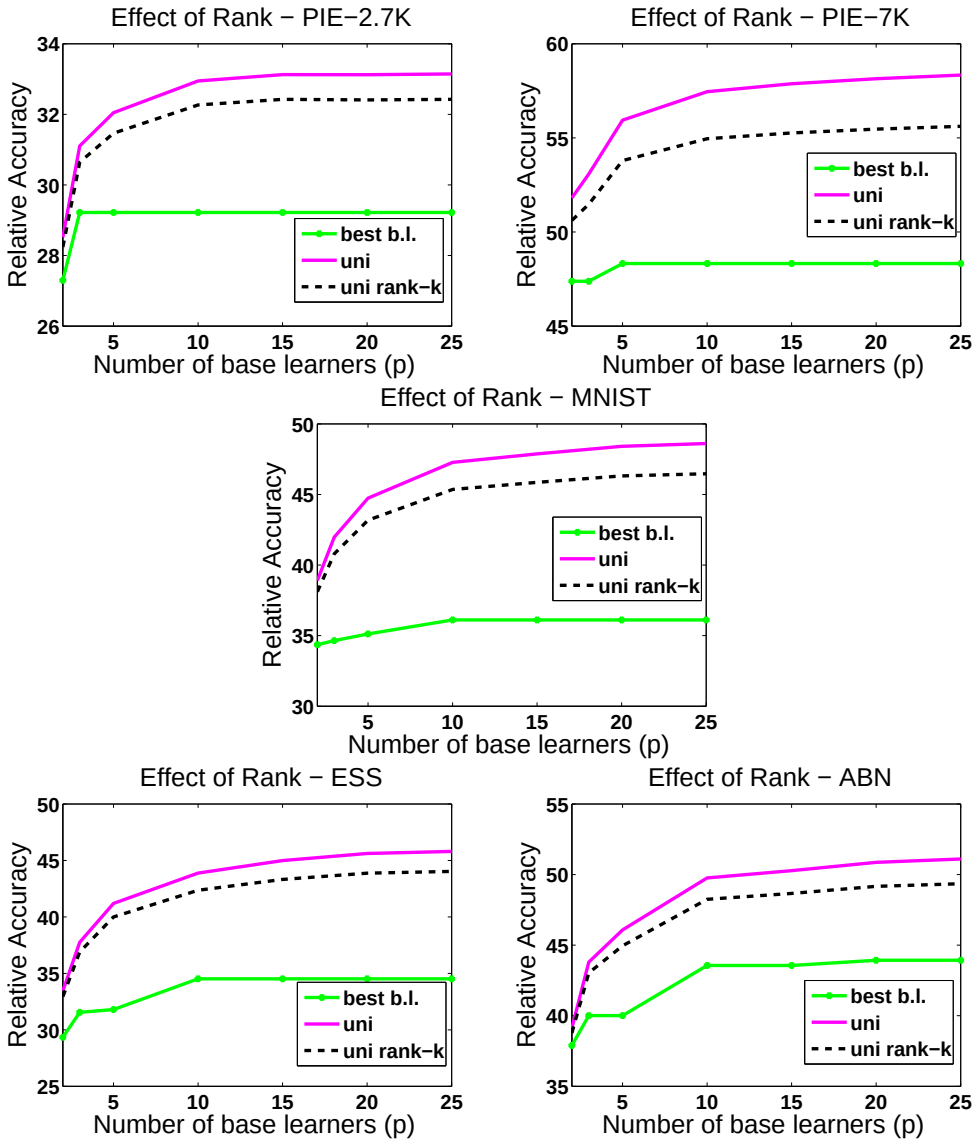



Figure 6: Relative accuracy for ensemble Nyström method using uniform ('uni') mixture weights, the optimal rank-$k$ approximation of the uniform ensemble result ('uni rank-$k$') as well as the best ('best b.l.') of the $p$ base learners used to create the ensemble approximations.

sampling without replacement, and holds for both the standard Nyström method as well as the ensemble Nyström method discussed in Section 5.

Our theoretical analysis of the Nyström method uses some results previously shown by Drineas and Mahoney (2005) as well as the following generalization of McDiarmid's concentration bound to sampling without replacement (Cortes et al., 2008).

**Theorem 1** *Let $Z_1, \ldots, Z_l$ be a sequence of random variables sampled uniformly without replacement from a fixed set of $l+u$ elements $Z$, and let $\phi\colon Z^l \to \mathbb{R}$ be a symmetric function such that for all*

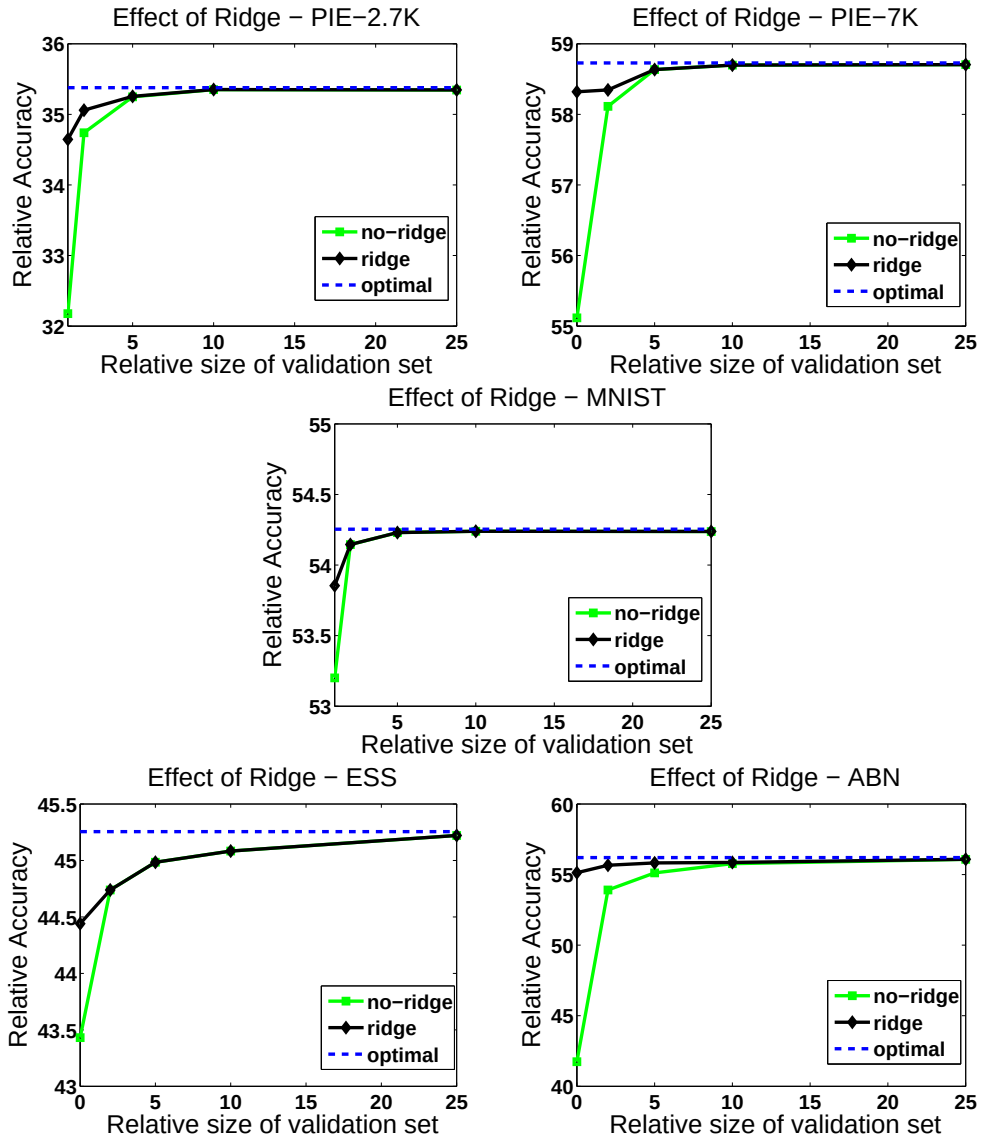



Figure 7: Comparison of relative accuracy for the ensemble Nyström method with $p = 10$ experts with weights derived from linear ('no-ridge') and ridge ('ridge') regression. The dotted line indicates the optimal combination. The relative size of the validation set equals $s/n \times 100$.

*$i \in [1,l]$ and for all $z_1,\ldots,z_l \in Z$ and $z'_1,\ldots,z'_l \in Z$, $|\phi(z_1,\ldots,z_l) - \phi(z_1,\ldots,z_{i-1},z'_i,z_{i+1},\ldots,z_l)| \leq c$. Then, for all $\varepsilon > 0$, the following inequality holds:*

$$\Pr\left[\phi - \mathbf{E}[\phi] \geq \varepsilon\right] \leq \exp\left[\frac{-2\varepsilon^2}{\alpha(l,u)c^2}\right],$$

*where* $\alpha(l,u) = \frac{lu}{l+u-1/2}\,\frac{1}{1-1/(2\max\{l,u\})}$.

We define the *selection matrix* corresponding to a sample of $l$ columns as the matrix $\mathbf{S} \in \mathbb{R}^{n \times l}$ defined by $\mathbf{S}_{ii} = 1$ if the $i$th column of $\mathbf{K}$ is among those sampled, $\mathbf{S}_{ij} = 0$ otherwise. Thus, $\mathbf{C} = \mathbf{KS}$ is the matrix formed by the columns sampled. Since $\mathbf{K}$ is SPSD, there exists $\mathbf{X} \in \mathbb{R}^{N \times n}$ such that $\mathbf{K} = \mathbf{X}^\top \mathbf{X}$. We shall denote by $\mathbf{K}_{\max}$ the maximum diagonal entry of $\mathbf{K}$, $\mathbf{K}_{\max} = \max_i \mathbf{K}_{ii}$, and by $d^{\mathbf{K}}_{\max}$ the distance $\max_{ij} \sqrt{\mathbf{K}_{ii} + \mathbf{K}_{jj} - 2\mathbf{K}_{ij}}$.

### 6.1 Standard Nyström Method

The following theorem gives an upper bound on the norm-2 error of the Nyström approximation of the form $\|\mathbf{K} - \widetilde{\mathbf{K}}\|_2 / \|\mathbf{K}\|_2 \leq \|\mathbf{K} - \mathbf{K}_k\|_2 / \|\mathbf{K}\|_2 + O(1/\sqrt{l})$ and an upper bound on the Frobenius error of the Nyström approximation of the form $\|\mathbf{K} - \widetilde{\mathbf{K}}\|_F / \|\mathbf{K}\|_F \leq \|\mathbf{K} - \mathbf{K}_k\|_F / \|\mathbf{K}\|_F + O(1/l^{\frac{1}{4}})$.

**Theorem 2** *Let $\widetilde{\mathbf{K}}$ denote the rank-k Nyström approximation of $\mathbf{K}$ based on $l$ columns sampled uniformly at random without replacement from $\mathbf{K}$, and $\mathbf{K}_k$ the best rank-k approximation of $\mathbf{K}$. Then, with probability at least $1 - \delta$, the following inequalities hold for any sample of size $l$:*

$$\|\mathbf{K} - \widetilde{\mathbf{K}}\|_2 \leq \|\mathbf{K} - \mathbf{K}_k\|_2 + \tfrac{2n}{\sqrt{l}} \mathbf{K}_{\max} \left[ 1 + \sqrt{\tfrac{n-l}{n-1/2} \tfrac{1}{\beta(l,n)} \log \tfrac{1}{\delta}} \, d^{\mathbf{K}}_{\max} / \mathbf{K}^{\frac{1}{2}}_{\max} \right],$$

$$\|\mathbf{K} - \widetilde{\mathbf{K}}\|_F \leq \|\mathbf{K} - \mathbf{K}_k\|_F + \left\lceil \tfrac{64k}{l} \right\rceil^{\frac{1}{4}} n \mathbf{K}_{\max} \left[ 1 + \sqrt{\tfrac{n-l}{n-1/2} \tfrac{1}{\beta(l,n)} \log \tfrac{1}{\delta}} \, d^{\mathbf{K}}_{\max} / \mathbf{K}^{\frac{1}{2}}_{\max} \right]^{\frac{1}{2}},$$

*where* $\beta(l,n) = 1 - \frac{1}{2\max\{l, n-l\}}$.

**Proof** To bound the norm-2 error of the Nyström method in the scenario of sampling without replacement, we start with the following general inequality given by Drineas and Mahoney (2005)[Proof of Lemma 4]:

$$\|\mathbf{K} - \widetilde{\mathbf{K}}\|_2 \leq \|\mathbf{K} - \mathbf{K}_k\|_2 + 2\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_2,$$

where $\mathbf{Z} = \sqrt{\frac{n}{l}} \mathbf{XS}$. We then apply the McDiarmid-type inequality of Theorem 1 to $\phi(\mathbf{S}) = \|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_2$. Let $\mathbf{S}'$ be a sampling matrix selecting the same columns as $\mathbf{S}$ except for one, and let $\mathbf{Z}'$ denote $\sqrt{\frac{n}{l}} \mathbf{XS}'$. Let $\mathbf{z}$ and $\mathbf{z}'$ denote the only differing columns of $\mathbf{Z}$ and $\mathbf{Z}'$, then

$$\begin{aligned} |\phi(\mathbf{S}') - \phi(\mathbf{S})| &\leq \|\mathbf{z}'\mathbf{z}'^\top - \mathbf{z}\mathbf{z}^\top\|_2 = \|(\mathbf{z}' - \mathbf{z})\mathbf{z}'^\top + \mathbf{z}(\mathbf{z}' - \mathbf{z})^\top\|_2 \\ &\leq 2\|\mathbf{z}' - \mathbf{z}\|_2 \max\{\|\mathbf{z}\|_2, \|\mathbf{z}'\|_2\}. \end{aligned}$$

Columns of $\mathbf{Z}$ are those of $\mathbf{X}$ scaled by $\sqrt{n/l}$. The norm of the difference of two columns of $\mathbf{X}$ can be viewed as the norm of the difference of two feature vectors associated to $\mathbf{K}$ and thus can be bounded by $d_{\mathbf{K}}$. Similarly, the norm of a single column of $\mathbf{X}$ is bounded by $\mathbf{K}^{\frac{1}{2}}_{\max}$. This leads to the following inequality:

$$|\phi(\mathbf{S}') - \phi(\mathbf{S})| \leq \frac{2n}{l} d^{\mathbf{K}}_{\max} \mathbf{K}^{\frac{1}{2}}_{\max}. \tag{7}$$

The expectation of $\phi$ can be bounded as follows:

$$\mathbf{E}[\Phi] = \mathbf{E}[\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_2] \leq \mathbf{E}[\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_F] \leq \frac{n}{\sqrt{l}} \mathbf{K}_{\max}, \tag{8}$$

where the last inequality follows Corollary 2 of Kumar et al. (2009a). The inequalities (7) and (8) combined with Theorem 1 give a bound on $\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_2$ and yield the statement of the theorem.

The following general inequality holds for the Frobenius error of the Nyström method (Drineas and Mahoney, 2005):

$$\|\mathbf{K}-\widetilde{\mathbf{K}}\|_F^2 \leq \|\mathbf{K}-\mathbf{K}_k\|_F^2 + \sqrt{64k}\,\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_F^2\, n\mathbf{K}_{ii}^{\max}. \tag{9}$$

Bounding the term $\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_F^2$ as in the norm-2 case and using the concentration bound of Theorem 1 yields the result of the theorem. ■

### 6.2 Ensemble Nyström Method

The following error bounds hold for ensemble Nyström methods based on a convex combination of Nyström approximations.

**Theorem 3** *Let $S$ be a sample of $pl$ columns drawn uniformly at random without replacement from $\mathbf{K}$, decomposed into $p$ subsamples of size $l$, $S_1,\ldots,S_p$. For $r\in[1,p]$, let $\widetilde{\mathbf{K}}_r$ denote the rank-$k$ Nyström approximation of $\mathbf{K}$ based on the sample $S_r$, and let $\mathbf{K}_k$ denote the best rank-$k$ approximation of $\mathbf{K}$. Then, with probability at least $1-\delta$, the following inequalities hold for any sample $S$ of size $pl$ and for any $\mu$ in the unit simplex $\Delta$ and $\widetilde{\mathbf{K}}^{ens}=\sum_{r=1}^p \mu_r\widetilde{\mathbf{K}}_r$:*

$$\begin{aligned}
\|\mathbf{K}-\widetilde{\mathbf{K}}^{ens}\|_2 &\leq \|\mathbf{K}-\mathbf{K}_k\|_2 + \\
&\qquad \tfrac{2n}{\sqrt{l}}\mathbf{K}_{\max}\Big[1+\mu_{\max}p^{\frac{1}{2}}\sqrt{\tfrac{n-pl}{n-1/2}\tfrac{1}{\beta(pl,n)}\log\tfrac{1}{\delta}}\;d_{\max}^{\mathbf{K}}/\mathbf{K}_{\max}^{\frac{1}{2}}\Big],\\
\|\mathbf{K}-\widetilde{\mathbf{K}}^{ens}\|_F &\leq \|\mathbf{K}-\mathbf{K}_k\|_F + \\
&\qquad \big[\tfrac{64k}{l}\big]^{\frac{1}{4}} n\mathbf{K}_{\max}\Big[1+\mu_{\max}p^{\frac{1}{2}}\sqrt{\tfrac{n-pl}{n-1/2}\tfrac{1}{\beta(pl,n)}\log\tfrac{1}{\delta}}\;d_{\max}^{\mathbf{K}}/\mathbf{K}_{\max}^{\frac{1}{2}}\Big]^{\frac{1}{2}},
\end{aligned}$$

*where* $\beta(pl,n)=1-\frac{1}{2\max\{pl,n-pl\}}$ *and* $\mu_{\max}=\max_{r=1}^p\mu_r$.

**Proof** For $r\in[1,p]$, let $\mathbf{Z}_r=\sqrt{n/l}\,\mathbf{X}\mathbf{S}_r$, where $\mathbf{S}_r$ denotes the selection matrix corresponding to the sample $S_r$. By definition of $\widetilde{\mathbf{K}}^{ens}$ and the upper bound on $\|\mathbf{K}-\widetilde{\mathbf{K}}_r\|_2$ already used in the proof of theorem 2, the following holds:

$$\begin{aligned}
\|\mathbf{K}-\widetilde{\mathbf{K}}^{ens}\|_2 &= \Big\|\sum_{r=1}^p \mu_r(\mathbf{K}-\widetilde{\mathbf{K}}_r)\Big\|_2 \leq \sum_{r=1}^p \mu_r\|\mathbf{K}-\widetilde{\mathbf{K}}_r\|_2\\
&\leq \sum_{r=1}^p \mu_r\big(\|\mathbf{K}-\mathbf{K}_k\|_2 + 2\|\mathbf{X}\mathbf{X}^\top-\mathbf{Z}_r\mathbf{Z}_r^\top\|_2\big)\\
&= \|\mathbf{K}-\mathbf{K}_k\|_2 + 2\sum_{r=1}^p \mu_r\|\mathbf{X}\mathbf{X}^\top-\mathbf{Z}_r\mathbf{Z}_r^\top\|_2.
\end{aligned}$$

We apply Theorem 1 to $\phi(S)=\sum_{r=1}^p \mu_r\|\mathbf{X}\mathbf{X}^\top-\mathbf{Z}_r\mathbf{Z}_r^\top\|_2$. Let $S'$ be a sample differing from $S$ by only one column. Observe that changing one column of the full sample $S$ changes only one subsample $S_r$ and thus only one term $\mu_r\|\mathbf{X}\mathbf{X}^\top-\mathbf{Z}_r\mathbf{Z}_r^\top\|_2$. Thus, in view of the bound (7) on the change to $\|\mathbf{X}\mathbf{X}^\top-\mathbf{Z}_r\mathbf{Z}_r^\top\|_2$, the following holds:

$$|\phi(S')-\phi(S)| \leq \frac{2n}{l}\mu_{\max}d_{\max}^{\mathbf{K}}\mathbf{K}_{\max}^{\frac{1}{2}}, \tag{10}$$

The expectation of $\Phi$ can be straightforwardly bounded by:

$$\mathbf{E}[\Phi(S)] = \sum_{r=1}^{p} \mu_r \mathbf{E}[\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}_r\mathbf{Z}_r^\top\|_2] \leq \sum_{r=1}^{p} \mu_r \frac{n}{\sqrt{l}}\mathbf{K}_{\max} = \frac{n}{\sqrt{l}}\mathbf{K}_{\max}$$

using the bound (8) for a single expert. Plugging in this upper bound and the Lipschitz bound (10) in Theorem 1 yields the norm-2 bound for the ensemble Nyström method.

For the Frobenius error bound, using the convexity of the Frobenius norm square $\|\cdot\|_F^2$ and the general inequality (9), we can write

$$\begin{aligned}
\|\mathbf{K} - \widetilde{\mathbf{K}}^{ens}\|_F^2 &= \Big\| \sum_{r=1}^{p} \mu_r (\mathbf{K} - \widetilde{\mathbf{K}}_r) \Big\|_F^2 \leq \sum_{r=1}^{p} \mu_r \|\mathbf{K} - \widetilde{\mathbf{K}}_r\|_F^2 \\
&\leq \sum_{r=1}^{p} \mu_r \Big[ \|\mathbf{K} - \mathbf{K}_k\|_F^2 + \sqrt{64k}\, \|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}_r\mathbf{Z}_r^\top\|_F\, n\mathbf{K}_{ii}^{\max} \Big]. \\
&= \|\mathbf{K} - \mathbf{K}_k\|_F^2 + \sqrt{64k} \sum_{r=1}^{p} \mu_r \|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}_r\mathbf{Z}_r^\top\|_F\, n\mathbf{K}_{ii}^{\max}.
\end{aligned}$$

The result follows by the application of Theorem 1 to $\psi(S) = \sum_{r=1}^{p} \mu_r \|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}_r\mathbf{Z}_r^\top\|_F$ in a way similar to the norm-2 case. ■

The bounds of Theorem 3 are similar in form to those of Theorem 2. However, the bounds for the ensemble Nyström are tighter than those for any Nyström expert based on a single sample of size $l$ even for a uniform weighting. In particular, for $\mu_i = 1/p$ for all $i$, the last term of the ensemble bound for norm-2 is smaller by a factor larger than $\mu_{\max} p^{\frac{1}{2}} = 1/\sqrt{p}$.

## 7. Conclusion

A key aspect of sampling-based matrix approximations is the method for the selection of representative columns. We discussed both fixed and adaptive methods for sampling the columns of a matrix. We saw that the approximation performance is significantly affected by the choice of the sampling algorithm and also that there is a tradeoff between choosing a more informative set of columns and the efficiency of the sampling algorithm. Furthermore, we introduced and discussed a new meta-algorithm based on an ensemble of several matrix approximations that generates favorable matrix reconstructions using base learners derived from either fixed or adaptive sampling schemes, and naturally fits within a distributed computing environment, thus making it quite efficient even in large-scale settings. We concluded with a theoretical analysis of the Nyström method (both the standard approach and the ensemble method) as it is often used in practice, namely using uniform sampling without replacement.

## Acknowledgments

AT was supported by NSF award No. 1122732. We thank the editor and the reviewers for several insightful comments that helped improve the original version of this paper.

## References

Dimitris Achlioptas and Frank Mcsherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, 54(2), 2007.

Sanjeev Arora, Elad Hazan, and Satyen Kale. A fast random sampling algorithm for sparsifying matrices. In *Approx-Random*, 2006.

Arthur Asuncion and David Newman. UCI machine learning repository. `http://www.ics.uci.edu/ mlearn/MLRepository.html`, 2007.

Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

Francis R. Bach and Michael I. Jordan. Predictive low-rank decomposition for kernel methods. In *International Conference on Machine Learning*, 2005.

Christopher T. Baker. *The Numerical Treatment of Integral Equations*. Clarendon Press, Oxford, 1977.

Mohamed A. Belabbas and Patrick J. Wolfe. Spectral methods in machine learning and new strategies for very large datasets. *Proceedings of the National Academy of Sciences of the United States of America*, 106(2):369–374, January 2009. ISSN 1091-6490.

Mohamed A. Belabbas and Patrick J. Wolfe. On landmark selection and sampling in high-dimensional data analysis. `arXiv:0906.4582v1 [stat.ML]`, 2009.

Bernhard E. Boser, Isabelle Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Conference on Learning Theory*, 1992.

Christos Boutsidis, Michael W. Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Symposium on Discrete Algorithms*, 2009.

Roland Bunschoten. `http://www.mathworks.com/matlabcentral/fileexchange/71-distance-m/`, 1999.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. `arXiv:0903.1476v1 [cs.IT]`, 2009.

Gavin Cawley and Nicola Talbot. Miscellaneous matlab software. `http://theoval.cmp.uea.ac.uk/matlab/default.html#cholinc`, 2004.

Corinna Cortes and Vladimir N. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Corinna Cortes, Mehryar Mohri, Dmitry Pechyony, and Ashish Rastogi. Stability of transductive regression algorithms. In *International Conference on Machine Learning*, 2008.

Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the impact of kernel approximation on learning accuracy. In *Conference on Artificial Intelligence and Statistics*, 2010.

Vin de Silva and Joshua Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Neural Information Processing Systems*, 2003.

Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. In *Symposium on Discrete Algorithms*, 2006.

Petros Drineas. Personal communication, 2008.

Petros Drineas and Michael W. Mahoney. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

Petros Drineas, Eleni Drinea, and Patrick S. Huggins. An experimental evaluation of a Monte-Carlo algorithm for svd. In *Panhellenic Conference on Informatics*, 2001.

Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices ii: computing a low-rank approximation to a matrix. *SIAM Journal of Computing*, 36(1), 2006.

Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

Shai Fine and Katya Scheinberg. Efficient svm training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2002.

Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the Nyström method. *Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.

Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In *Foundation of Computer Science*, 1998.

Gene Golub and Charles Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 2nd edition, 1983. ISBN 0-8018-3772-3 (hardcover), 0-8018-3739-1 (paperback).

Sergei A. Goreinov, Eugene E. Tyrtyshnikov, and Nickolai L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra and Its Applications*, 261:1–21, 1997.

Genevieve Gorrell. Generalized Hebbian algorithm for incremental singular value decomposition in natural language processing. In *European Chapter of the Association for Computational Linguistics*, 2006.

Ming Gu and Stanley C. Eisenstat. Efficient algorithms for computing a strong rank-revealing qr factorization. *SIAM Journal of Scientific Computing*, 17(4):848–869, 1996.

Adam Gustafson, Evan Snitkin, Stephen Parker, Charles DeLisi, and Simon Kasif. Towards the identification of essential genes using targeted genome sequencing and comparative analysis. *BMC:Genomics*, 7:265, 2006.

Nathan Halko, Per Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: stochastic algorithms for constructing approximate matrix decompositions. arXiv:0909.4061v1 [math.NA], 2009.

Sariel Har-peled. Low-rank matrix approximation in linear time, manuscript, 2006.

Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM*, 53(3):307–323, 2006.

William B. Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling techniques for the Nyström method. In *Conference on Artificial Intelligence and Statistics*, 2009a.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. On sampling-based approximate spectral decomposition. In *International Conference on Machine Learning*, 2009b.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Ensemble Nyström method. In *Neural Information Processing Systems*, 2009c.

Yann LeCun and Corinna Cortes. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998.

Mu Li, James T. Kwok, and Bao-Liang Lu. Making large-scale Nyström approximation possible. In *International Conference on Machine Learning*, 2010.

Edo Liberty. *Accelerated Dense Random Projections*. Ph.D. thesis, computer science department, Yale University, New Haven, CT, 2009.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

Rong Liu, Varun Jain, and Hao Zhang. Subsampling for efficient spectral mesh processing. In *Computer Graphics International Conference*, 2006.

Michael W Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

Evert J. Nyström. Über die praktische auflösung von linearen integralgleichungen mit anwendungen auf randwertaufgaben der potentialtheorie. *Commentationes Physico-Mathematicae*, 4(15):1–52, 1928.

Marie Ouimet and Yoshua Bengio. Greedy spectral embedding. In *Artificial Intelligence and Statistics*, 2005.

Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: a probabilistic analysis. In *Principles of Database Systems*, 1998.

John C. Platt. Fast embedding of sparse similarity graphs. In *Neural Information Processing Systems*, 2004.

Vladimir Rokhlin, Arthur Szlam, and Mark Tygert. A randomized algorithm for principal component analysis. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1100–1124, 2009.

Mark Rudelson and Roman Vershynin. Sampling from large matrices: an approach through geometric functional analysis. *Journal of the ACM*, 54(4):21, 2007.

Anthony F. Ruston. Auerbach's theorem and tensor products of banach spaces. *Mathematical Proceedings of the Cambridge Philosophical Society*, 58:476–480, 1962.

Bernhard Schölkopf and Alex Smola. *Learning with Kernels*. MIT Press: Cambridge, MA, 2002.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression database. In *Conference on Automatic Face and Gesture Recognition*, 2002.

Alex J. Smola. SVLab. `http://alex.smola.org/data/svlab.tgz`, 2000.

Alex J. Smola and Bernhard Schölkopf. Sparse greedy matrix approximation for machine learning. In *International Conference on Machine Learning*, 2000.

G. W. Stewart. Four algorithms for the efficient computation of truncated pivoted qr approximations to a sparse matrix. *Numerische Mathematik*, 83(2):313–323, 1999.

Ameet Talwalkar and Afshin Rostamizadeh. Matrix coherence and the Nyström method. In *Conference on Uncertainty in Artificial Intelligence*, 2010.

Ameet Talwalkar, Sanjiv Kumar, and Henry Rowley. Large-scale manifold learning. In *Conference on Vision and Pattern Recognition*, 2008.

Mark Tygert. `http://www.mathworks.com/matlabcentral/fileexchange/21524-principal-component-analysis`, 2009.

Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Neural Information Processing Systems*, 2000.

Kai Zhang and James T. Kwok. Density-weighted Nyström method for computing large kernel eigensystems. *Neural Computation*, 21(1):121–146, 2009.

Kai Zhang, Ivor Tsang, and James Kwok. Improved Nyström low-rank approximation and error analysis. In *International Conference on Machine Learning*, 2008.