# Matrix Approximation for Large-scale Learning

by

Ameet Talwalkar

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

Courant Institute of Mathematical Sciences

New York University

May 2010

Mehryar Mohri—Advisor

*For Aai and Baba*

# Acknowledgments

I would first like to thank my advisor, Mehryar Mohri, for his guidance throughout my doctoral studies. He gave me an opportunity to pursue a PhD, patiently taught me about the field of machine learning and guided me towards exciting research questions. He also introduced me to my mentors and collaborators at Google Research, Sanjiv Kumar and Corinna Cortes, both of whom have been tremendous role models for me throughout my studies. I would also like to thank the final two members of my thesis committee, Dennis Shasha and Mark Tygert, as well as Subhash Khot, who sat on my DQE and thesis proposal, for their encouragement and helpful advice.

During my time at Courant and my summers at Google, I have had the good fortune to work and interact with several other exceptional people. In particular, I would like to thank Eugene Weinstein, Ameesh Makadia, Cyril Allauzen, Dejan Jovanović, Shaila Musharoff, Ashish Rastogi, Rosemary Amico, Michael Riley, Henry Rowley and Jeremy Shute for helping me along the way and making my studies and research more enjoyable over these past four years. I would especially like to thank my partner in crime, Afshin Rostamizadeh, for being a supportive officemate and a considerate friend throughout our countless hours working together.

Last, but not least, I would like to thank my friends and family for their unwavering support. In particular, I have consistently drawn strength from my lovely girlfriend Jessica, my brother Jaideep, my sister-in-law Kristen and the three cutest little men in the world, my nephews Kavi, Nayan and Dev. And to my parents, Rohini and Shrirang, to whom this thesis is dedicated, I am infinitely grateful. They are my sources of inspiration and my greatest teachers, and any achievement I may have is a credit to them. Thank you, Aai and Baba.

# Abstract

Modern learning problems in computer vision, natural language processing, computational biology, and other areas are often based on large data sets of tens of thousands to millions of training instances. However, several standard learning algorithms, such as *kernel-based* algorithms, e.g., Support Vector Machines, Kernel Ridge Regression, Kernel PCA, do not easily scale to such orders of magnitude. This thesis focuses on sampling-based matrix approximation techniques that help scale kernel-based algorithms to large-scale datasets. We address several fundamental theoretical and empirical questions including:

1. *What approximation should be used?* We discuss two common sampling-based methods, providing novel theoretical insights regarding their suitability for various applications and experimental results motivated by this theory. Our results show that one of these methods, the Nyström method, is superior in the context of large-scale learning.

2. *Do these approximations work in practice?* We show the effectiveness of approximation techniques on a variety of problems. In the largest study

to-date for manifold learning, we use the Nyström method to extract low-dimensional structure from high-dimensional data to effectively cluster face images. We also report good empirical results for Kernel Ridge Regression and Kernel Logistic Regression.

3. *How should we sample columns?* A key aspect of sampling-based algorithms is the distribution according to which columns are sampled. We study both fixed and adaptive sampling schemes as well as a promising ensemble technique that can be easily parallelized and generates superior approximations, both in theory and in practice.

4. *How well do these approximations work in theory?* We provide theoretical analyses of the Nyström method to understand when this technique should be used. We present guarantees on approximation accuracy based on various matrix properties and analyze the effect of matrix approximation on actual kernel-based algorithms.

This work has important consequences for the machine learning community since it extends to large-scale applications the benefits of kernel-based algorithms. The crucial aspect of this research, involving low-rank matrix approximation, is of independent interest within the field of linear algebra.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

Machine Learning can be defined as a set of computational methods that uses *experience* to improve performance and make accurate predictions. In today's data-driven society, this experience often takes the form of large-scale data, e.g., images from the web, sequence data from the human genome, graphs representing friendship networks, time-series data of stock prices, speech corpora of news broadcasts, etc. Hence, modern learning problems in computer vision, natural language processing, computational biology, and other areas are often based on large data sets of tens of thousands to millions of training instances. In this thesis, we ask the fundamental question: *How can machine learning algorithms handle such large-scale data?*

In particular, we focus our attention on kernel-based algorithms (Schölkopf

& Smola, 2002; Shawe-Taylor & Cristianini, 2004). This broad class of learning algorithms has rich theoretical underpinnings and state-of-the-art empirical performance for a variety of problems, e.g., Support Vector Machines (SVMs) and Kernel Logistic Regression (KLR) for classification, Support Vector Regression (SVR) and Kernel Ridge Regression (KRR) for regression, Kernel Principle Component Analysis (KPCA) for non-linear dimensionality reduction, SVM-Rank for ranking, etc. Kernel methods rely solely on similarity measures between pairs of data points, namely inner products. The power of these algorithms stems from the ability to replace the standard inner product with some other kernel function, allowing context-specific information to be incorporated into these algorithms via the choice of kernel function. More specifically, data points can be mapped in a non-linear fashion from their input space into some high-dimensional feature space, and inner products in this feature space can be used to solve a variety of learning problems. Popular kernels include polynomial, Gaussian, sigmoid and sequence kernels. Indeed, the flexibility in choice of kernel is a major benefit of these algorithms, as the kernel function can be chosen arbitrarily so long as it is positive definite symmetric, which means that for any set of $n$ data points, the similarity matrix derived from the kernel function must be symmetric positive semidefinite (see Section 2.1.1 for further discussion).

Despite the favorable properties of kernel methods in terms of theory, empirical performance and flexibility, scalability remains a major drawback. These algorithms require $\mathrm{O}(n^2)$ space to store the kernel matrix. Furthermore, they often require O($n^3$) time, requiring matrix inversion, Singular Value Decomposition (SVD) or quadratic programming in the case of SVMs. For large-scale data sets, both the space and time requirements quickly become intractable. For instance, when working with a dataset of 18M data points (as we will discuss in Section 3.1), storing the entire kernel matrix would require $\sim$1300TB, and even if we could somehow store all of this data, performing O($n^3$) operations would be completely infeasible. Various optimization methods have been introduced to speed up kernel methods, e.g., SMO (Platt, 1999), shrinking (Joachims, 1999), chunking (Boser et al., 1992), parallelized SVMs (Chang et al., 2008) and parallelized KLR (Mann et al., 2009). However for large-scale problems, the storage and processing costs can nonetheless be intractable.

In this thesis, we will focus on an attractive solution to this problem that involves efficiently generating low-rank approximations to the kernel matrix. Low-rank approximation appears in a wide variety of applications including lossy data compression, image processing, text analysis and cryptography, and is at the core of widely used algorithms such as Principle Component Analysis, Multidimensional Scaling and Latent Semantic Indexing. In the context of kernel methods, kernel functions are sometimes chosen such that the resulting kernel matrix is sparse, in which case sparse computation methods can be used. However, in many applications the kernel matrix is dense, but can be well approximated by a low-rank matrix. SVD can be used to find 'optimal' low-rank approximations, as we will formalize in Section 2.1.1. However SVD requires storage of the full kernel matrix and the runtime is superlinear in $n$, and hence does not scale well for large-scale applications. The sampling-based approaches that we discuss attempt to construct low-rank matrices that are nearly 'optimal' while also having linear space and time constraints with respect to $n$.

## 1.2 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 (Johnson & 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 & Loan, 1983; Gorrell, 2006; Rokhlin et al., 2009;

Halko et al., 2009) for generating low-rank approximations. These iterative methods require computation of matrix-vector products at each step and involve multiple passes through the data. Once again, these algorithms are not suitable for large, dense matrices. Matrix sparsification algorithms (Achlioptas & 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 (defined in Section 2.2.2) can be bounded relative to the optimal rank-$k$ approximation of the matrix (Ruston, 1964). Deterministic algorithms such as rank-revealing QR (Gu & 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 leverage scores, i.e., the squared $L_2$ norms of the columns (Frieze et al., 1998; Drineas et al., 2006a; Rudelson & 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). However, as we will discuss Section 2.2.2, the task of matrix projection involves projecting the input matrix onto a low-rank subspace, and for kernel matrices this requires superlinear time and space with respect to $n$.

There does however, exist 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 & Drineas, 2009). In the context of symmetric positive semidefinite matrices, the Nyström method is the most 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 & Schölkopf, 2000; Drineas & Mahoney, 2005; Zhang et al., 2008; Zhang & Kwok, 2009; Kumar et al., 2009c; Kumar et al., 2009b; Kumar et al., 2009a; Belabbas & Wolfe, 2009; Belabbas & Wolfe, 2009; Talwalkar & 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 & Scheinberg, 2002; Bach & Jordan, 2002; Bach & Jordan, 2005), can also be viewed as a specific sampling technique associated with the Nyström method (Bach & Jordan, 2005). As noted by Candès and Recht (2009); Talwalkar and Rostamizadeh (2010), the Nyström approximation is related to the problem of matrix completion (Candès & Recht, 2009; Candès & Tao, 2009), which attempts to complete a low-rank matrix from a random sample of its entries. However, the matrix completion setting assumes that the target matrix is low-rank and only allows for limited access to the data. In contrast, the Nyström method, and sampling-based low-rank approximation algorithms in general, 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 in our access to the original matrix.

## 1.3 Contributions

In this thesis, we provide a unified treatment of sampling-based matrix approximation in the context of machine learning, in part based on work from the following publications: Talwalkar et al. (2008); Kumar et al. (2009c);

Kumar et al. (2009b); Kumar et al. (2009a); Cortes et al. (2010); Talwalkar and Rostamizadeh (2010). We address several fundamental theoretical and empirical questions including:

1. *What approximation should be used?* We discuss two recently introduced sampling-based methods that estimate the SVD of a positive semidefinite matrix from a small subset of its columns. We present a theoretical comparison between the two methods, provide novel insights regarding their suitability for various applications, and include experimental results motivated by this theory. Our results show that one of these methods, the Nyström method, is superior in the context of large-scale kernel-based algorithms on the scale of millions of training instances.

2. *Do these approximations work in practice?* We show the effectiveness of matrix approximation techniques on a variety of problems. We first focus on the task of large-scale manifold learning, which involves extracting low-dimensional structure from high-dimensional data in an unsupervised manner. In this study, the largest such study on manifold learning to-date involving 18M face images, we are able to use the low-dimensional embeddings to more effectively cluster face images. In fact, the techniques we describe are currently used by Google as part of its social networking application (Kumar & Rowley, 2010). We further show the effectiveness of the Nyström method to scale algorithms such as Kernel Ridge Regression and Kernel Logistic Regression.

3. *How should we sample columns?* A key aspect of sampling-based algorithms is the distribution according to which the columns are sampled. We study both fixed and adaptive sampling schemes. In a fixed distribution scheme, the distribution over the columns remains the same throughout the procedure. In contrast, adaptive schemes iteratively update the distribution after each round of sampling. Furthermore, we introduce a promising ensemble technique that can be easily parallelized and generates superior approximations, both in theory and in practice when working with millions of training instances.

4. *How well do these approximations work in theory?* We provide theoretical analyses of the Nyström method to understand when these sampling techniques should be used. We present a variety of guarantees on the approximation accuracy based on various properties of the kernel matrix. In addition to studying the quality of matrix approximation relative to original kernel matrix, we also provide a theoretical analysis of the effect of matrix approximation on actual kernel-based algorithms such as SVMs, SVR, KPCA and KRR, as this is a major application of these sampling techniques.

This work has important consequences for the machine learning community since it extends to large-scale applications the benefits of kernel-based algorithms. The crucial aspect of this research, involving low-rank matrix approximation, is of independent interest within the field of linear algebra.

# Chapter 2

# Low Rank Approximations

## 2.1 Preliminaries

In this chapter, we introduce the two most common sampling-based techniques for matrix approximation and compare their performance on a variety of tasks. The content of this chapter is primarily based on results presented in Kumar et al. (2009b). We begin by introducing notation and basic definitions.

### 2.1.1 Notation

Let $\mathbf{T} \in \mathbb{R}^{a \times b}$ be an arbitrary matrix. We define $\mathbf{T}^{(j)}$, $j = 1 \ldots b$, as the $j$th column vector of $\mathbf{T}$ and $\mathbf{T}_{(i)}$, $i = 1 \ldots a$, 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 $\text{rank}(\mathbf{T}) = r$, 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 $\mathbf{T}_k = \operatorname{argmin}_{\mathbf{V} \in \mathbb{R}^{a \times b}, \operatorname{rank}(\mathbf{V}) = k} \|\mathbf{T} - \mathbf{V}\|_\xi$, 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 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 $\operatorname{rank}(\mathbf{K}) = r \leq n$, i.e. 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 $\mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_r)$ 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 $\mathbf{K}$, i.e., $\mathbf{K}_k = \operatorname{argmin}_{\mathbf{K}' \in \mathbb{R}^{n \times n}, \operatorname{rank}(\mathbf{K}') = k} \|\mathbf{K} - \mathbf{K}'\|_{\xi \in \{2, F\}}$, 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 & 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 we sample columns uniformly without replacement, though various methods have been proposed to select columns, and Chapter 4 is devoted to this crucial aspect of sampling-based algorithms. 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:

$$\mathbf{K} = \begin{bmatrix} \mathbf{W} & \mathbf{K}_{21}^\top \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} \mathbf{W} \\ \mathbf{K}_{21} \end{bmatrix}. \tag{2.1}$$

The approximation techniques discussed next use the SVD of $\mathbf{W}$ and $\mathbf{C}$ to generate approximations for $\mathbf{K}$.

### 2.1.2 Nyström method

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 used in applications ranging from manifold learning to image segmentation (Platt, 2004; Fowlkes et al., 2004; Talwalkar et al., 2008). The Nyström method uses $\mathbf{W}$ and $\mathbf{C}$ from (2.1) to approximate $\mathbf{K}$, and for 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:

$$\widetilde{\mathbf{K}}_k^{nys} = \mathbf{C}\mathbf{W}_k^+\mathbf{C}^\top \approx \mathbf{K}, \tag{2.2}$$

where $\mathbf{W}_k$ is the best $k$-rank approximation of $\mathbf{W}$ for the Frobenius norm and $\mathbf{W}_k^+$ denotes the pseudo-inverse of $\mathbf{W}_k$. If we write the SVD of $\mathbf{W}$ as $\mathbf{W} = \mathbf{U}_W \boldsymbol{\Sigma}_W \mathbf{U}_W^\top$, then plugging into (2.2) we can write

$$\begin{aligned}\widetilde{\mathbf{K}}_k^{nys} &= \mathbf{C}\mathbf{U}_{W,k}\boldsymbol{\Sigma}_{W,k}^+\mathbf{U}_{W,k}^\top\mathbf{C}^\top \\ &= \left(\sqrt{\frac{l}{n}}\mathbf{C}\mathbf{U}_{W,k}\boldsymbol{\Sigma}_{W,k}^+\right)\left(\frac{n}{l}\boldsymbol{\Sigma}_{W,k}\right)\left(\sqrt{\frac{l}{n}}\mathbf{C}\mathbf{U}_{W,k}\boldsymbol{\Sigma}_{W,k}^+\right)^\top,\end{aligned}$$

and hence the Nyström method approximates the top $k$ singular values ($\boldsymbol{\Sigma}_k$) and singular vectors ($\mathbf{U}_k$) of $\mathbf{K}$ as:

$$\widetilde{\boldsymbol{\Sigma}}_{nys} = \left(\frac{n}{l}\right)\boldsymbol{\Sigma}_{W,k} \quad \text{and} \quad \widetilde{\mathbf{U}}_{nys} = \sqrt{\frac{l}{n}}\mathbf{C}\mathbf{U}_{W,k}\boldsymbol{\Sigma}_{W,k}^+. \tag{2.3}$$

Since the running time complexity of SVD on $\mathbf{W}$ is in $O(l^3)$ and matrix multiplication with $\mathbf{C}$ takes $O(kln)$, the total complexity of the Nyström approximation computation is in $O(l^3+kln)$.

### 2.1.3 Column-sampling method

The Column-sampling method was introduced to approximate the SVD of any rectangular matrix (Frieze et al., 1998). It generates approximations of $\mathbf{K}$ by using the SVD of $\mathbf{C}$. If we write the SVD of $\mathbf{C}$ as $\mathbf{C} = \mathbf{U}_C\boldsymbol{\Sigma}_C\mathbf{V}_C^\top$ then the Column-sampling method approximates the top $k$ singular values ($\boldsymbol{\Sigma}_k$) and singular vectors ($\mathbf{U}_k$) of $\mathbf{K}$ as:

$$\widetilde{\mathbf{\Sigma}}_{col} = \sqrt{\frac{n}{l}}\mathbf{\Sigma}_C \quad \text{and} \quad \widetilde{\mathbf{U}}_{col} = \mathbf{U}_C = \mathbf{C}\mathbf{V}_C\mathbf{\Sigma}_C^+. \tag{2.4}$$

The runtime of the Column-sampling method is dominated by the SVD of $\mathbf{C}$. Even when only $k$ singular values and singular vectors are required, the algorithm takes $\mathrm{O}(nl^2)$ time to perform SVD on $C$, and is thus more expensive than the Nyström method. Often, in practice, the SVD of $\mathbf{C}^\top\mathbf{C}$ is performed in $\mathrm{O}(l^3)$ time instead of the running SVD of $\mathbf{C}$. However, this procedure is still more expensive than the Nyström method due to the additional cost of computing $\mathbf{C}^\top\mathbf{C}$ which is in $\mathrm{O}(nl^2)$.

## 2.2 Nyström vs Column-sampling

Given that two sampling-based techniques exist to approximate the SVD of SPSD matrices, we pose a natural question: which method should one use to approximate singular values, singular vectors and low-rank approximations?[1] We first analyze the form of these approximations and then empirically evaluate their performance in Section 2.2.3 on a variety of datasets.

[1] We will address the performance of the Nyström and Column-sampling methods for computing low-dimensional embeddings separately in Section 3.1 when we discuss the problem of manifold learning, since low-dimensional embedding is fundamentally related to the problem of manifold learning.

### 2.2.1 Singular values and singular vectors

As shown in (2.3) and (2.4), the singular values of $\mathbf{K}$ are approximated as the scaled singular values of $\mathbf{W}$ and $\mathbf{C}$, respectively. The scaling terms are quite rudimentary and are primarily meant to *compensate* for the 'small sample size' effect for both approximations. However, the form of singular vectors is more interesting. The Column-sampling singular vectors ($\widetilde{\mathbf{U}}_{col}$) are orthonormal since they are the singular vectors of $\mathbf{C}$. In contrast, the Nyström singular vectors ($\widetilde{\mathbf{U}}_{nys}$) are approximated by *extrapolating* the singular vectors of $\mathbf{W}$ as shown in (2.3), and are *not* orthonormal. It is easy to verify that $\widetilde{\mathbf{U}}_{nys}^{\top}\widetilde{\mathbf{U}}_{nys} \neq \mathbf{I}_l$, where $\mathbf{I}_l$ is the identity matrix of size $l$. As we show in Section 2.2.3, this adversely affects the accuracy of singular vector approximation from the Nyström method.

It is possible to orthonormalize the Nyström singular vectors by using QR decomposition. Since $\widetilde{\mathbf{U}}_{nys} \propto \mathbf{C}\mathbf{U}_W\mathbf{\Sigma}_W^{+}$, where $\mathbf{U}_W$ is orthogonal and $\mathbf{\Sigma}_W$ is diagonal, this simply implies that QR decomposition creates an orthonormal span of $\mathbf{C}$ rotated by $\mathbf{U}_W$. However, the complexity of QR decomposition of $\widetilde{\mathbf{U}}_{nys}$ is the same as that of the SVD of $\mathbf{C}$. Thus, the computational cost of orthogonalizing $\widetilde{\mathbf{U}}_{nys}$ would nullify the computational benefit of the Nyström method over Column-sampling.

### 2.2.2 Low-rank approximation

Several studies have empirically shown that the accuracy of low-rank approximations of kernel matrices is tied to the performance of kernel-based learning algorithms (Williams & Seeger, 2000; Talwalkar et al., 2008; Zhang et al., 2008). Furthermore, we will theoretically show the effect of an approximation in the kernel matrix on the *hypothesis* generated by several widely used kernel-based learning algorithms in Section 5.3. Hence, accurate low-rank approximations are of great practical interest in machine learning. As discussed in Section 2.1.1, the optimal $\mathbf{K}_k$ is given by,

$$\mathbf{K}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{U}_k^\top = \mathbf{U}_k \mathbf{U}_k^\top \mathbf{K} = \mathbf{K} \mathbf{U}_k \mathbf{U}_k^\top \tag{2.5}$$

where the columns of $\mathbf{U}_k$ are the $k$ singular vectors of $\mathbf{K}$ corresponding to the top $k$ singular values of $\mathbf{K}$. We refer to $\mathbf{U}_k \mathbf{\Sigma}_k \mathbf{U}_k^\top$ as *Spectral Reconstruction*, since it uses both the singular values and vectors of $\mathbf{K}$, and $\mathbf{U}_k \mathbf{U}_k^\top \mathbf{K}$ as *Matrix Projection*, since it uses only singular vectors to compute the projection of $\mathbf{K}$ onto the space spanned by vectors $\mathbf{U}_k$. These two low-rank approximations are equal only if $\mathbf{\Sigma}_k$ and $\mathbf{U}_k$ contain the true singular values and singular vectors of $\mathbf{K}$. Since this is not the case for approximate methods such as the Nyström method and Column-sampling, these two measures generally give different errors. Thus, we analyze each measure separately in the following sections.

### Matrix projection

For Column-sampling, using (2.4), the low-rank approximation via matrix projection is

$$\widetilde{\mathbf{K}}_k^{col} = \widetilde{\mathbf{U}}_{col,k}\widetilde{\mathbf{U}}_{col,k}^\top \mathbf{K} = \mathbf{U}_{C,k}\mathbf{U}_{C,k}^\top \mathbf{K} = \mathbf{C}((\mathbf{C}^\top\mathbf{C})_k)^+\mathbf{C}^\top\mathbf{K}, \tag{2.6}$$

where $(\mathbf{C}^\top\mathbf{C})_k = \mathbf{V}_{C,k}(\mathbf{\Sigma}_{C,k}^2)^+\mathbf{V}_{C,k}^\top$. Clearly, if $k = l$, $(\mathbf{C}^\top\mathbf{C})_k = \mathbf{C}^\top\mathbf{C}$. Similarly, using (2.3), the Nyström matrix projection is

$$\widetilde{\mathbf{K}}_k^{nys} = \widetilde{\mathbf{U}}_{nys,k}\widetilde{\mathbf{U}}_{nys,k}^\top \mathbf{K} = \frac{l}{n}\mathbf{C}(\mathbf{W}_k^2)^+\mathbf{C}^\top\mathbf{K}, \tag{2.7}$$

where $\mathbf{W}_k = \mathbf{W}$ if $k = l$.

As shown in (2.6) and (2.7), the two methods have similar expressions for matrix projection, except that $\mathbf{C}^\top\mathbf{C}$ is replaced by a scaled $\mathbf{W}^2$. Furthermore, the scaling term appears only in the expression for the Nyström method. We now present Theorem 2.1 and Observations 2.1 and 2.2, which provide further insights about these two methods in the context of matrix projection.

**Theorem 2.1** *The Column-sampling and Nyström matrix projections are of the form* $\mathbf{U}_C\mathbf{R}\mathbf{U}_C^\top\mathbf{K}$*, where* $\mathbf{R} \in \mathbb{R}^{l\times l}$ *is SPSD. Further, Column-sampling gives the lowest reconstruction error (measured in* $\|\cdot\|_F$*) among all such approximations if* $k = l$*.*

*Proof.* From (2.6), it is easy to see that

$$\widetilde{\mathbf{K}}_k^{col} = \mathbf{U}_{C,k}\mathbf{U}_{C,k}^\top\mathbf{K} = \mathbf{U}_C\mathbf{R}_{col}\mathbf{U}_C^\top\mathbf{K}, \tag{2.8}$$

where $\mathbf{R}_{col} = \left[\begin{smallmatrix}\mathbf{I}_k & 0\\ 0 & 0\end{smallmatrix}\right]$. Similarly, from (2.7) we can derive

$$\widetilde{\mathbf{K}}_k^{nys} = \mathbf{U}_C\mathbf{R}_{nys}\mathbf{U}_C^\top\mathbf{K} \;\text{ where } \mathbf{R}_{nys} = \mathbf{Y}(\mathbf{\Sigma}_{W,k}^2)^+\mathbf{Y}^\top, \tag{2.9}$$

and $\mathbf{Y} = \sqrt{l/n}\mathbf{\Sigma}_C\mathbf{V}_C^\top\mathbf{U}_{W,k}$. Note that both $\mathbf{R}_{col}$ and $\mathbf{R}_{nys}$ are SPSD matrices. Furthermore, if $k = l$, $\mathbf{R}_{col} = \mathbf{I}_l$. Let $\mathbf{E}$ be the (squared) reconstruction error for an approximation of the form $\mathbf{U}_C\mathbf{R}\mathbf{U}_C^\top\mathbf{K}$, where $\mathbf{R}$ is an arbitrary SPSD matrix. Hence, when $k = l$, the difference in reconstruction error between the generic and the Column-sampling approximations is

$$\begin{aligned}
\mathbf{E} - \mathbf{E}_{col} &= \|\mathbf{K} - \mathbf{U}_C\mathbf{R}\mathbf{U}_C^\top\mathbf{K}\|_F^2 - \|\mathbf{K} - \mathbf{U}_C\mathbf{U}_C^\top\mathbf{K}\|_F^2\\
&= \mathrm{Tr}\left[\mathbf{K}^\top(\mathbf{I}_n - \mathbf{U}_C\mathbf{R}\mathbf{U}_C^\top)^\top(\mathbf{I}_n - \mathbf{U}_C\mathbf{R}\mathbf{U}_C^\top)\mathbf{K}\right]\\
&\quad - \mathrm{Tr}\left[\mathbf{K}^\top(\mathbf{I}_n - \mathbf{U}_C\mathbf{U}_C^\top)^\top(\mathbf{I}_n - \mathbf{U}_C\mathbf{U}_C^\top)\mathbf{K}\right]\\
&= \mathrm{Tr}\left[\mathbf{K}^\top(\mathbf{U}_C\mathbf{R}^2\mathbf{U}_C^\top - 2\mathbf{U}_C\mathbf{R}\mathbf{U}_C^\top + \mathbf{U}_C\mathbf{U}_C^\top)\mathbf{K}\right]\\
&= \mathrm{Tr}\left[((\mathbf{R} - \mathbf{I}_n)\mathbf{U}_C^\top\mathbf{K})^\top((\mathbf{R} - \mathbf{I}_n)\mathbf{U}_C^\top\mathbf{K})\right]\\
&\geq 0.
\end{aligned} \tag{2.10}$$

We used the facts that $\mathbf{U}_C^\top\mathbf{U}_C = \mathbf{I}_n$ and $\mathbf{A}^\top\mathbf{A}$ is SPSD for any matrix $\mathbf{A}$. □

**Observation 2.1** *For $k = l$, matrix projection for Column-sampling recon-*

*structs* $\mathbf{C}$ *exactly. This can be seen by block-decomposing* $\mathbf{K}$ *as:* $\mathbf{K} = [\mathbf{C} \quad \bar{\mathbf{C}}]$*, where* $\bar{\mathbf{C}} = [\mathbf{K}_{21} \quad \mathbf{K}_{22}]^\top$*, and using (2.6):*

$$\widetilde{\mathbf{K}}_l^{col} = \mathbf{C}(\mathbf{C}^\top\mathbf{C})^+\mathbf{C}^\top\mathbf{K} = [\mathbf{C} \;\; \mathbf{C}(\mathbf{C}^\top\mathbf{C})^+\mathbf{C}^\top\bar{\mathbf{C}}] = [\mathbf{C} \quad \bar{\mathbf{C}}]. \tag{2.11}$$

**Observation 2.2** *For* $k = l$*, the span of the orthogonalized Nyström singular vectors equals the span of* $\widetilde{\mathbf{U}}_{col}$*, as discussed in Section 2.2.1. Hence, matrix projection is identical for Column-sampling and Orthonormal Nyström for* $k = l$*.*

From an application point of view, matrix projection approximations tend to be more accurate than the spectral reconstruction approximations discussed in the next section. However, these low-rank approximations are not necessarily symmetric and require storage of and multiplication with $\mathbf{K}$. Hence, although matrix projection is often analyzed theoretically, for large-scale problems, the storage and computational requirements may be inefficient or even infeasible.

### Spectral reconstruction

Using (2.3), the Nyström spectral reconstruction is:

$$\widetilde{\mathbf{K}}_k^{nys} = \widetilde{\mathbf{U}}_{nys,k}\widetilde{\mathbf{\Sigma}}_{nys,k}\widetilde{\mathbf{U}}_{nys,k}^\top = \mathbf{C}\mathbf{W}_k^+\mathbf{C}^\top. \tag{2.12}$$

When $k = l$, this approximation perfectly reconstructs three blocks of $\mathbf{K}$, and $\mathbf{K}_{22}$ is approximated by the Schur Complement of $\mathbf{W}$ in $\mathbf{K}$:

$$\widetilde{\mathbf{K}}_l^{nys} = \mathbf{C}\mathbf{W}^+\mathbf{C}^\top = \begin{bmatrix} \mathbf{W} & \mathbf{K}_{21}^\top \\ \mathbf{K}_{21} & \mathbf{K}_{21}\mathbf{W}^+\mathbf{K}_{21} \end{bmatrix}. \tag{2.13}$$

The Column-sampling spectral reconstruction has a similar form as (2.12):

$$\widetilde{\mathbf{K}}_k^{col} = \widetilde{\mathbf{U}}_{col,k}\widetilde{\mathbf{\Sigma}}_{col,k}\widetilde{\mathbf{U}}_{col,k}^\top = \sqrt{n/l}\,\mathbf{C}\big((\mathbf{C}^\top\mathbf{C})_k^{\frac{1}{2}}\big)^+\mathbf{C}^\top. \tag{2.14}$$

In contrast with matrix projection, the scaling term now appears in the Column-sampling reconstruction. To analyze the two approximations, we consider an alternative characterization using the fact that $\mathbf{K} = \mathbf{X}^\top\mathbf{X}$ for some $\mathbf{X} \in \mathbb{R}^{N\times n}$. Similar to Drineas and Mahoney (2005), we define a zero-one sampling matrix, $\mathbf{S} \in \mathbb{R}^{n\times l}$, that selects $l$ columns from $\mathbf{K}$, i.e., $\mathbf{C} = \mathbf{K}\mathbf{S}$. Each column of $\mathbf{S}$ has exactly one non-zero entry per column. Further, $\mathbf{W} = \mathbf{S}^\top\mathbf{K}\mathbf{S} = (\mathbf{X}\mathbf{S})^\top\mathbf{X}\mathbf{S} = \mathbf{X}'^\top\mathbf{X}'$, where $\mathbf{X}' \in \mathbb{R}^{N\times l}$ contains $l$ sampled columns of $\mathbf{X}$ and $\mathbf{X}' = \mathbf{U}_{X'}\mathbf{\Sigma}_{X'}\mathbf{V}_{X'}^\top$ is the SVD of $\mathbf{X}'$. We use these definitions to present Theorems 2.2 and 2.3.

**Theorem 2.2** *Column-sampling and Nyström spectral reconstructions of rank $k$ are of the form $\mathbf{X}^\top\mathbf{U}_{X',k}\mathbf{Z}\mathbf{U}_{X',k}^\top\mathbf{X}$, where $\mathbf{Z} \in \mathbb{R}^{k\times k}$ is SPSD. Further, among all approximations of this form, neither the Column-sampling nor the Nyström approximation is optimal (in $\|\cdot\|_F$).*

*Proof.* If $\alpha = \sqrt{n/l}$, then starting from (2.14) and expressing $\mathbf{C}$ and $\mathbf{W}$ in terms of $\mathbf{X}$ and $\mathbf{S}$, we have

$$\begin{aligned}\widetilde{\mathbf{K}}_k^{col} =& \alpha \mathbf{K}\mathbf{S}((\mathbf{S}^\top \mathbf{K}^2 \mathbf{S})_k^{1/2})^{+} \mathbf{S}^\top \mathbf{K}^\top \\ =& \alpha \mathbf{X}^\top \mathbf{X}' \big((\mathbf{V}_{C,k} \mathbf{\Sigma}_{C,k}^2 \mathbf{V}_{C,k}^\top)^{1/2}\big)^{+} \mathbf{X}'^\top \mathbf{X} \\ =& \mathbf{X}^\top \mathbf{U}_{X',k} \mathbf{Z}_{col} \mathbf{U}_{X',k}^\top \mathbf{X}, \end{aligned} \tag{2.15}$$

where $\mathbf{Z}_{col} = \alpha \mathbf{\Sigma}_{X'} \mathbf{V}_{X'}^\top \mathbf{V}_{C,k} \mathbf{\Sigma}_{C,k}^{+} \mathbf{V}_{C,k}^\top \mathbf{V}_{X'} \mathbf{\Sigma}_{X'}$. Similarly, from (2.12) we have:

$$\begin{aligned}\widetilde{\mathbf{K}}_k^{nys} =& \mathbf{K}\mathbf{S}(\mathbf{S}^\top \mathbf{K}\mathbf{S})_k^{+} \mathbf{S}^\top \mathbf{K}^\top \\ =& \mathbf{X}^\top \mathbf{X}' \big(\mathbf{X}'^\top \mathbf{X}'\big)_k^{+} \mathbf{X}'^\top \mathbf{X} \\ =& \mathbf{X}^\top \mathbf{U}_{X',k} \mathbf{U}_{X',k}^\top \mathbf{X}. \end{aligned} \tag{2.16}$$

Clearly, $\mathbf{Z}_{nys} = \mathbf{I}_k$. Next, we analyze the error, $\mathbf{E}$, for an arbitrary $\mathbf{Z}$, which yields the approximation $\widetilde{\mathbf{K}}_k^{Z}$:

$$\mathbf{E} = \|\mathbf{K} - \widetilde{\mathbf{K}}_k^{Z}\|_F^2 = \|\mathbf{X}^\top (\mathbf{I}_N - \mathbf{U}_{X',k} \mathbf{Z} \mathbf{U}_{X',k}^\top) \mathbf{X}\|_F^2. \tag{2.17}$$

Let $\mathbf{X} = \mathbf{U}_X \mathbf{\Sigma}_X \mathbf{V}_X^\top$ and $\mathbf{Y} = \mathbf{U}_X^\top \mathbf{U}_{X',k}$. Then,

$$\begin{aligned}
\mathbf{E} &= \mathrm{Tr}\left[\left((\mathbf{I}_N - \mathbf{U}_{X',k}\mathbf{Z}\mathbf{U}_{X',k}^\top)\mathbf{U}_X\mathbf{\Sigma}_X^2\mathbf{U}_X^\top\right)^2\right] \\
&= \mathrm{Tr}\left[\left(\mathbf{U}_X\mathbf{\Sigma}_X\mathbf{U}_X^\top(\mathbf{I}_N - \mathbf{U}_{X',k}\mathbf{Z}\mathbf{U}_{X',k}^\top)\mathbf{U}_X\mathbf{\Sigma}_X\mathbf{U}_X^\top\right)^2\right] \\
&= \mathrm{Tr}\left[\left(\mathbf{U}_X\mathbf{\Sigma}_X(\mathbf{I}_N - \mathbf{Y}\mathbf{Z}\mathbf{Y}^\top)\mathbf{\Sigma}_X\mathbf{U}_X^\top\right)^2\right] \\
&= \mathrm{Tr}\left[\mathbf{\Sigma}_X(\mathbf{I}_N - \mathbf{Y}\mathbf{Z}\mathbf{Y}^\top)\mathbf{\Sigma}_X^2(\mathbf{I}_N - \mathbf{Y}\mathbf{Z}\mathbf{Y}^\top)\mathbf{\Sigma}_X)\right] \\
&= \mathrm{Tr}\left[\mathbf{\Sigma}_X^4 - 2\mathbf{\Sigma}_X^2\mathbf{Y}\mathbf{Z}\mathbf{Y}^\top\mathbf{\Sigma}_X^2 + \mathbf{\Sigma}_X\mathbf{Y}\mathbf{Z}\mathbf{Y}^\top\mathbf{\Sigma}_X^2\mathbf{Y}\mathbf{Z}\mathbf{Y}^\top\mathbf{\Sigma}_X)\right]. \qquad (2.18)
\end{aligned}$$

To find $\mathbf{Z}^*$, the $\mathbf{Z}$ that minimizes (2.18), we use the convexity of (2.18) and set:

$$\partial\mathbf{E}/\partial\mathbf{Z} = -2\mathbf{Y}^\top\mathbf{\Sigma}_X^4\mathbf{Y} + 2(\mathbf{Y}^\top\mathbf{\Sigma}_X^2\mathbf{Y})\mathbf{Z}^*(\mathbf{Y}^\top\mathbf{\Sigma}_X^2\mathbf{Y}) = 0$$

and solve for $\mathbf{Z}^*$, which gives us:

$$\mathbf{Z}^* = (\mathbf{Y}^\top\mathbf{\Sigma}_X^2\mathbf{Y})^+(\mathbf{Y}^\top\mathbf{\Sigma}_X^4\mathbf{Y})(\mathbf{Y}^\top\mathbf{\Sigma}_X^2\mathbf{Y})^+.$$

$\mathbf{Z}^* = \mathbf{Z}_{nys} = \mathbf{I}_k$ if $\mathbf{Y} = \mathbf{I}_k$, though $\mathbf{Z}^*$ does not in general equal either $\mathbf{Z}_{col}$ or $\mathbf{Z}_{nys}$, which is clear by comparing the expressions of these three matrices, and also by example (see results for 'DEXT' dataset in Figure 2.3(a)). Furthermore, since $\mathbf{\Sigma}_X^2 = \mathbf{\Sigma}_K$, $\mathbf{Z}^*$ depends on the spectrum of $\mathbf{K}$. □

While Theorem 2.2 shows that the optimal approximation is data-dependent and may differ from the Nyström and Column-sampling approximations, Theorem 2.3 reveals that in certain instances the Nyström method is optimal. In contrast, the Column-sampling method enjoys no such guarantee.

**Theorem 2.3** *Suppose* $r = \text{rank}(\mathbf{K}) \leq k \leq l$ *and* $\text{rank}(\mathbf{W}) = r$. *Then, the Nyström approximation is exact for spectral reconstruction. In contrast, Column-sampling is exact iff* $\mathbf{W} = \big((l/n)\mathbf{C}^\top\mathbf{C}\big)^{1/2}$. *When this very specific condition holds, Column-Sampling trivially reduces to the Nyström method.*

*Proof.* Since $\mathbf{K} = \mathbf{X}^\top\mathbf{X}$, $\text{rank}(\mathbf{K}) = \text{rank}(\mathbf{X}) = r$. Similarly, $\mathbf{W} = \mathbf{X}'^\top\mathbf{X}'$ implies $\text{rank}(\mathbf{X}') = r$. Thus the columns of $\mathbf{X}'$ span the columns of $\mathbf{X}$ and $\mathbf{U}_{X',r}$ is an orthonormal basis for $\mathbf{X}$, i.e., $\mathbf{I}_N - \mathbf{U}_{X',r}\mathbf{U}_{X',r}^\top \in \text{Null}(\mathbf{X})$. Since $k \geq r$, from (2.16) we have

$$\|\mathbf{K} - \widetilde{\mathbf{K}}_k^{nys}\|_F = \|\mathbf{X}^\top(\mathbf{I}_N - \mathbf{U}_{X',r}\mathbf{U}_{X',r}^\top)\mathbf{X}\|_F = 0. \tag{2.19}$$

To prove the second part of the theorem, we note that $\text{rank}(C) = r$. Thus, $\mathbf{C} = \mathbf{U}_{C,r}\mathbf{\Sigma}_{C,r}\mathbf{V}_{C,r}^\top$ and $(\mathbf{C}^\top\mathbf{C})_k^{1/2} = (\mathbf{C}^\top\mathbf{C})^{1/2} = \mathbf{V}_{C,r}\mathbf{\Sigma}_{C,r}\mathbf{V}_{C,r}^\top$ since $k \geq r$. If $\mathbf{W} = (1/\alpha)(\mathbf{C}^\top\mathbf{C})^{1/2}$, then the Column-sampling and Nyström approximations are identical and hence exact. Conversely, to exactly reconstruct $\mathbf{K}$, Column-sampling necessarily reconstructs $\mathbf{C}$ exactly. Using $\mathbf{C}^\top = [\mathbf{W} \quad \mathbf{K}_{21}^\top]$ in (2.14)

Table 2.1: Description of the datasets used in our experiments comparing sampling-based matrix approximations (Sim et al., 2002; LeCun & Cortes, 1998; Talwalkar et al., 2008). '$n$' denotes the number of points and '$d$' denotes the number of features in input space.

| Dataset | Data | $n$ | $d$ | Kernel |
|---|---|---|---|---|
| PIE-2.7K | faces | 2731 | 2304 | linear |
| PIE-7K | faces | 7412 | 2304 | linear |
| MNIST | digits | 4000 | 784 | linear |
| ESS | proteins | 4728 | 16 | RBF |
| ABN | abalones | 4177 | 8 | RBF |

we have:

$$\widetilde{\mathbf{K}}_k^{col} = \mathbf{K} \quad \Longrightarrow \quad \alpha\mathbf{C}\big((\mathbf{C}^\top\mathbf{C})_k^{\frac{1}{2}}\big)^{+}\mathbf{W} = \mathbf{C} \tag{2.20}$$

$$\Longrightarrow \quad \alpha\mathbf{U}_{C,r}\mathbf{V}_{C,r}^\top\mathbf{W} = \mathbf{U}_{C,r}\mathbf{\Sigma}_{C,r}\mathbf{V}_{C,r}^\top \tag{2.21}$$

$$\Longrightarrow \quad \alpha\mathbf{V}_{C,r}\mathbf{V}_{C,r}^\top\mathbf{W} = \mathbf{V}_{C,r}\mathbf{\Sigma}_{C,r}\mathbf{V}_{C,r}^\top \tag{2.22}$$

$$\Longrightarrow \quad \mathbf{W} = \frac{1}{\alpha}(\mathbf{C}^\top\mathbf{C})^{1/2}. \tag{2.23}$$

In (2.22) we use $\mathbf{U}_{C,r}^\top\mathbf{U}_{C,r} = \mathbf{I}_r$, while (2.23) follows since $\mathbf{V}_{C,r}\mathbf{V}_{C,r}^\top$ is an orthogonal projection onto the span of the rows of $\mathbf{C}$ and the columns of $\mathbf{W}$ lie within this span implying $\mathbf{V}_{C,r}\mathbf{V}_{C,r}^\top\mathbf{W} = \mathbf{W}$. □

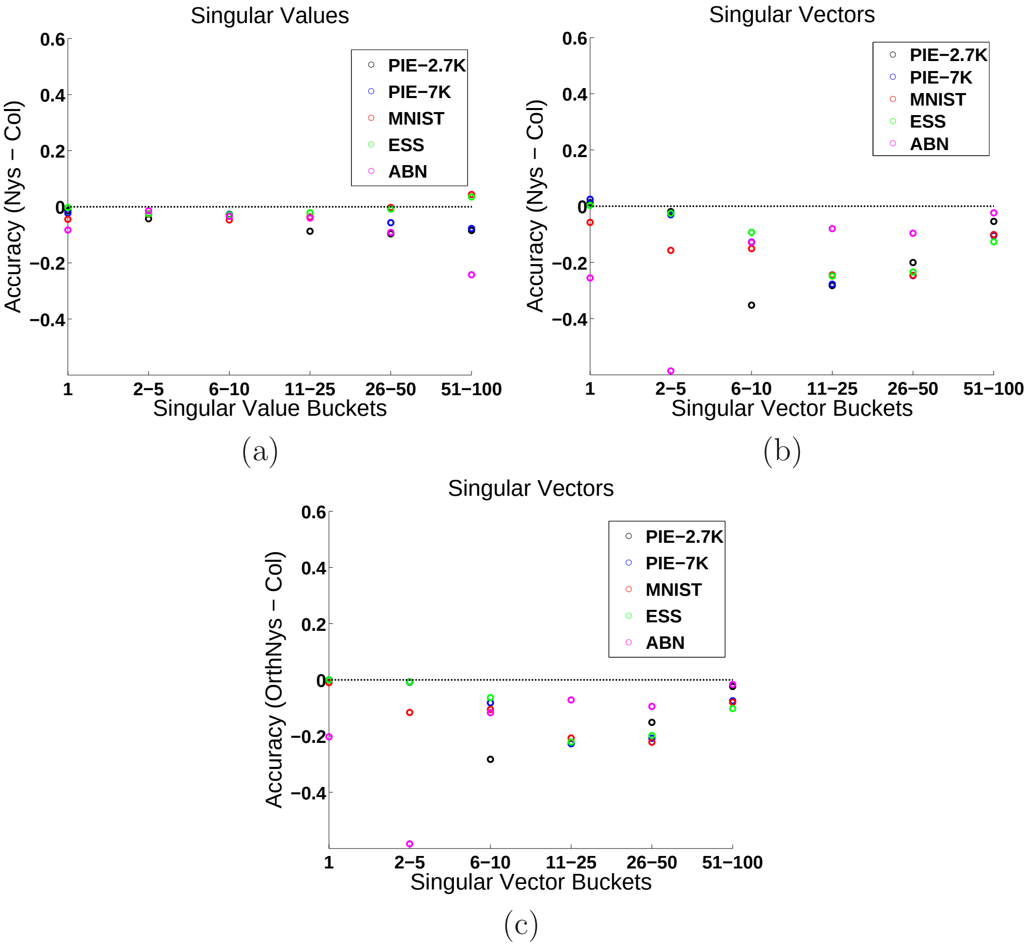
### 2.2.3 Empirical comparison

To test the accuracy of singular values/vectors and low-rank approximations for different methods, we used several kernel matrices arising in different applications, as described in Table 2.1. We worked with datasets containing less than ten thousand points to be able to compare with exact SVD. We fixed $k$ to be 100 in all the experiments, which captures more than 90% of the spectral energy for each dataset.

For singular values, we measured percentage accuracy of the approximate singular values with respect to the exact ones. For a fixed $l$, we performed 10 trials by selecting columns uniformly at random from $\mathbf{K}$. We show in Figure 2.1(a) the difference in mean percentage accuracy for the two methods for $l = n/10$, with results bucketed by groups of singular values. The empirical results show that the Column-sampling method generates more accurate singular values than the Nyström method. A similar trend was observed for other values of $l$.

For singular vectors, the accuracy was measured by the dot product i.e., cosine of principal angles between the exact and the approximate singular vectors. Figure 2.1(b) shows the difference in mean accuracy between Nyström and Column-sampling methods bucketed by groups of singular vectors. The top 100 singular vectors were all better approximated by Column-sampling for all datasets. This trend was observed for other values of $l$ as well. Furthermore, even when the Nyström singular vectors are orthogonalized, the Column-sampling approximations are superior, as shown in Figure 2.1(c).

Next we compared the low-rank approximations generated by the two methods using matrix projection and spectral reconstruction as described in Section 2.2.2 and Section 2.2.2, respectively. We measured the accuracy of



(a) (b) (c)

Figure 2.1: Differences in accuracy between Nyström and Column-Sampling. Values above zero indicate better performance of Nyström and vice-versa. (a) Top 100 singular values with $l = n/10$. (b) Top 100 singular vectors with $l = n/10$. (c) Comparison using orthogonalized Nyström singular vectors.

reconstruction relative to the optimal rank-$k$ approximation, $\mathbf{K}_k$, as:

$$\text{relative accuracy} = \frac{\|\mathbf{K} - \mathbf{K}_k\|_F}{\|\mathbf{K} - \widetilde{\mathbf{K}}_k^{nys/col}\|_F}. \tag{2.24}$$

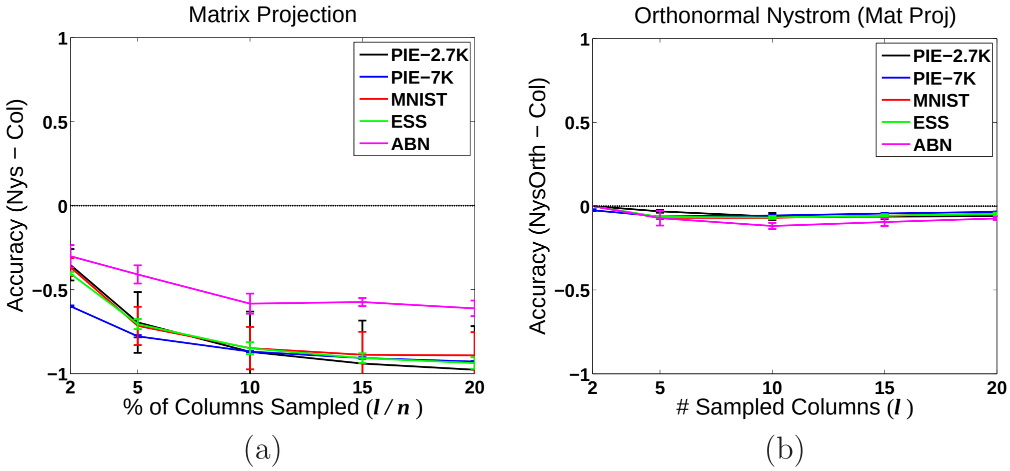


Figure 2.2: Performance accuracy of various matrix projection approximations with $k = 100$. Values below zero indicate better performance of the Column-sampling method. (a) Nyström versus Column-sampling. (b) Orthonormal Nyström versus Column-sampling.

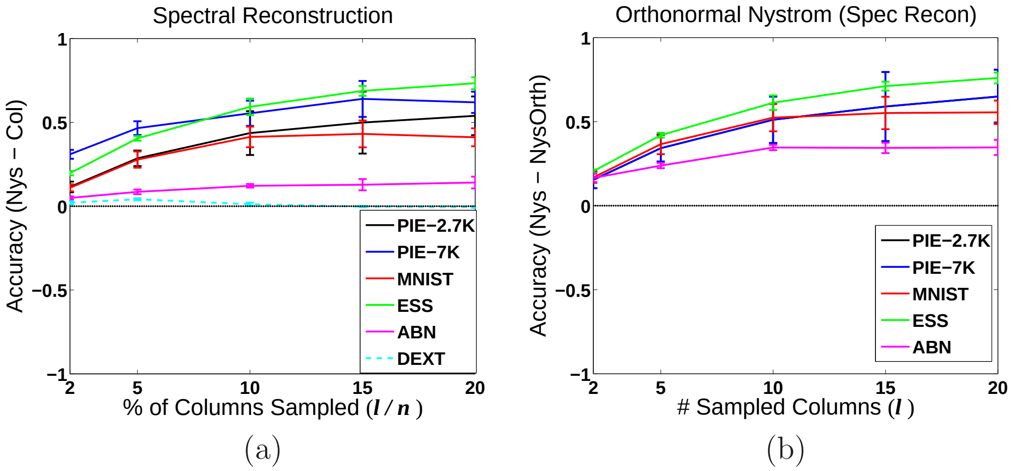
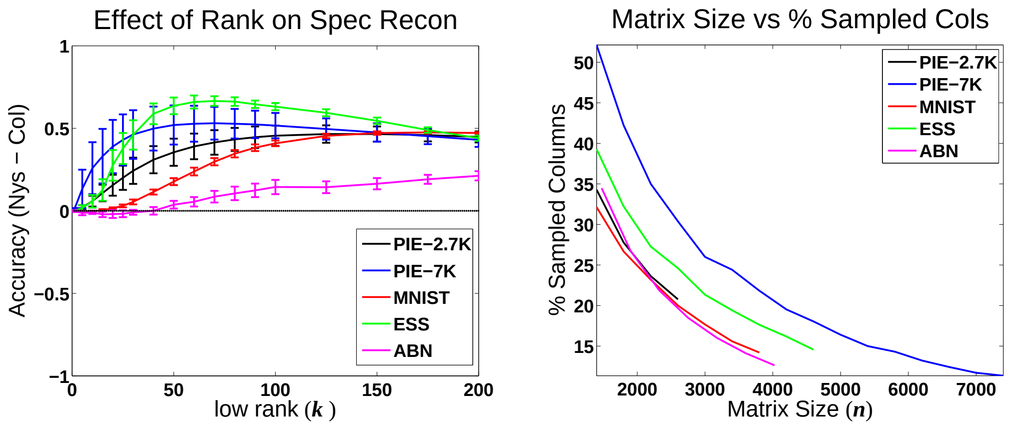


Figure 2.3: Performance accuracy of spectral reconstruction approximations for different methods with $k = 100$. Values above zero indicate better performance of the Nyström method. (a) Nyström versus Column-sampling. (b) Nyström versus Orthonormal Nyström.

The relative accuracy will approach one for good approximations. Results are shown in Figures 2.2(a) and 2.3(a). As motivated by Theorem 2.1 and in agreement with the superior performance of Column-sampling in approximating singular values and vectors, Column-sampling generates better reconstructions via matrix projection. This was observed not only for $l = k$ but also for other values of $l$. In contrast, the Nyström method produces superior results for spectral reconstruction. These results are somewhat surprising given the relatively poor quality of the singular values/vectors for the Nyström method, but they are in agreement with the consequences of Theorem 2.3. We also note that for both reconstruction measures, the methods that exactly reconstruct subsets of the original matrix when $k = l$ (see (2.11) and (2.13)) generate better approximations. Interestingly, these are also the two methods that do not contain scaling terms (see (2.6) and (2.12)).

Further, as stated in Theorem 2.2, the optimal spectral reconstruction approximation is tied to the spectrum of $\mathbf{K}$. Our results suggest that the relative accuracies of Nyström and Column-sampling spectral reconstructions are also tied to this spectrum. When we analyzed spectral reconstruction performance on a sparse kernel matrix with a slowly decaying spectrum, we found that Nyström and Column-sampling approximations were roughly equivalent ('DEXT' in Figure 2.3(a)). This result contrasts the results for dense kernel matrices with exponentially decaying spectra arising from the other datasets used in the experiments.

One factor that impacts the accuracy of the Nyström method for some



(a) (b)

Figure 2.4: Properties of spectral reconstruction approximations. (a) Difference in spectral reconstruction accuracy between Nyström and Column-sampling for various $k$ and fixed $l$. Values above zero indicate better performance of Nyström method. (a) Percentage of columns ($l/n$) needed to achieve 75% relative accuracy for Nyström spectral reconstruction as a function of $n$.

tasks is the non-orthonormality of its singular vectors (Section 2.2.1). When orthonormalized, the Nyström matrix projection error is reduced considerably as shown in Figure 2.2(b). Further, as discussed in Observation 2.2 Orthonormal Nyström is identical to Column-sampling when $k = l$. However, since orthonormalization is computationally costly, it is avoided in practice. Moreover, the accuracy of Orthonormal Nyström spectral reconstruction is actually worse relative to the standard Nyström approximation, as shown in Figure 2.3(b). This surprising result can be attributed to the fact that orthonormalization of the singular vectors leads to the loss of some of the unique properties described in Section 2.2.2. For instance, Theorem 2.3 no longer holds and the scaling terms do not cancel out, i.e., $\widetilde{\mathbf{K}}_k^{nys} \neq \mathbf{C}\mathbf{W}_k^{+}\mathbf{C}^{\top}$.

Even though matrix projection tends to produce more accurate approximations, spectral reconstruction is of great practical interest for large-scale problems since, unlike matrix projection, it does not use all entries in $\mathbf{K}$ to produce a low-rank approximation. Thus, we further expand upon the results from Figure 2.3. We first tested the accuracy of spectral reconstruction for the two methods for varying values of $k$ and a fixed $l$. We found that the Nyström method outperforms Column-sampling across all tested values of $k$, as shown in Figure 2.4(a). Next, we addressed another basic issue: how many columns do we need to obtain reasonable reconstruction accuracy? For very large matrices ($n \approx 10^6$), one would wish to select only a small fraction of the samples. Hence, we performed an experiment in which we fixed $k$ and varied the size of our dataset ($n$). For each $n$, we performed grid search over $l$ to find the minimal $l$ for which the relative accuracy of Nyström spectral reconstruction was at least 75%. Figure 2.4(a) shows that the required percentage of columns ($l/n$) decreases quickly as $n$ increases, lending support to the use of sampling-based algorithms for large-scale data.

## 2.3 Summary

We presented an analysis of two sampling-based techniques for approximating SVD on large dense SPSD matrices, and provided a theoretical and empirical comparison. Although the Column-sampling method generates more accurate singular values/vectors and low-rank matrix projections, the Nyström method constructs better low-rank spectral approximations, which are of great practical interest as they do not use the full matrix.

# Chapter 3

# Applications

In the previous chapter, we discussed two sampling-based techniques that generate approximations for kernel matrices. Although we analyzed the effectiveness of these techniques for approximating singular values, singular vectors and low-rank matrix reconstruction, we have yet to discuss the effectiveness of these techniques in the context of actual machine learning tasks. In fact, the Nyström method has been shown to be successful on a variety of learning tasks including Support Vector Machines (Fine & Scheinberg, 2002), Gaussian Processes (Williams & Seeger, 2000), Spectral Clustering (Fowlkes et al., 2004), manifold learning (Talwalkar et al., 2008), Kernel Logistic Regression (Karsmakers et al., 2007), Kernel Ridge Regression (Cortes et al., 2010) and more generally to approximate regularized matrix inverses via the Woodbury approximation (Williams & Seeger, 2000). In this chapter, we will discuss in detail two specific applications of these approximations, particularly in the context of large-scale applications. First, we will discuss how approximate embeddings can be used in the context of manifold learning, as initially presented in Talwalkar et al. (2008). We will next show the connection between approximate spectral reconstruction and the Woodbury approximation, and will present associated experimental results for Kernel Logistic Regression and Kernel Ridge Regression.

## 3.1 Large-scale Manifold Learning

The problem of dimensionality reduction arises in many computer vision applications, where it is natural to represent images as vectors in a high-dimensional space. Manifold learning techniques extract low-dimensional structure from high-dimensional data in an unsupervised manner. These techniques typically try to unfold the underlying manifold so that some quantity, e.g., pairwise geodesic distances, is maintained invariant in the new space. This makes certain applications such as $K$-means clustering more effective in the transformed space.

In contrast to linear dimensionality reduction techniques such as Principal Component Analysis (PCA), manifold learning methods provide more powerful non-linear dimensionality reduction by preserving the local structure of the input data. Instead of assuming global linearity, these methods typically make a weaker local-linearity assumption, i.e., for nearby points in high-dimensional input space, $l_2$ distance is assumed to be a good measure of geodesic distance, or distance along the manifold. Good sampling of the underlying manifold is essential for this assumption to hold. In fact, many manifold learning techniques provide guarantees that the accuracy of the recovered manifold increases as the number of data samples increases. In the limit of infinite samples, one can recover the true underlying manifold for certain classes of manifolds (Tenenbaum et al., 2000; Belkin & Niyogi, 2006; Donoho & Grimes, 2003). However, there is a trade-off between improved sampling of the manifold and the computational cost of manifold learning algorithms. This paper addresses the computational challenges involved in learning manifolds given millions of face images extracted from the Web.

Several manifold learning techniques have recently been proposed, e.g., Semidefinite Embedding (SDE) (Weinberger & Saul, 2006), Isomap (Tenenbaum et al., 2000), Laplacian Eigenmaps (Belkin & Niyogi, 2001), and Local Linear Embedding (LLE) (Roweis & Saul, 2000). SDE aims to preserve distances and angles between all neighboring points. It is formulated as an instance of semidefinite programming, and is thus prohibitively expensive for large-scale problems. Isomap constructs a dense matrix of approximate geodesic distances between *all* pairs of inputs, and aims to find a low dimensional space that best preserves these distances. Other algorithms, e.g., Laplacian Eigenmaps and LLE, focus only on preserving local neighborhood relationships in the input space. They generate low-dimensional representations via manipulation of the graph Laplacian or other sparse matrices related to the graph Laplacian (Chapelle et al., 2006). In this work, we focus mainly on Isomap and Laplacian Eigenmaps, as both methods have good theoretical properties and the differences in their approaches allow us to make interesting comparisons between dense and sparse methods.

All of the manifold learning methods described above can be viewed as specific instances of Kernel PCA (Ham et al., 2004). These kernel-based algorithms require SVD of matrices of size $n \times n$, where $n$ is the number of samples. This generally takes $\mathrm{O}(n^3)$ time. When only a few singular values and singular vectors are required, there exist less computationally intensive techniques such as Jacobi, Arnoldi, Hebbian and more recent randomized methods (Golub & Loan, 1983; Gorrell, 2006; Rokhlin et al., 2009). These iterative methods require computation of matrix-vector products at each step and involve multiple passes through the data. When the matrix is sparse, these techniques can be implemented relatively efficiently. However, when dealing with a large, dense matrix, as in the case of Isomap, these products become expensive to compute. Moreover, when working with 18M data points, it is not possible even to store the full matrix (∼1300TB), rendering the iterative methods infeasible. Random sampling techniques provide a powerful alternative for approximate SVD and only operate on a subset of the matrix. In this section, we work with both the Nyström and Column-sampling methods described in Section 2, providing the first direct comparison between their performances on practical applications.

Apart from SVD, the other main computational hurdle associated with Isomap and Laplacian Eigenmaps is large-scale graph construction and manipulation. These algorithms first need to construct a local neighborhood graph in the input space, which is an $\mathrm{O}(n^2)$ problem. Moreover, Isomap requires shortest paths between every pair of points resulting in $\mathrm{O}(n^2 \log n)$ computation. Both steps are intractable when $n$ is as large as 18M. In this work, we use approximate nearest neighbor methods, and show that random sampling based singular value decomposition requires the computation of shortest paths only for a subset of points. Furthermore, these approximations allow for an efficient distributed implementation of the algorithms.

We now summarize our main contributions of this section. First, we present the largest scale study so far on manifold learning, using 18M data points. To date, the largest manifold learning study involves the analysis of music data using 267K points (Platt, 2004). In computer vision, the largest study is limited to less than 10K images (He et al., 2005). Our work is thus the largest scale study on face manifolds by a large margin, and is two orders of magnitude larger than any other manifold learning study. Second, we show connections between two random sampling based spectral decomposition algorithms and provide the first direct comparison of the performances of the Nyström and Column-sampling methods for a learning task. Finally, we provide a quantitative comparison of Isomap and Laplacian Eigenmaps for large-scale face manifold construction on clustering and classification tasks.

### 3.1.1 Manifold learning

Manifold learning considers the problem of extracting low-dimensional structure from high-dimensional data. Given $n$ input points, $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ and $\mathbf{x}_i \in \mathbb{R}^d$, the goal is to find corresponding outputs $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^n$, where $\mathbf{y}_i \in \mathbb{R}^k$, $k \ll d$, such that $\mathbf{Y}$ 'faithfully' represents $\mathbf{X}$. We now briefly review the Isomap and Laplacian Eigenmaps techniques to discuss their computational complexity.

#### Isomap

Isomap aims to extract a low-dimensional data representation that best preserves all pairwise distances between input points, as measured by their geodesic distances along the manifold (Tenenbaum et al., 2000). It approximates the geodesic distance assuming that input space distance provides good approximations for nearby points, and for faraway points it estimates distance as a series of hops between neighboring points. This approximation becomes exact in the limit of infinite data. Isomap can be viewed as an adaptation of Classical Multidimensional Scaling (Cox et al., 2000), in which geodesic distances replace Euclidean distances.

Computationally, Isomap requires three steps:

1. Find $t$ nearest neighbors for each point in input space and construct an undirected neighborhood graph, $\mathcal{G}$, with points as nodes and links between neighbors as edges. This requires $\mathrm{O}(n^2)$ time.

2. Compute approximate geodesic distances, $\Delta_{ij}$, between all pairs of nodes $(i, j)$ by finding shortest paths in $\mathcal{G}$ using Dijkstra's algorithm at each node. Construct a dense, $n \times n$ similarity matrix, $\mathbf{K}$, by centering $\Delta_{ij}^2$, where centering converts distances into similarities. This step takes $\mathrm{O}(n^2 \log n)$ time, dominated by the calculation of geodesic distances.

3. Find the optimal $k$ dimensional representation, $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^n$, such that $\mathbf{Y} = \operatorname{argmin}_{\mathbf{Y}'} \sum_{i,j} \big(\|\mathbf{y}_i' - \mathbf{y}_j'\|_2^2 - \Delta_{ij}^2\big)$. The solution is given by,

$$\mathbf{Y} = (\mathbf{\Sigma}_k)^{1/2}\mathbf{U}_k^\top \tag{3.1}$$

   where $\mathbf{\Sigma}_k$ is the diagonal matrix of the top $k$ singular values of $\mathbf{K}$ and $\mathbf{U}_k$ are the associated singular vectors. This step requires $\mathrm{O}(n^2)$ space for storing $\mathbf{K}$, and $\mathrm{O}(n^3)$ time for its SVD. The time and space complexities for all three steps are intractable for $n = 18\mathrm{M}$.

### Laplacian Eigenmaps

Laplacian Eigenmaps aims to find a low-dimensional representation that best preserves neighborhood relations as measured by a weight matrix $\mathbf{W}$ (Belkin & Niyogi, 2001).[1] The algorithm works as follows:

1. Similar to Isomap, first find $t$ nearest neighbors for each point. Then construct $\mathbf{W}$, a sparse, symmetric $n \times n$ matrix, where $\mathbf{W}_{ij} = \exp\big(-$

[1] The weight matrix should not be confused with the subsampled SPSD matrix, $\mathbf{W}$, associated with the Nyström method. Since sampling-based approximation techniques will not be used with Laplacian Eigenmaps, the notation should be clear from the context.

$\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/\sigma^2\big)$ if $(\mathbf{x}_i, \mathbf{x}_j)$ are neighbors, 0 otherwise, and $\sigma$ is a scaling parameter.

2. Construct the diagonal matrix $\mathbf{D}$, such that $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$, in $\mathrm{O}(tn)$ time.

3. Find the $k$ dimensional representation by minimizing the normalized, weighted distance between neighbors as,

$$\mathbf{Y} = \underset{\mathbf{Y}'}{\operatorname{argmin}} \sum_{i,j} \left( \frac{\mathbf{W}_{ij} \|\mathbf{y}'_i - \mathbf{y}'_j\|_2^2}{\sqrt{\mathbf{D}_{ii}\mathbf{D}_{jj}}} \right). \tag{3.2}$$

   This objective function penalizes nearby inputs for being mapped to faraway outputs, with 'nearness' measured by the weight matrix $\mathbf{W}$ (Chapelle et al., 2006). To find $\mathbf{Y}$, we define $\mathcal{L} = \mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ where $\mathcal{L} \in \mathbb{R}^{n \times n}$ is the symmetrized, normalized form of the graph Laplacian, given by $\mathbf{D} - \mathbf{W}$. Then, the solution to the minimization in (3.2) is

$$\mathbf{Y} = \mathbf{U}_{\mathcal{L},k}^{\top} \tag{3.3}$$

   where $\mathbf{U}_{\mathcal{L},k}^{\top}$ are the bottom $k$ singular vectors of $\mathcal{L}$, excluding the last singular vector corresponding to the singular value 0. Since $\mathcal{L}$ is sparse, it can be stored in $\mathrm{O}(tn)$ space, and iterative methods can be used to find these $k$ singular vectors relatively quickly.

To summarize, in both the Isomap and Laplacian Eigenmaps methods, the two main computational efforts required are neighborhood graph construction/manipulation and SVD of a symmetric positive semidefinite (SPSD) matrix. In the next section, we will further discuss the Nyström and Column-sampling methods in the context of manifold learning, and we will describe the graph operations in Section 3.1.3.

### 3.1.2 Approximation experiments

Since we aimed to use sampling-based SVD approximation to scale Isomap, we first examined how well the Nyström and Column-sampling methods approximated low-dimensional embeddings, i.e., $\mathbf{Y} = (\mathbf{\Sigma}_k)^{1/2}\mathbf{U}_k^\top$. Using (2.3), the Nyström low-dimensional embeddings are:

$$\widetilde{\mathbf{Y}}^{nys} = \widetilde{\mathbf{\Sigma}}^{1/2}_{nys,k}\widetilde{\mathbf{U}}^\top_{nys,k} = \big(\mathbf{\Sigma}_W)_k^{1/2}\big)^+\mathbf{U}^\top_{W,k}\mathbf{C}^\top. \tag{3.4}$$

Similarly, from (2.4) we can express the Column-sampling low-dimensional embeddings as:

$$\widetilde{\mathbf{Y}}^{col} = \widetilde{\mathbf{\Sigma}}^{1/2}_{col,k}\widetilde{\mathbf{U}}^\top_{col,k} = \sqrt[4]{\frac{n}{l}}\big(\mathbf{\Sigma}_C)_k^{1/2}\big)^+\mathbf{V}^\top_{C,k}\mathbf{C}^\top. \tag{3.5}$$

Both approximations are of a similar form. Further, notice that the optimal low-dimensional embeddings are in fact the square root of the optimal rank $k$ approximation to the associated SPSD matrix, i.e., $\mathbf{Y}^\top\mathbf{Y} = \mathbf{K}_k$, for Isomap. As such, there is a connection between the task of approximating low-dimensional embeddings and the task of generating low-rank approximate spectral reconstructions, as discussed in Section 2.2.2. Recall that the theoretical analysis in Section 2.2.2 as well as the empirical results in Section 2.2.3 both suggested that the Nyström method was superior in its spectral reconstruction accuracy. Hence, we performed an empirical study using the datasets from Table 2.1 to measure the quality of the low-dimensional embeddings generated by the two techniques and see if the same trend exists.

We measured the quality of the low-dimensional embeddings by calculating the extent to which they preserve distances, which is the appropriate criterion in the context of manifold learning. For each dataset, we started with a kernel matrix, $\mathbf{K}$, from which we computed the associated $n \times n$ squared distance matrix, $\mathbf{D}$, using the fact that $\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \mathbf{K}_{ii} + \mathbf{K}_{jj} - 2\mathbf{K}_{ij}$. We then computed the approximate low-dimensional embeddings using the Nyström and Column-sampling methods, and then used these embeddings to compute the associated approximate squared distance matrix, $\widetilde{\mathbf{D}}$. We measured accuracy using the notion of relative accuracy defined in (2.24), which can be expressed in terms of distance matrices as:

$$\text{relative accuracy} = \frac{\|\mathbf{D} - \mathbf{D}_k\|_F}{\|\mathbf{D} - \widetilde{\mathbf{D}}\|_F},$$

where $\mathbf{D}_k$ corresponds to the distance matrix computed from the optimal $k$ dimensional embeddings obtained using the singular values and singular vectors of $\mathbf{K}$. In our experiments, we set $k = 100$ and used various numbers of sampled columns, ranging from $l = n/50$ to $l = n/5$. Figure 3.1 presents the results of

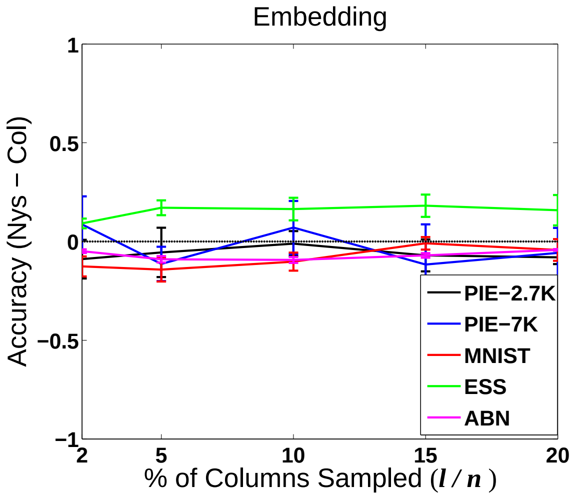


Figure 3.1: Embedding accuracy of Nyström and Column-Sampling. Values above zero indicate better performance of Nyström and vice-versa.

our experiments. Surprisingly, we do not see the same trend in our empirical results for embeddings as we previously observed for spectral reconstruction, as the two techniques exhibit roughly similar behavior across datasets. As a result, we decided to use both the Nyström and Column-sampling methods for our subsequent manifold learning study.

### 3.1.3 Large-scale learning

The following sections outline the process of learning a manifold of faces. We first describe the datasets used in Section 3.1.3. Section 3.1.3 explains how to extract nearest neighbors, a common step between Laplacian Eigenmaps and Isomap. The remaining steps of Laplacian Eigenmaps are straightforward, so the subsequent sections focus on Isomap, and specifically on the computational efforts required to generate a manifold using Webfaces-18M.

### Datasets

We used two datasets of faces consisting of 35K and 18M images. The CMU PIE face dataset (Sim et al., 2002) contains $41,368$ images of 68 subjects under 13 different poses and various illumination conditions. A standard face detector extracted $35,247$ faces (each $48 \times 48$ pixels), which comprised our 35K set (PIE-35K). We used this set because, being labeled, it allowed us to perform quantitative comparisons. The second dataset, named Webfaces-18M, contains 18.2 million images of faces extracted from the Web using the same face detector. For both datasets, face images were represented as 2304 dimensional pixel vectors which were globally normalized to have zero mean and unit variance. No other pre-processing, e.g., face alignment, was performed. In contrast, He et al. (2005) used well-aligned faces (as well as much smaller data sets) to learn face manifolds. Constructing Webfaces-18M, including face detection and duplicate removal, took 15 hours using a cluster of several hundred machines. We used this cluster for all experiments requiring distributed processing and data storage.

### Nearest neighbors and neighborhood graph

The cost of naive nearest neighbor computation is $\mathrm{O}(n^2)$, where $n$ is the size of the dataset. It is possible to compute exact neighbors for PIE-35K, but for Webfaces-18M this computation is prohibitively expensive. So, for this set, we used a combination of random projections and spill trees (Liu et al., 2004) to get approximate neighbors. Computing 5 nearest neighbors in parallel with spill trees took ∼2 days on the cluster. Figure 3.2 shows the top 5 neighbors for a few randomly chosen images in Webfaces-18M. In addition to this visualization, comparison of exact neighbors and spill tree approximations for smaller subsets suggested good performance of spill trees.

We next constructed the neighborhood graph by representing each image as a node and connecting all neighboring nodes. Since Isomap and Laplacian Eigenmaps require this graph to be connected, we used depth-first search to find its largest connected component. These steps required $O(tn)$ space and time. Constructing the neighborhood graph for Webfaces-18M and finding the largest connected component took 10 minutes on a single machine using the OpenFST library (Allauzen et al., 2007).

For neighborhood graph construction, the ‘right’ choice of number of neighbors, $t$, is crucial. A small $t$ may give too many disconnected components, while a large $t$ may introduce unwanted edges. These edges stem from inadequately sampled regions of the manifold and false positives introduced by the face detector. Since Isomap needs to compute shortest paths in the neighborhood graph, the presence of bad edges can adversely impact these computations. This is known as the problem of leakage or ‘short-circuits’ (Balasubramanian & Schwartz, 2002). Here, we chose $t = 5$ and also enforced an upper limit on neighbor distance to alleviate the problem of leakage. We used a distance limit corresponding to the 95th percentile of neighbor distances in the PIE-35K

|  | No Upper Limit |  | Upper Limit Enforced |  |
|---|---|---|---|---|
| $t$ | # Comp | % Largest | # Comp | % Largest |
| 1 | 1.7M | 0.05 % | 4.3M | 0.03 % |
| 2 | 97K | 97.2 % | 285K | 80.1 % |
| 3 | 18K | 99.3 % | 277K | 82.2 % |
| 5 | 1.9K | 99.9 % | 275K | 83.1 % |

Table 3.1: Number of components in the Webfaces-18M neighbor graph and the percentage of images within the largest connected component ('% Largest') for varying numbers of neighbors ($t$) with and without an upper limit on neighbor distances.

dataset.

Table 3.1 shows the effect of choosing different values for $t$ with and without enforcing the upper distance limit. As expected, the size of the largest connected component increases as $t$ increases. Also, enforcing the distance limit reduces the size of the largest component. Figure 3.3 shows a few random samples from the largest component. Images not within the largest component are either part of a strongly connected set of images (Figure 3.4) or do not have any neighbors within the upper distance limit (Figure 3.5). There are significantly more false positives in Figure 3.5 than in Figure 3.3, although some of the images in Figure 3.5 are actually faces. Clearly, the distance limit introduces a trade-off between filtering out non-faces and excluding actual faces from the largest component.[2]

[2] To construct embeddings with Laplacian Eigenmaps, we generated **W** and **D** from nearest neighbor data for images within the largest component of the neighborhood graph and solved (3.3) using a sparse eigensolver.

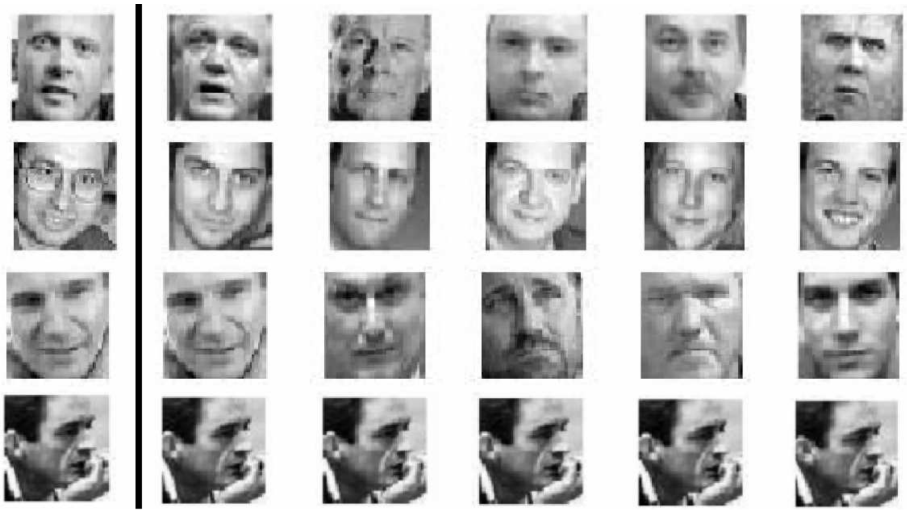


Figure 3.2: Visualization of neighbors for Webfaces-18M. The first image in each row is the target, and the next five are its neighbors.

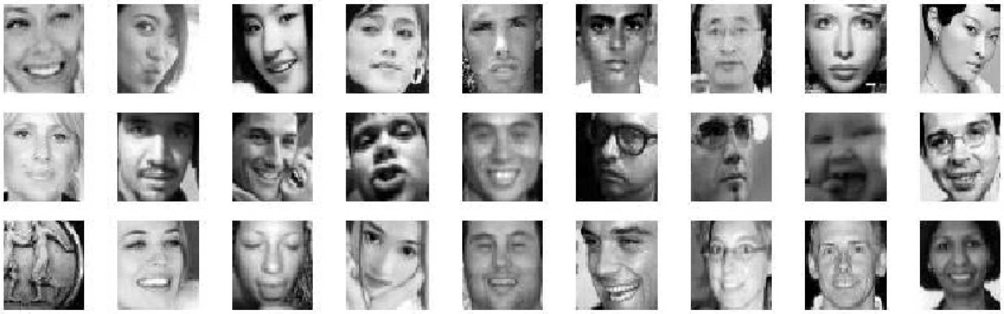


Figure 3.3: A few random samples from the largest connected component of the Webfaces-18M neighborhood graph.

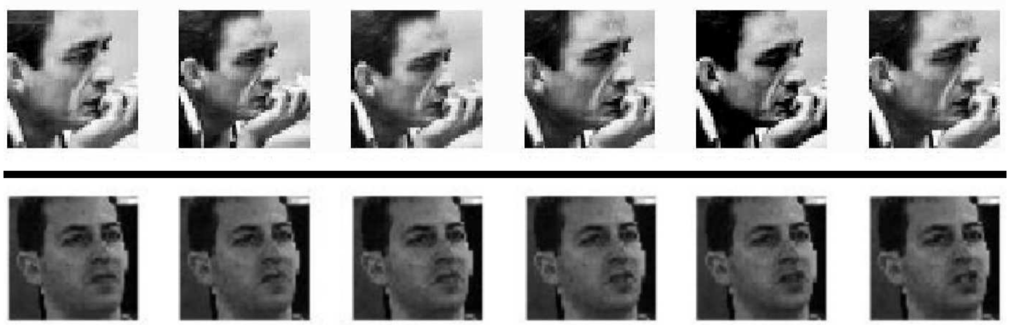


Figure 3.4: Visualization of disconnected components of the neighborhood graphs from Webfaces-18M (top row) and from PIE-35K (bottom row). The neighbors for each of these images are all within this set, thus making the entire set disconnected from the rest of the graph. Note that these images are not exactly the same.

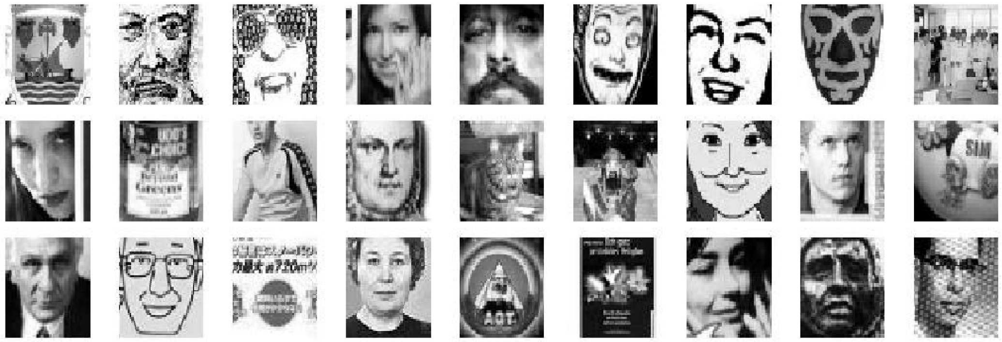


Figure 3.5: Visualization of disconnected components containing exactly one image. Although several of the images above are not faces, some are actual faces, suggesting that certain areas of the face manifold are not adequately sampled by Webfaces-18M.

### Approximating geodesics

To construct the similarity matrix $\mathbf{K}$ in Isomap, one approximates geodesic distance by shortest-path lengths between every pair of nodes in the neighborhood graph. This requires $\mathrm{O}(n^2 \log n)$ time and $\mathrm{O}(n^2)$ space, both of which are prohibitive for 18M nodes. However, since we use sampling-based approximate decomposition, we need only $l \ll n$ columns of $\mathbf{K}$, which form the submatrix $\mathbf{C}$. We thus computed geodesic distance between $l$ randomly selected nodes (called landmark points) and the rest of the nodes, which required $\mathrm{O}(ln \log n)$ time and $\mathrm{O}(ln)$ space. Since this computation can easily be parallelized, we performed geodesic computation on the cluster and stored the output in a distributed fashion. The overall procedure took 60 minutes for Webfaces-18M using $l = 10\text{K}$. The bottom four rows in Figure 3.7 show sample shortest paths for images within the largest component for Webfaces-18M, illustrating smooth transitions between images along each path.

### Generating low-dimensional embeddings

Before generating low-dimensional embeddings in Isomap, one needs to convert distances into similarities using a process called centering (Cox et al., 2000). For the Nyström approximation, we computed $\mathbf{W}$ by double centering $\mathbf{D}$, the $l \times l$ matrix of squared geodesic distances between all landmark nodes, as $\mathbf{W} = -\frac{1}{2}\mathbf{HDH}$, where $\mathbf{H} = \mathbf{I}_l - \frac{1}{l}\mathbf{1}\mathbf{1}^\top$ is the centering matrix, $\mathbf{I}_l$ is the $l \times l$ identity matrix and $\mathbf{1}$ is a column vector of all ones. Similarly, the matrix $\mathbf{C}$ was obtained from squared geodesic distances between the landmark nodes and all other nodes using single-centering as described in de Silva and Tenenbaum (2003).

For the Column-sampling approximation, we decomposed $\mathbf{C}^\top\mathbf{C}$, which we constructed by performing matrix multiplication in parallel on $\mathbf{C}$. For both approximations, decomposition on an $l \times l$ matrix ($\mathbf{C}^\top\mathbf{C}$ or $\mathbf{W}$) took about one hour. Finally, we computed low-dimensional embeddings by multiplying the scaled singular vectors from approximate decomposition with $\mathbf{C}$. For Webfaces-18M, generating low dimensional embeddings took 1.5 hours for the Nyström method and 6 hours for the Column-sampling method.

### 3.1.4 Manifold evaluation

Manifold learning techniques typically transform the data such that Euclidean distance in the transformed space between *any* pair of points is meaningful, under the assumption that in the original space Euclidean distance is meaningful only in local neighborhoods. Since $K$-means clustering computes Euclidean distances between all pairs of points, it is a natural choice for evaluating these techniques. We also compared the performance of various techniques using nearest neighbor classification. Since CMU-PIE is a labeled dataset, we first focused on quantitative evaluation of different embeddings using face pose as class labels. The PIE set contains faces in 13 poses, and such a fine sampling of the pose space makes clustering and classification tasks very challenging. In all the experiments we fixed the dimension of the reduced space, $k$, to be 100.

The first set of experiments was aimed at finding how well different Isomap approximations perform in comparison to exact Isomap. We used a subset of PIE with 10K images (PIE-10K) since, for this size, exact SVD could be done on a single machine within reasonable time and memory limits. We fixed the number of clusters in our experiments to equal the number of pose classes, and measured clustering performance using two measures, *Purity* and *Accuracy*. Purity measures the frequency of data belonging to the same cluster sharing the same class label, while Accuracy measures the frequency of data from the same class appearing in a single cluster. Thus, ideal clustering will have 100% Purity and 100% Accuracy.

Table 3.2 shows that clustering with Nyström Isomap with just $l = 1\text{K}$ performs almost as well as exact Isomap on this dataset[3]. This matches with the observation made in Williams and Seeger (2000), where the Nyström approximation was used to speed up kernel machines. Further, Column-sampling Isomap performs slightly worse than Nyström Isomap. The clustering results on the full PIE-35K set (Table 3.3) with $l = 10\text{K}$ also affirm this observation. Figure 3.6 shows the optimal 2D projections from different methods for PIE-35K. The Nyström method separates the pose clusters better than Column-sampling, verifying the quantitative results.

The fact that Nyström outperforms Column-sampling is somewhat surprising given the experimental evaluations in Section 3.1.2, where we found the two approximation techniques to achieve similar performance. We believe that the poor performance of Column-sampling Isomap is due to the form of the

[3] The differences are statistically insignificant.

| Methods | Purity (%) | Accuracy (%) |
|---:|---|---|
| PCA | 54.3 (±0.8) | 46.1 (±1.4) |
| Exact Isomap | 58.4 (±1.1) | 53.3 (±4.3) |
| Nyström Isomap | 59.1 (±0.9) | 53.3 (±2.7) |
| Col-Sampling Isomap | 56.5 (±0.7) | 49.4 (±3.8) |
| Laplacian Eigenmaps | 35.8 (±5.0) | 69.2 (±10.8) |

Table 3.2: Results of $K$-means clustering of face poses applied to PIE-10K for different algorithms. Results are averaged over 10 random $K$-means initializations.

similarity matrix $\mathbf{K}$. When using a finite number of data points for Isomap, $\mathbf{K}$ is not guaranteed to be SPSD. We verified that $\mathbf{K}$ was not SPSD in our experiments, and a significant number of top eigenvalues, i.e., those with largest magnitudes, were negative. The two approximation techniques differ in their treatment of negative eigenvalues and the corresponding eigenvectors. The Nyström method allows one to use eigenvalue decomposition (EVD) of $\mathbf{W}$ to yield signed eigenvalues, making it possible to discard the negative eigenvalues and the corresponding eigenvectors. On the contrary, it is not possible to discard these in the Column-based method, since the signs of eigenvalues are lost in the SVD of the rectangular matrix $\mathbf{C}$ (or EVD of $\mathbf{C}^\top\mathbf{C}$), i.e., the presence of negative eigenvalues deteriorates the performance of Column-sampling method more than the Nyström method.

Tables 3.2 and 3.3 also show a significant difference in the Isomap and Laplacian Eigenmaps results. The 2D embeddings of PIE-35K (Figure 3.6) reveal that Laplacian Eigenmaps projects data points into a small compact region, consistent with its objective function defined in (3.2), as it tends to map

| Methods | Purity (%) | Accuracy (%) |
|---:|---|---|
| PCA | 54.6 (±1.3) | 46.8 (±1.3) |
| Nyström Isomap | 59.9 (±1.5) | 53.7 (±4.4) |
| Col-Sampling Isomap | 56.1 (±1.0) | 50.7 (±3.3) |
| Laplacian Eigenmaps | 39.3 (±4.9) | 74.7 (±5.1) |

Table 3.3: Results of $K$-means clustering of face poses applied to PIE-35K for different algorithms. Results are averaged over 10 random $K$-means initializations.

neighboring inputs as nearby as possible in the low-dimensional space. When used for clustering, these compact embeddings lead to a few large clusters and several tiny clusters, thus explaining the high accuracy and low purity of the clusters. This indicates poor clustering performance of Laplacian Eigenmaps, since one can achieve even 100% Accuracy simply by grouping all points into a single cluster. However, the Purity of such clustering would be very low. Finally, the improved clustering results of Isomap over PCA for both datasets verify that the manifold of faces is not linear in the input space.

We also compared the performance of Laplacian Eigenmaps and Isomap embeddings on pose classification.[4] The data was randomly split into a training and a test set, and $K$-Nearest Neighbor (KNN) was used for classification. $K = 1$ gives lower error than higher $K$ as shown in Table 3.4. Also, the classification error is lower for both exact and approximate Isomap than for Laplacian Eigenmaps, suggesting that neighborhood information is better preserved by Isomap (Tables 3.4 and 3.5). Note that, similar to clustering, the

[4] KNN only uses nearest neighbor information for classification. Since neighborhoods are considered to be locally linear in the input space, we expect KNN to perform well in the input space. Hence, using KNN to compare low-level embeddings indirectly measures how well nearest neighbor information is preserved.

| Methods | $K = 1$ | $K = 3$ | $K = 5$ |
|---:|---|---|---|
| Exact Isomap | 10.9 (±0.5) | 14.1 (±0.7) | 15.8 (±0.3) |
| Nyström Isomap | 11.0 (±0.5) | 14.0 (±0.6) | 15.8 (±0.6) |
| Col-Sampling Isomap | 12.0 (±0.4) | 15.3 (±0.6) | 16.6 (±0.5) |
| Laplacian Eigenmaps | 12.7 (±0.7) | 16.6 (±0.5) | 18.9 (±0.9) |

Table 3.4: $K$-nearest neighbor classification error (%) of face pose applied to PIE-10K subset for different algorithms. Results are averaged over 10 random splits of training and test sets. $K = 1$ gives the lowest error.

| Nyström Isomap | Col-Sampling Isomap | Laplacian Eigenmaps |
|:---:|:---:|:---:|
| 9.8 (±0.2) | 10.3 (±0.3) | 11.1 (±0.3) |

Table 3.5: 1-nearest neighbor classification error (%) of face pose applied to PIE-35K for different algorithms. Results are averaged over 10 random splits of training and test sets.

Nyström approximation performs as well as Exact Isomap (Table 3.4). Better clustering and classification results, combined with 2D visualizations, imply that approximate Isomap outperforms exact Laplacian Eigenmaps. Moreover, the Nyström approximation is computationally cheaper and empirically more effective than the Column-sampling approximation. Thus, we used Nyström Isomap to generate embeddings for Webfaces-18M.

After learning a face manifold from Webfaces-18M, we analyzed the results with various visualizations. The top row of Figure 3.7 shows the 2D embeddings from Nyström Isomap. The top left figure shows the face samples from various locations in the manifold. It is interesting to see that embeddings tend to cluster the faces by pose. These results support the good clustering performance observed using Isomap on PIE data. Also, two groups (bottom left and top right) with similar poses but different illuminations are projected at

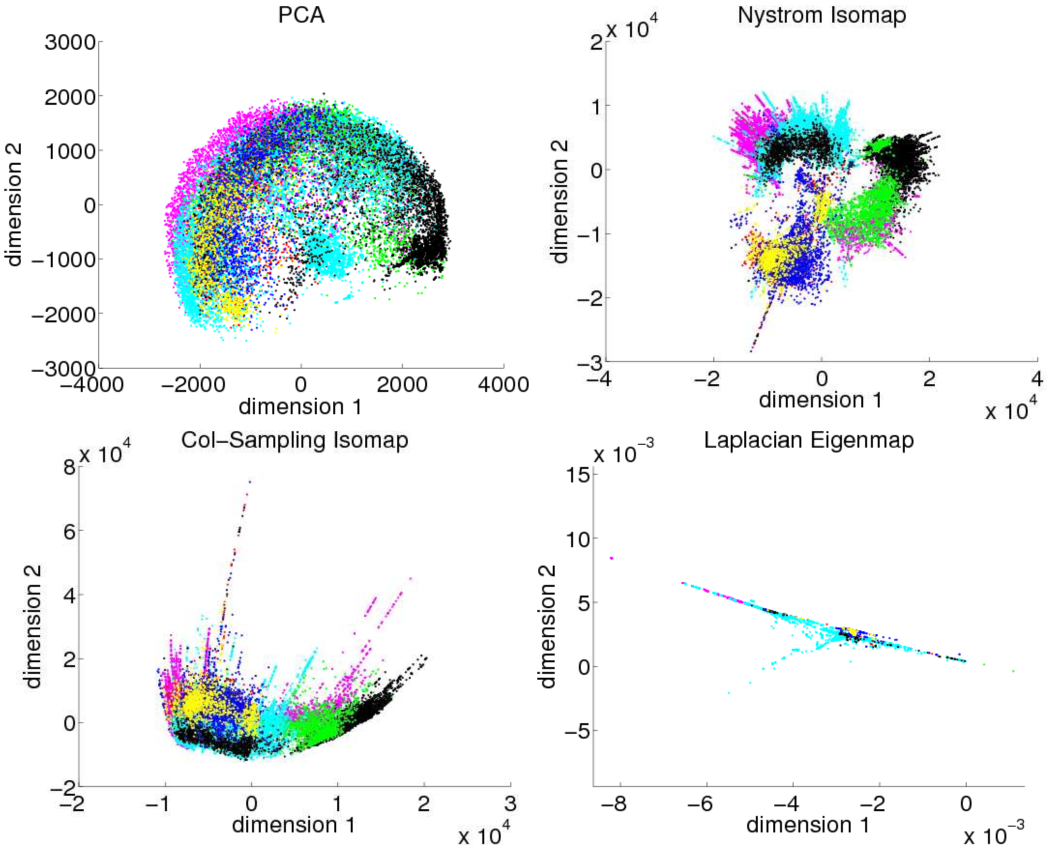


Figure 3.6: Optimal 2D projections of PIE-35K where each point is color coded according to its pose label. Top Left: PCA projections tend to spread the data to capture maximum variance, Top Right: Isomap projections with Nyström approximation tend to separate the clusters of different poses while keeping the cluster of each pose compact, Bottom Left: Isomap projections with Column-sampling approximation have more overlap than with Nyström approximation. Bottom Right: Laplacian Eigenmaps projects the data into a very compact range.

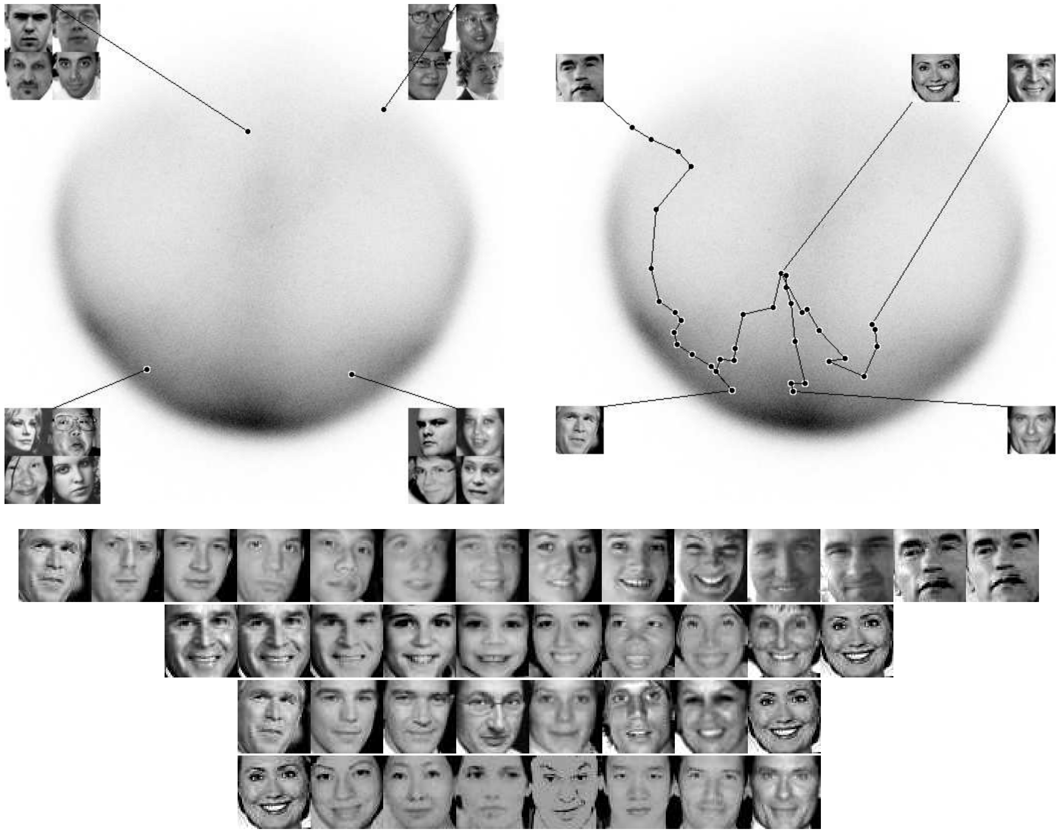


Figure 3.7: 2D embedding of Webfaces-18M using Nyström Isomap (Top row). Darker areas indicate denser manifold regions. Top Left: Face samples at different locations on the manifold. Top Right: Approximate geodesic paths between celebrities. The corresponding shortest-paths are shown in the bottom four rows.

different locations. Additionally, since 2D projections are very condensed for 18M points, one can expect more discrimination for higher $k$, e.g., $k = 100$.

In Figure 3.7, the top right figure shows the shortest paths on the manifold between different public figures. The images along the corresponding paths have smooth transitions as shown in the bottom of the figure. In the limit of infinite samples, Isomap guarantees that the distance along the shortest path between any pair of points will be preserved as Euclidean distance in the embedded space. Even though the paths in the figure are reasonable approximations of straight lines in the embedded space, these results suggest that 18M faces are perhaps not enough samples to learn the face manifold exactly.

## 3.2 Woodbury Approximation

The previous section focused on the application of approximate embeddings in the context of manifold learning. In this section, we will shift focus to matrix reconstruction. We will show how approximate spectral reconstruction can be used in conjunction with the Woodbury inversion lemma to speed up a variety of kernel-based algorithms. Given the superior performance of the Nyström method for spectral reconstruction, as discussed in Chapter 2, we will focus in this section on spectral approximations generated via the Nyström method.

The Woodbury inversion lemma states that the inverse of a rank-$k$ correction of some matrix can be computed by performing a rank-$k$ correction to the inverse of the original matrix. As suggested by Williams and Seeger (2000), low-rank approximations can be combined with the Woodbury inversion lemma to derive an efficient algorithm for inverting kernel matrices. Using the rank-$k$ approximation $\widetilde{\mathbf{K}}$ given by the Nyström method, instead of $\mathbf{K}$, and applying the inversion lemma yields:

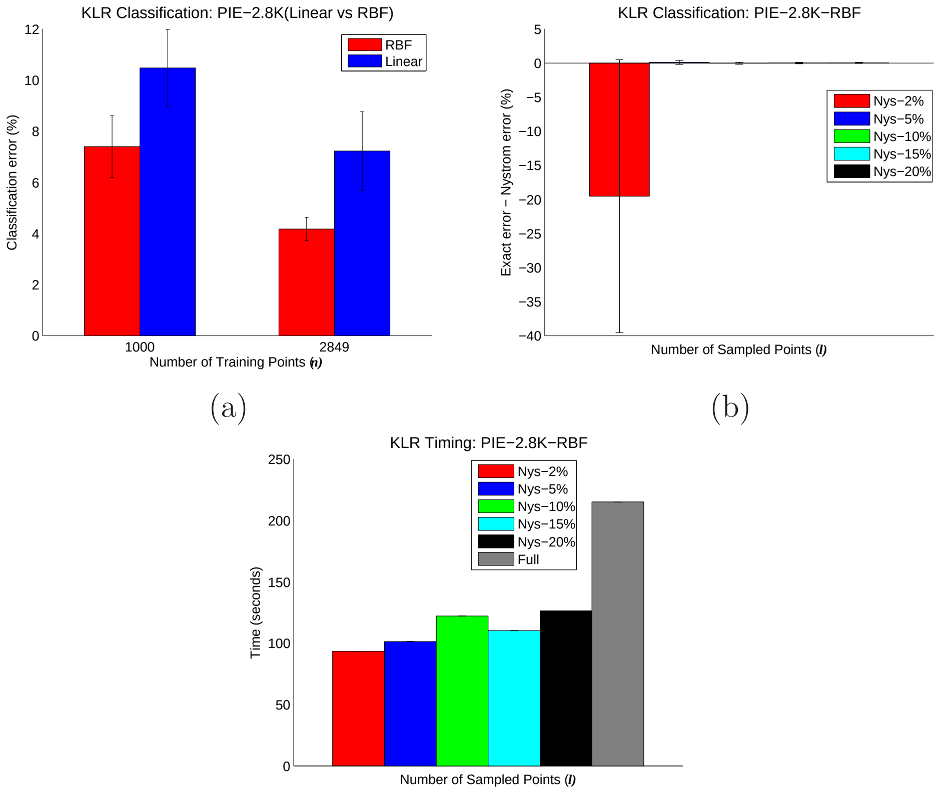
$$
\begin{aligned}
&(\lambda\mathbf{I} + \mathbf{K})^{-1} && (3.6)\\
&\approx \big(\lambda\mathbf{I} + \widetilde{\mathbf{K}}\big)^{-1} && (3.7)\\
&\approx \big(\lambda\mathbf{I} + \mathbf{C}\mathbf{W}_k^{+}\mathbf{C}^\top\big)^{-1} && (3.8)\\
&= \frac{1}{\lambda}\Big(\mathbf{I} - \mathbf{C}\big[\lambda\mathbf{I}_k + \mathbf{W}_k^{+}\mathbf{C}^\top\mathbf{C}\big]^{-1}\mathbf{W}_k^{+}\mathbf{C}^\top\Big). && (3.9)
\end{aligned}
$$

Thus, only an inversion of a matrix of size $k$ is needed as opposed to the original problem of size $n$.

This technique has been previously used to speed up various algorithms including Support Vector Machines (Fine & Scheinberg, 2002) and Gaussian Processes (Williams & Seeger, 2000). In the remainder of this section, we present experiments showing how this technique can also be used to effectively speed up Kernel Logistic Regression and Kernel Ridge Regression.

### 3.2.1 Nyström Logistic Regression

Logistic Regression is a classification algorithm with $N$ parameters, where $N$ is the number of features (Hastie & Tibshirani, 1990). Let $\mathbf{X} \in \mathbb{R}^{N\times n}$ contain $n$ datapoints, with $\mathbf{x}_i = \mathbf{X}^{(i)}$ representing the $i$th datapoint, and let $\mathbf{y} \in \mathbb{R}^n$ be



(a)

(b)

(c)

Figure 3.8: Pose classification experiments for a set of 2.8K faces images with two similar poses. (a) Classification performance for Kernel Logistic Regression with empirical kernel map using linear and RBF kernels. (b) Relative classification performance for Kernel Logistic Regression with empirical kernel map using the Nyström approximation with different percentages of sampled columns. Results are reported in comparison to the performance obtained using the standard Newton method. (c) Timing comparison for the same experiments as in (b), where 'Full' corresponds to the standard Newton method.

the corresponding labels. If we assume $\mathbf{y}_i \in \{0, 1\}$, then Logistic Regression aims to maximize the following loss function:

$$
\begin{aligned}
l(\mathbf{w}) &= \sum_{i=1}^{n} \left[\mathbf{y}_i \ln \mathbf{s}_i + (1-\mathbf{y}_i)\ln(1-\mathbf{s}_i))\right] && (3.10)\\
&= \sum_{i=1}^{n} \left[\mathbf{y}_i \mathbf{w}^\top \mathbf{x}_i - \ln(1+\exp(\mathbf{w}^\top \mathbf{x}_i))\right], && (3.11)
\end{aligned}
$$

where $\sigma(a) = (1+\exp(-a))^{-1}$ is the sigmoid function and $\mathbf{s} \in \mathbb{R}^n$ with $\mathbf{s}_i = \sigma(\mathbf{w}^\top \mathbf{x}_i)$. To avoid overfitting of the training data, a regularization term is often added to the error function. $L_2$ regularization is commonly used in practice, in which case we aim to minimize the following regularized loss:

$$
\min_{\mathbf{w}} E(\mathbf{w}) = \frac{\lambda}{2}\mathbf{w}^\top \mathbf{w} - l(\mathbf{w}), \tag{3.12}
$$

where $\lambda$ is the regularization parameter. Since $E(\mathbf{w})$ is concave, a unique minimum value exists and the function can be minimized using iterative techniques.

Newton's method is an efficient iterative method for minimizing the negative log likelihood in (3.12), by setting its derivatives to zero. The Newton step can be expressed as:

$$
\begin{aligned}
\mathbf{w}_{new} &= \mathbf{w}_{old} - \mathbf{H}^{-1}\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} && (3.13)\\
&= \left(\lambda \mathbf{I}_N + \mathbf{X}\mathbf{R}\mathbf{X}^\top\right)^{-1}\mathbf{X}\mathbf{R}(\mathbf{X}^\top \lambda \mathbf{w}_{old} + \mathbf{R}^{-1}(\mathbf{y}-\mathbf{s})), && (3.14)
\end{aligned}
$$

where $\mathbf{H}$ is the Hessian matrix whose elements are the second derivatives of

$E(\mathbf{w})$ with respect to $\mathbf{w}$ and $\mathbf{R} \in \mathbb{R}^{n \times n}$ is the diagonal matrix with $\mathbf{R}_{ii} = \mathbf{s}_i(1 - \mathbf{s}_i)$. Thus, each iteration of the Newton method requires the inversion of an $N \times N$ matrix that depends on $\mathbf{w}_{old}$, taking approximately $\mathrm{O}(N^3)$.

Our proposed algorithm involves speeding up the computation of the matrix inverse using the Nyström method. For each Newton iteration, we define $\mathbf{T} = \mathbf{X}\mathbf{R}\mathbf{X}^\top$ and generate the associated Nyström approximation: $\widetilde{\mathbf{T}} = \mathbf{C}\mathbf{W}_k^+\mathbf{C}^\top$. We can now speed up the inverse computation in (3.14) as follows:

$$\begin{aligned}
(\lambda\mathbf{I} + \mathbf{T})^{-1} &\approx \big(\lambda\mathbf{I} + \widetilde{\mathbf{T}}\big)^{-1} \\
&= \big(\lambda\mathbf{I} + \mathbf{C}\mathbf{W}_k^+\mathbf{C}^\top\big)^{-1} \qquad (3.15) \\
&= \frac{1}{\lambda}\Big(\mathbf{I}_N - \mathbf{C}\big[\lambda\mathbf{I}_k + \mathbf{W}_k^+\mathbf{C}^\top\mathbf{C}\big]^{-1}\mathbf{W}_k^+\mathbf{C}^\top\Big), \qquad (3.16)
\end{aligned}$$

where we get (3.16) via the Woodbury Inversion Lemma, i.e., (3.9).

As shown in Chapter 2, the Nyström method does not typically generate good approximations for small singular values and their associated singular vectors, which are exactly the singular values/vectors that have the largest impact on the inverse of the matrix. However, the regularization term damps the effect of small singular values, thus making the Nyström method feasible. Each iteration of Nyström Newton takes $O(k^3)$ time to calculate $[\lambda\mathbf{I}_k + \mathbf{W}_k^+\mathbf{C}^\top\mathbf{C}]^{-1}$ as well as $O(nN + N^2)$ for matrix multiplication in (3.14). In order to kernelize this algorithm, we replace $\mathbf{X}$ with the kernel matrix, $\mathbf{K} \in \mathbb{R}^{n \times n}$, i.e., using the empirical kernel map as features.[5] In this case, the runtime is $O(n^2)$.

[5] Using the empirical kernel map leads to a different formulation than the standard Kernel Logistic Regression algorithm (KLR) as described in Keerthi et al. (2005). The difference

In Figure 3.8, we present results that show the performance of our proposed algorithm for a sample dataset consisting of a subset of the PIE dataset containing 2.8K images with two similar face poses. In these experiments, we ran 10 trials, and for each trial 2.4K points were used for training while the remaining 400 points were held out for testing. For all experiments, the regularization parameter and the RBF parameter were determined via cross validation. In Figure 3.8(a), we compare classification performance using linear and RBF kernels, and the results show the benefit of using non-linear kernels for this classification task. Next, we performed experiments to see the effect of the Nyström approximation on the quality and runtime of the algorithm. Figure 3.8(b) shows that using the Nyström approximation with as little as 5% of sampled columns converges to same solution as exact Logistic Regression with empirical kernel map, while Figure 3.8(c) shows the associated speed gain obtained by using of the Nyström method.

### 3.2.2 Kernel Ridge Regression

Kernel Ridge Regression (KRR) is a powerful regression algorithm that is commonly used in practice. The dual optimization problem solved by KRR (Saunders et al., 1998) can be written as follows:

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^m} \lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha} + \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^\top\mathbf{y}, \tag{3.17}$$

stems from the use of slightly different regularization terms: the $L_2$ regularizer for Logistic Regression with empirical map is of the form $\mathbf{w}^\top\mathbf{w}$ while the penalty term in KLR is of the form $\mathbf{w}^\top\mathbf{K}\mathbf{w}$.

where $\lambda = n\lambda_0 > 0$ is the ridge parameter. The problem admits the closed form solution $\boldsymbol{\alpha} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}$. Clearly, substituting $\mathbf{K}$ by $\widetilde{\mathbf{K}}$ would allow us to employ the Woodbury approximation as presented in (3.9).

| Dataset | # Points ($n$) | # Features ($d$) | Kernel | Largest label ($M$) |
|---|---|---|---|---|
| ABALONE | 4177 | 8 | RBF | 29 |
| KIN-8nm | 4000 | 8 | RBF | 1.5 |

Table 3.6: Description of datasets used in our KRR perturbation experiments (Asuncion & Newman, 2007; Ghahramani, 1996). Note that $M$ denotes the largest possible magnitude of the regression labels.

We next present experiments that directly illustrate the connection between spectral reconstruction error and the quality of the Woodbury approximation, i.e., we will show the impact of kernel perturbation on the output of Kernel Ridge Regression. For our experiments, we worked with the datasets listed in Table 3.6, and for each dataset, we randomly selected 80% of the points to generate $\mathbf{K}$ and used the remaining 20% as the test set, $\mathcal{T}$. For each test-train split, we first performed grid search to determine the optimal ridge for $\mathbf{K}$, as well as the associated optimal hypothesis, $h(\cdot)$. Next, using this optimal ridge, we generated a set of Nyström approximations, using various numbers of sampled columns, i.e., $l$ ranging from 1% to 50% of $n$. For each Nyström approximation, $\widetilde{\mathbf{K}}$, we computed the associated hypothesis $h'(\cdot)$ using the same ridge and measured the distance between $h$ and $h'$ as follows:

$$\text{average absolute error} = \frac{\sum_{x \in \mathcal{T}} |h'(x) - h(x)|}{|\mathcal{T}|}. \tag{3.18}$$

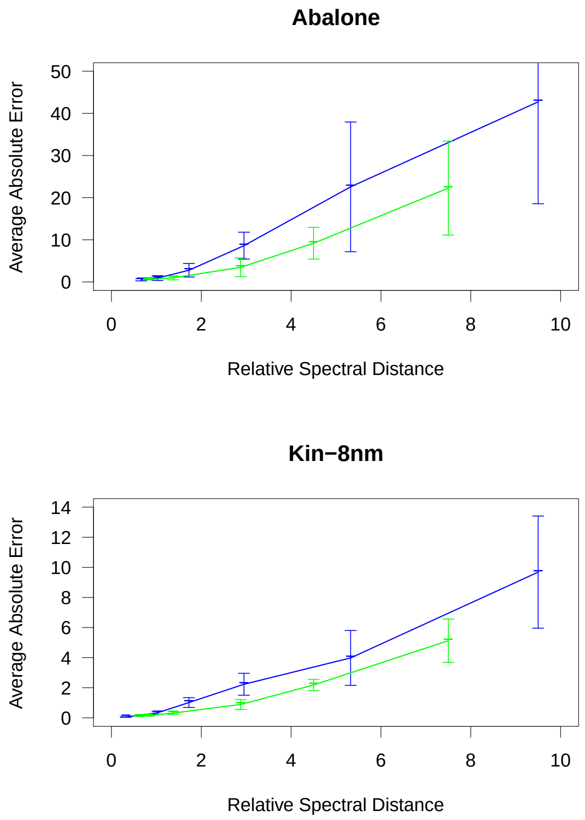


Figure 3.9: Average absolute error of the Kernel Ridge Regression hypothesis, $h'(\cdot)$, generated from the Nyström approximation, $\widetilde{\mathbf{K}}$, as a function of relative spectral distance $\|\widetilde{\mathbf{K}}-\mathbf{K}\|_2/\|\mathbf{K}\|_2$. For each dataset, the reported results show the average absolute error as a function of relative spectral distance for both the full dataset and for a subset of the data containing $n = 2000$ points. Results for the same value of $n$ are connected with a line. The different points along the lines correspond to various numbers of sampled columns, i.e., $l$ ranging from 1% to 50% of $l$.

We measured the distance between $\widetilde{\mathbf{K}}$ and $\mathbf{K}$ as follows:

$$\text{relative spectral distance} = \frac{\|\widetilde{\mathbf{K}} - \mathbf{K}\|_2}{\|\mathbf{K}\|_2} \times 100. \tag{3.19}$$

Figure 3.9 presents results for each dataset using all $n$ points and a subset of 2000 points. The plots show the average absolute error of $h(\cdot)$ as a function of relative spectral distance. Results corresponding to the same value of $n$ are connected with a line, and points along each line correspond to various numbers of sampled columns, i.e., $l$ ranging from 1% to 50% of $n$. The values of the relative spectral distance are computed as the means over the splits for the same value of $l$. The values of the average absolute error are computed correspondingly. These experiments clearly show the impact that spectral reconstruction accuracy has on the hypothesis generated by KRR, as the results suggest a linear relationship between kernel approximation and average absolute error. Corresponding theoretical results, presented in Section 5.3 (specifically Proposition 5.1), corroborate these empirical findings.

## 3.3 Summary

Sampling-based matrix approximations have been used in a variety of large-scale machine learning tasks. In this chapter, we focused on two specific applications. We first studied non-linear dimensionality reduction using unsupervised manifold learning and our experimental results reveal that Isomap coupled with the Nyström approximation can effectively extract low-dimensional structure from datasets containing millions of images. In fact, the techniques we described in the context of large-scale manifold learning are currently used by Google for its "People Hopper" application which runs on the social networking site Orkut (Kumar & Rowley, 2010). Next, we discussed how matrix approximation techniques can be used in conjunction with the Woodbury approximation, and we presented results for Kernel Logistic Regression and Kernel Ridge Regression that show the effectiveness of this approach.

# Chapter 4

# Sampling Schemes

Although we have been focusing on sampling based techniques for matrix approximation, to this point, we have sidestepped the important issue of how to sample columns of the matrix, i.e., how to obtain $\mathbf{C}$ from $\mathbf{K}$. As we shall see in this chapter, the selection of columns can significantly influence the accuracy of approximation, and hence we will now discuss various sampling options used to select columns from $\mathbf{K}$. Furthermore, given the favorable performance of the Nyström method relative to the column-sampling method, both in the analysis in Chapter 2 and the empirical studies in Chapter 3, our discussion in this chapter will focus on the Nyström method.

The material in this chapter is organized as follows. Section 4.1 presents the most basic sampling techniques which involve sampling columns from a fixed distribution over the columns. Next, in Section 4.2 we will discuss more sophisticated adaptive sampling techniques that choose a better subset of columns but at a greater cost. Finally, in Section 4.3 we present an ensemble meta-algorithm for combining multiple matrix approximations that generate improved approximations and naturally fit within a distributed computing environment, an issue of great practical significance given the prevalence of distributed computing frameworks to handle large-scale learning problems.

## 4.1 Fixed Sampling

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., 2006b; Drineas & 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.[1] In Drineas and Mahoney (2005) however, the

[1] Although Drineas and Mahoney (2005) claims to weight each column proportional to $\mathbf{K}_{ii}^2$, they in fact use the diagonal sampling we present in this work, i.e., weights proportional to $\mathbf{K}_{ii}$ (Drineas, 2008).

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.

In the remainder of this section we present novel experimental results comparing the performance of these sampling methods on several data sets. Previous works 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 more scalable for large-scale applications and which has been utilized in some theoretical analyses of the Nyström method.

### 4.1.1 Datasets

We used 5 datasets from a variety of applications, e.g., computer vision and biology, as described in Table 4.1. SPSD kernel matrices were generated by mean centering the datasets 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.

| 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 |

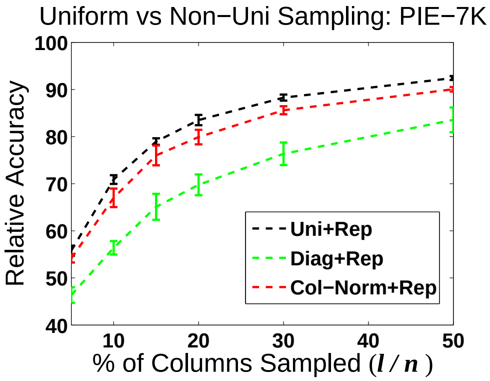
Table 4.1: Description of the datasets and kernels used in our fixed and adaptive sampling experiments (Sim et al., 2002; LeCun & Cortes, 1998; Gustafson et al., 2006; Asuncion & Newman, 2007). '$d$' denotes the number of features in input space.

### 4.1.2 Experiments

We used the datasets 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 accuracies were measured relative to the best rank-$k$ approximation ($\mathbf{K}_k$) using the same notion of relative accuracy originally defined in (2.24):

$$\text{relative accuracy} = \frac{\|\mathbf{K} - \mathbf{K}_k\|_F}{\|\mathbf{K} - \tilde{\mathbf{K}}_k\|_F}.$$

We fixed $k = 100$ for all experiments, a value that captures more than 90% of the spectral energy for each dataset. We first compared the effectiveness of the three sampling techniques using sampling with replacement. The results for PIE-7K are presented in Figure 4.1(a) and summarized for all datasets in Figure 4.1(b). The results across all datasets 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



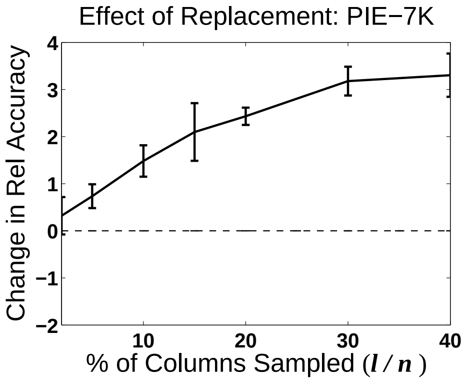
(a)

| $l/n$ | Dataset | 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 4.1: (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.

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)

| Dataset | 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 4.2: 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 datasets for various $l/n$ percentages.

Next, we compared the performance of uniform sampling with and without replacement. Figure 4.2(a) illustrates the effect of replacement for the PIE-7K dataset for different $l/n$ ratios. Similar results for the remaining datasets are summarized in Figure 4.2(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 datasets (it is also the most commonly used method in practice). In Chapter 5 we will present a theoretical analysis of the Nyström method using precisely this type of sampling.

## 4.2 Adaptive Sampling

In Section 4.1, 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 $\mathrm{O}(ln)$ entries of $\mathbf{K}$. The Sparse Matrix Greedy Approximation (SMGA) (Smola & Schölkopf, 2000) and the Incomplete Cholesky Decomposition (ICL) (Fine & Scheinberg, 2002; Bach & Jordan, 2002) were the first such adaptive schemes suggested for the Nyström method. SGMA 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 $\mathrm{O}(sl^2n)$, which is of the same order as the Nyström method itself when $s$ is a constant and $k = l$ (see Section 2.1.2 for details).

Whereas SGMA 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 $\mathrm{O}(l^2 n)$. 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 & 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); 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 SGMA, ICL, $K$-means and the *adaptive-full* techniques.

**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 4.3: 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.

### 4.2.1 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

```
Update-Probability-Partial($R$)
$\mathbf{C}' \leftarrow$ columns of $\mathbf{K}$ corresponding to indices in $R$
$k' \leftarrow$ Choose-rank() $\rhd$ low-rank ($k$) or $\frac{|R|}{2}$
$\mathbf{\Sigma}_{nys}$, $\mathbf{U}_{nys} \leftarrow$ Do-Nyström ($\mathbf{C}', k'$) $\rhd$ see eq (2.3)
$\mathbf{C}'_{nys} \leftarrow$ Spectral reconstruction using $\mathbf{\Sigma}_{nys}$, $\mathbf{U}_{nys}$
$\mathbf{E} \leftarrow \mathbf{C}' - \mathbf{C}'_{nys}$
for $j \in [1 \ldots n]$ do
    if $j \in R$ then
        $P_j \leftarrow 0$ $\rhd$ sample without replacement
    else $P_j \leftarrow ||\, \mathbf{E}_{(j)} \,||_2^2$
$P \leftarrow \frac{P}{||P||_2}$
return $P$
```

Figure 4.4: The proposed adaptive sampling technique that uses a small subset of the original matrix **K** to adaptively choose columns. It does not need to store or operate on **K**.

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 4.3. Note that the sampling step, UPDATE-PROBABILITY-FULL, requires a full pass over **K** at each step, and hence O($n^2$) time and space.

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 **K** is updated proportional to this error. We com-

Table 4.2: Nyström spectral reconstruction accuracy for various sampling methods for all datasets 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 dataset, i.e., each row of the table, while numbers in italics indicate adaptive techniques that were outperformed by random sampling on each dataset. Dashes ('-') indicate experiments that were too costly to run on the larger datasets (ESS, PIE-7K).

| $l/n\%$ | Dataset | Uniform | ICL | SGMA | Adapt-Part | $K$-means | Adapt-Full |
|---|---|---|---|---|---|---|---|
| 5% | PIE-2.7K | 39.3 (2.1) | 41.6 | 54.4 (0.5) | 42.4 (1.4) | **61.7** (0.7) | 44.2 (0.9) |
| | PIE-7K | 56.8 (0.7) | *50.1* | **68.1** (0.9) | 62.1 (0.9) | - | - |
| | MNIST | 47.0 (1.0) | *41.5* | 59.1 (0.7) | 49.1 (0.9) | **72.3** (0.7) | 50.1 (0.6) |
| | ESS | 44.6 (2.4) | *25.2* | **61.9** (0.5) | 50.1 (0.4) | 58.5 (1.8) | - |
| | ABN | 48.7 (7.2) | *15.6* | **67.1** (1.4) | *20.3* (4.5) | 65.0 (2.8) | 53.5 (1.6) |
| 10% | PIE-2.7K | 58.0 (0.8) | 61.1 | **73.1** (0.4) | 61.3 (0.7) | **73.1** (2.6) | 62.7 (0.7) |
| | PIE-7K | 73.1 (1.1) | *60.8* | 74.5 (0.6) | **76.8** (0.8) | - | - |
| | MNIST | 67.5 (0.9) | *58.3* | 72.2 (0.4) | 69.2 (0.6) | **80.4** (2.8) | 68.8 (0.3) |
| | ESS | 66.7 (1.6) | *39.1* | 74.7 (0.5) | 70.0 (1.8) | **76.3** (2.3) | - |
| | ABN | 61.3 (4.3) | *25.8* | 68.5 (2.3) | *32.4* (8.4) | **78.2** (6.9) | *60.2* (2.5) |
| 20% | PIE-2.7K | 75.1 (1.3) | 80.5 | 85.9 (0.2) | 78.3 (0.6) | **86.5** (0.7) | 80.2 (0.5) |
| | PIE-7K | **86.4** (0.3) | *69.5* | *79.4* (0.5) | *85.7* (1.0) | - | - |
| | MNIST | 83.2 (0.2) | *77.9* | *78.9* (0.3) | 83.9 (0.3) | **90.4** (0.2) | *80.9* (0.6) |
| | ESS | 82.9 (0.9) | *55.3* | *79.4* (0.7) | **83.9** (0.5) | 83.5 (0.3) | - |
| | ABN | 83.1 (1.3) | *41.2* | *66.2* (3.7) | *43.8* (12.6) | **90.0** (0.7) | *61.5* (3.3) |

pute the error for $\mathbf{C}'$, which is much smaller than $\mathbf{K}$, thus avoiding the $\mathrm{O}(n^2)$ computation. As described in (2.13), 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 itera-

Table 4.3: Run times (in seconds) corresponding to Nyström spectral reconstruction results in Table 4.2. Numbers in bold indicate the fastest algorithm for each dataset, i.e., each row of the table, while numbers in italics indicate the slowest algorithm for each dataset. Dashes ('-') indicate experiments that were too costly to run on the larger datasets (ESS, PIE-7K).

| $l/n\%$ | Dataset | Uniform | ICL | SGMA | Adapt-Part | $K$-means | Adapt-Full |
|---|---|---|---|---|---|---|---|
| 5% | PIE-2.7K | **1** | 3 | 12 | 1 | *65* | 33 |
| | PIE-7K | **2** | 18 | *59* | 6 | - | - |
| | MNIST | **1** | 6 | 21 | 2 | 42 | *65* |
| | ESS | **1** | 16 | *62* | 9 | 4 | - |
| | ABN | **1** | 7 | 24 | 2 | 3 | *100* |
| 10% | PIE-2.7K | **1** | 13 | 40 | 5 | *428* | 41 |
| | PIE-7K | **13** | 69 | *244* | 36 | - | - |
| | MNIST | **2** | 17 | 66 | 6 | *142* | 74 |
| | ESS | **3** | 55 | *152* | 13 | 20 | - |
| | ABN | **1** | 25 | 84 | 6 | 7 | *90* |
| 20% | PIE-2.7K | **5** | 34 | 126 | 17 | *1351* | 61 |
| | PIE-7K | **79** | 280 | *1107* | 250 | - | - |
| | MNIST | **8** | 37 | 134 | 27 | *860* | 112 |
| | ESS | **12** | 87 | *458* | 48 | 62 | - |
| | ABN | **9** | 73 | *332* | 28 | 24 | 108 |

tion. 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 is $\mathrm{O}(nlk' + l^3)$ and requires at most the storage of $l$ columns of $\mathbf{K}$. The details of the proposed sampling technique are outlined in Figure 4.4.

### 4.2.2 Experiments

We used the datasets already shown in Table 2.1, and compared the effect of different sampling techniques on the relative accuracy of Nyström spectral reconstruction for $k = 100$. Experiments were conducted in Matlab, with ICL code from Cawley and Talbot (2004), SGMA code from Smola (2000) and $K$-means code from authors of Zhang et al. (2008). The relative accuracy results across datasets for varying values of $l$ are presented in Table 4.2, while the corresponding timing results are detailed in Table 4.2. These empirical results reveal the strengths and weaknesses of the adaptive techniques. SGMA and $K$-means often generate the best relative accuracy, but are also the most expensive algorithms. $K$-means in particular is costly when working with a large number of features, e.g., it was difficult to run $K$-means on our 7K dataset containing 2304 features. Our 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. In fact, it is faster than all other adaptive techniques and outperforms uniform sampling on 4 of 5 datasets. ICL is also fast, though its performance is 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 strongly 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.

In summary, the 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 with approximation accuracy roughly equal to approximations generated from slightly larger subsets of randomly sampled columns. This trade-off between time and approximation quality will be revisited in Section 4.3.2, where we compare the approximation performance given fixed-time constraints.

## 4.3 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 $\widetilde{\mathbf{K}}_r$. 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 $\mathbf{W}_r$.[2] The learner further receives a

[2] In this study, we focus on the class of base learners generated from the Nyström approximation with uniform sampling of columns. Alternatively, base learners could be generated using other (or a combination of) sampling schemes discussed in Sections 4.1 and 4.2.

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

$$\widetilde{\mathbf{K}}^{ens} = \sum_{r=1}^{p} \mu_r \widetilde{\mathbf{K}}_r. \tag{4.1}$$

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

$$\min_{\boldsymbol{\mu}} \ \lambda \|\boldsymbol{\mu}\|_2^2 + \| \sum_{r=1}^{p} \mu_r \widetilde{\mathbf{K}}_r^V - \mathbf{K}_V \|_F^2, \tag{4.2}$$

where $\mathbf{K}_V$ denotes the matrix formed by the columns of the samples $S$ and $V$ and $\lambda > 0$. 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 $O(pl^3+plkn+C_{\boldsymbol{\mu}})$, where $C_{\boldsymbol{\mu}}$ is the cost of computing the mixture weights, $\boldsymbol{\mu}$, used to combine the $p$ Nyström approximations. In general, the cubic term dominates the complexity since the mixture weights can be computed in constant time for the uniform method, in $O(psn)$ for the exponential weight method, or in $O(p^3+pls)$ for the ridge regression method. 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 5.2.3 for further details). In terms of CPU requirements, we note that our 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.

### 4.3.1 Ensemble Woodbury approximation

In Section 3.2, the Woodbury approximation was presented as a useful tool to use alongside low-rank approximations to efficiently (and approximately) invert kernel matrices. Recall that 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:

$$(\mathbf{A}+\mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}, \tag{4.3}$$

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, i.e., $\widetilde{\mathbf{K}}_r$ 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 (4.3) 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 $\log_2(p)$ more time than using the standard Nyström method.

| Dataset | Type of data | # Points ($n$) | # Features ($d$) | Kernel |
|---|---|---|---|---|
| PIE-2.7K | face images | 2731 | 2304 | linear |
| MNIST | digit images | 4000 | 784 | linear |
| ESS | proteins | 4728 | 16 | RBF |
| AB-S | abalones | 4177 | 8 | RBF |
| DEXT | bag of words | 2000 | 20000 | linear |
| SIFT-1M | Image features | 1M | 128 | RBF |

Table 4.4: Description of the datasets used in our ensemble Nyström experiments (Sim et al., 2002; LeCun & Cortes, 1998; Gustafson et al., 2006; Asuncion & Newman, 2007; Lowe, 2004).

### 4.3.2 Experiments

In this section, we present experimental results that illustrate the performance of the ensemble Nyström method. We work with the datasets listed in Table 4.4, and compare the performance of various methods for calculating the mixture weights ($\mu_r$). Throughout our experiments, we measure the accuracy of a low-rank approximation $\widetilde{\mathbf{K}}$ by calculating the relative error in Frobenius and spectral norms, that is, if we let $\xi = \{2, F\}$, then we calculate the following quantity:[3]

$$\%\,\text{error} = \frac{\|\mathbf{K} - \widetilde{\mathbf{K}}\|_\xi}{\|\mathbf{K}\|_\xi} \times 100. \tag{4.4}$$

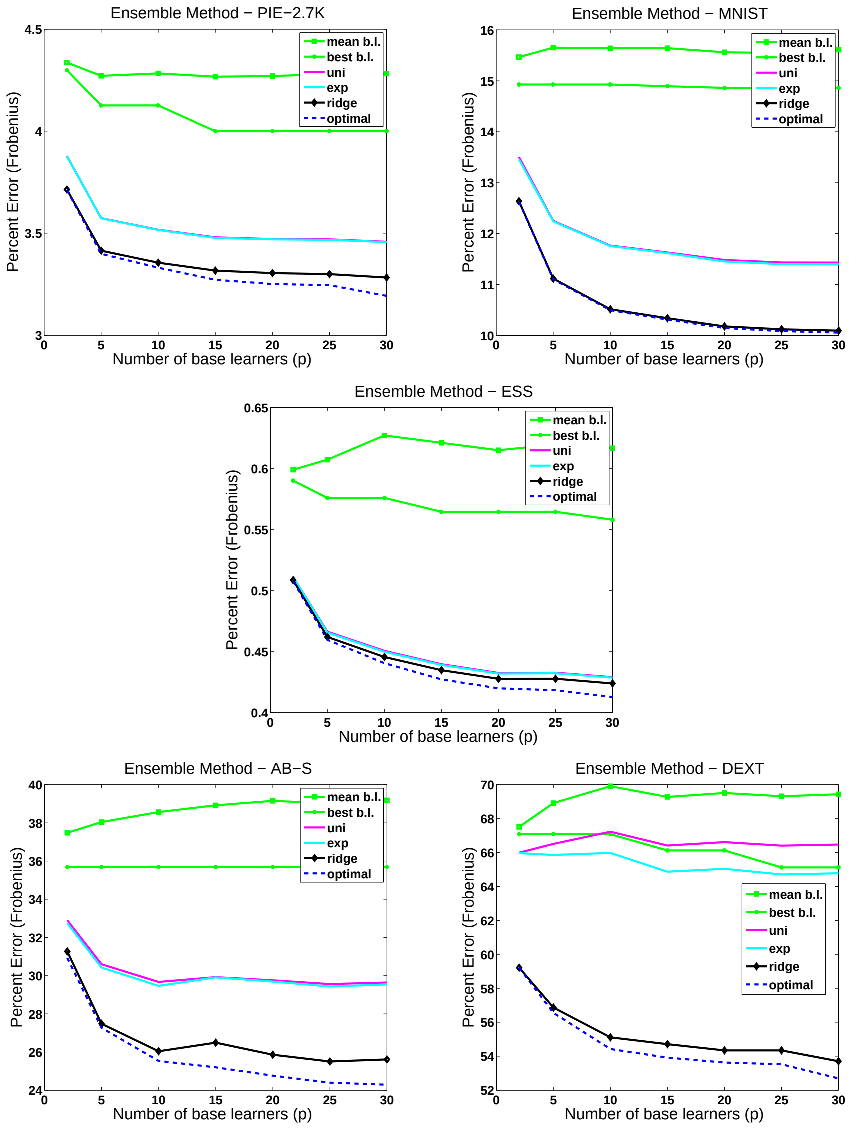


Figure 4.5: Percent error in Frobenius norm for ensemble Nyström method using uniform ('uni'), exponential ('exp'), ridge ('ridge') and optimal ('optimal') mixture weights as well as the best ('best b.l.') and mean ('mean b.l.') of the $p$ base learners used to create the ensemble approximations.

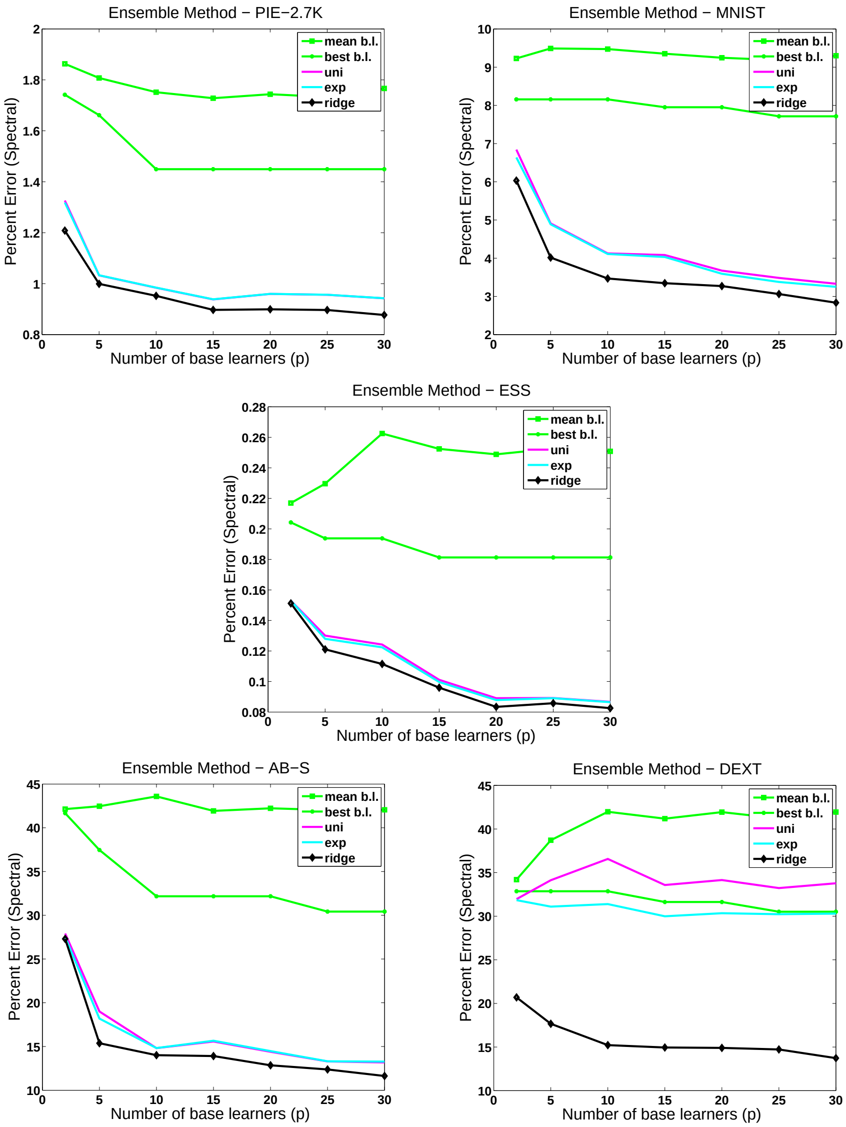
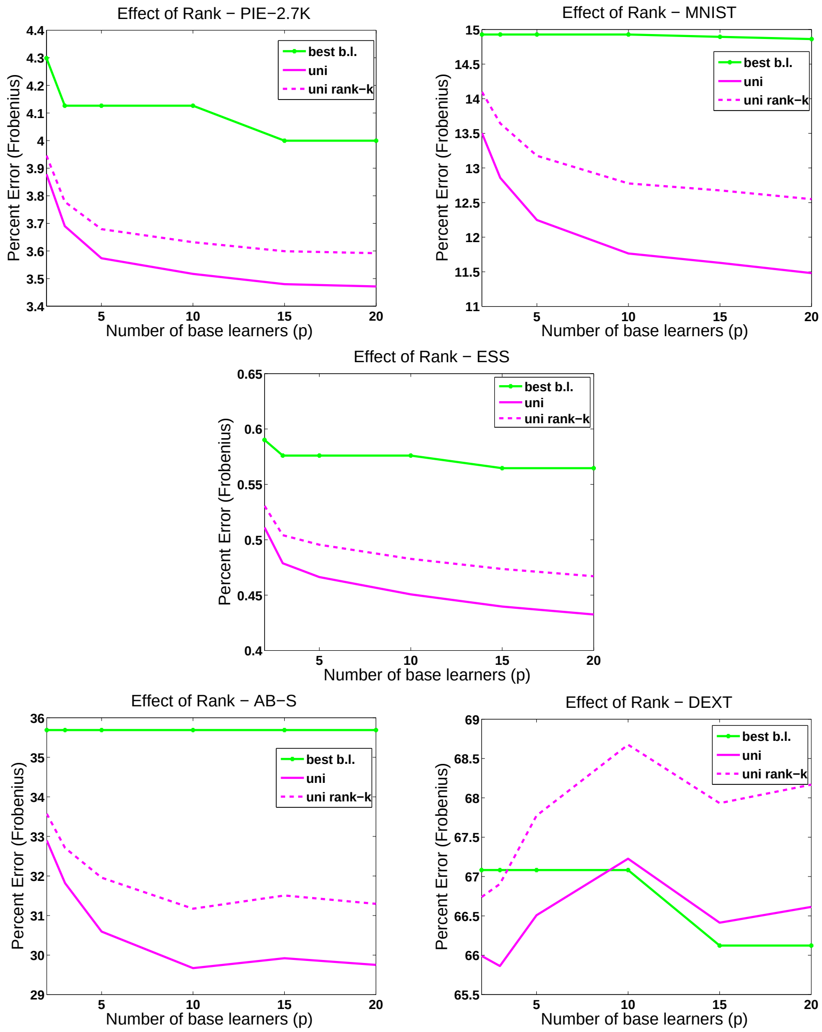


Figure 4.6: Percent error in spectral norm for ensemble Nyström method using various mixture weights and the best/mean of the $p$ approximations. Legend entries are the same as in Figure 4.5.

### Ensemble Nyström with various mixture weights

In this set of experiments, we show results for our ensemble Nyström method using different techniques to choose the mixture weights as previously dis-

[3] Note that we are not using relative accuracy, as in the empirical results presented earlier in this chapter and in Chapter 2, since relative accuracy requires computation of


Effect of Rank − PIE−2.7K

Effect of Rank − MNIST

Effect of Rank − ESS

Effect of Rank − AB-S

Effect of Rank − DEXT

Figure 4.7: Percent error in Frobenius norm 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.

cussed. We first experimented with the first five datasets shown in Table 4.4.

the best low-rank approximation, $\mathbf{K}_k$, which is not possible to compute for our large-scale experiments. However, percentage error and relative accuracy are highly correlated and are both valid measurements for quality of reconstruction.

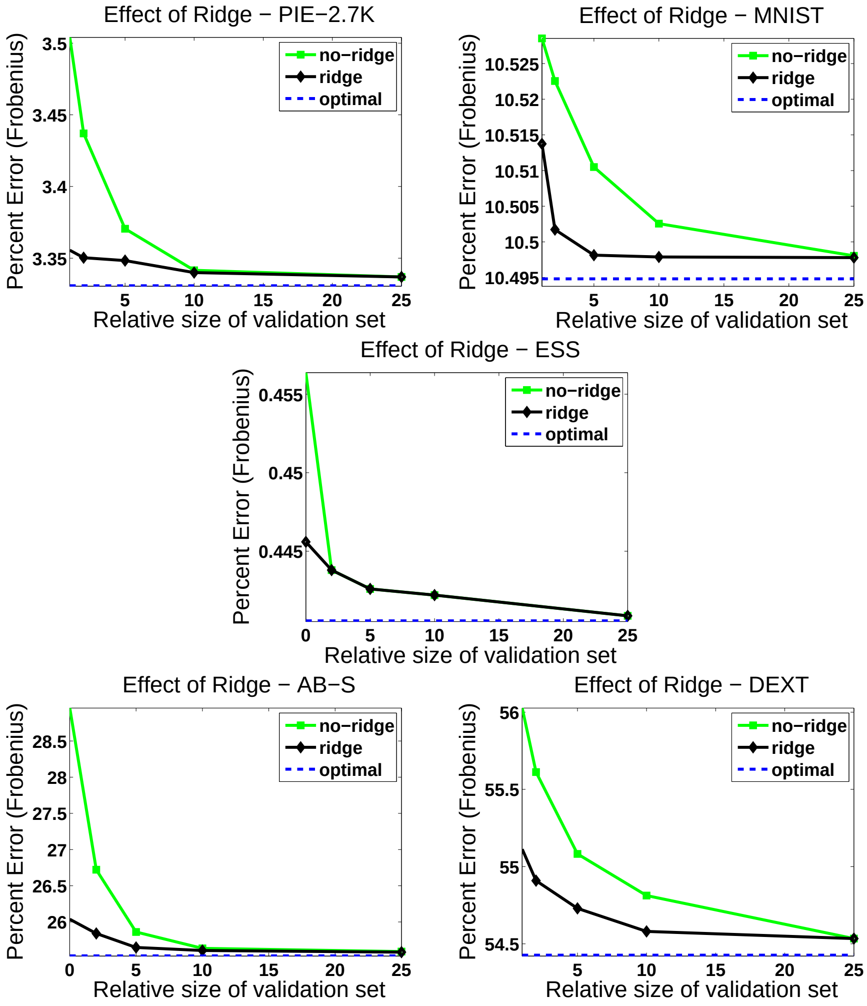


Figure 4.8: Comparison of percent error in Frobenius norm for the ensemble Nyström method with $p = 10$ experts with weights derived from linear ('no-ridge') and ridge ('ridge') regression. The dotted line indicates the optimal combination. The relative size of the validation set equals $s/n \times 100$.

For each dataset, we fixed the reduced rank to $k = 50$, and set the number of sampled columns to $l = 3\% \times n$.[4] Furthermore, for the exponential and the

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

ridge regression variants, we sampled an additional 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 30. As a baseline, we also measured the minimal and mean percent error across the $p$ Nyström approximations used to construct $\widetilde{\mathbf{K}}^{ens}$. For the Frobenius norm, we also calculated the performance when using the optimal $\boldsymbol{\mu}$, 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 4.5 for the Frobenius norm and in Figure 4.6 for the spectral norm. These results clearly show that the ensemble Nyström performance is significantly better than any of the individual Nyström approximations. 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 Figures 4.5 and 4.6, 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). The results of this experiment, presented in Figure 4.7, suggest that the performance gain of the ensemble methods is not due to this increased rank.

Furthermore, the ridge regression technique is the best of the proposed techniques and generates nearly the optimal solution in terms of the percent error in 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 4.8 shows this comparison between linear regression and ridge regression for varying values of $s$ using a fixed number of experts ($p = 10$). Finally we 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.

### Large-scale experiments

We now present an empirical study of the effectiveness of the ensemble Nyström method on the SIFT-1M dataset in Table 2.1 containing 1 *million* data points. As is common practice with large-scale datasets, we worked on a cluster of several machines for this dataset. We present results comparing the performance of the ensemble Nyström method, using both uniform and ridge regression mixture weights, with that of the best and mean performance across the $p$ Nyström approximations used to construct $\widetilde{\mathbf{K}}^{ens}$. We also make comparisons with the $K$-means adaptive sampling technique introduced in Section 4.2 (Zhang et al., 2008; Zhang & Kwok, 2009). Although the $K$-means technique is quite effective at generating informative columns by exploiting the data distribution, the cost of performing $K$-means becomes expensive for even moderately sized datasets, making it difficult to use in large-scale settings. Nevertheless, in this work, we include the $K$-means method in our comparison, and we present results for various subsamples of the SIFT-1M dataset, with $n$ ranging from 5K to 1M.

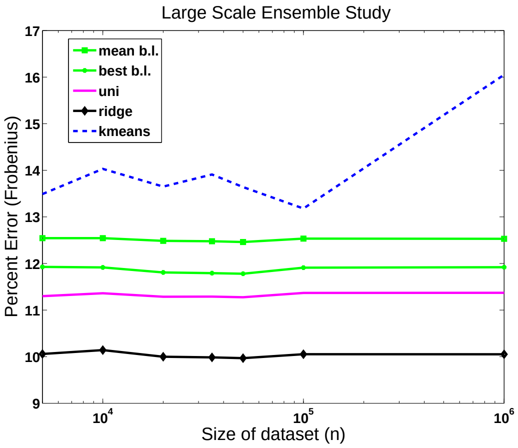


Figure 4.9: Large-scale performance comparison with SIFT-1M dataset. For a fixed computational time, the ensemble Nyström approximation with ridge weights tends to outperform other techniques.

To fairly compare these techniques, we performed 'fixed-time' experiments. We first searched for an appropriate $l$ such that the percent error for the ensemble Nyström method with ridge weights was approximately 10%, and measured the time required by the cluster to construct this approximation. We then allotted an equal amount of time (within 1 second) for the other techniques, and measured the quality of the resulting approximations. For these experiments, we set $k = 50$ and $p = 10$, based on the results from the previous section. Furthermore, in order to speed up computation on this large dataset, we decreased the size of the validation and hold-out sets to $s = 2$ and $s' = 2$, respectively.

The results of this experiment, presented in Figure 4.9, clearly show that the ensemble Nyström method is the most effective technique given a fixed amount of time. Furthermore, even with the small values of $s$ and $s'$, ensemble Nyström with ridge-regression weighting outperforms the uniform ensemble Nyström method. We also observe that due to the high computational cost of $K$-means for large datasets, the $K$-means approximation does not perform well in this 'fixed-time' experiment. It generates an approximation that is worse than the mean standard Nyström approximation and its performance increasingly deteriorates as $n$ approaches 1M. Finally, we note that although the space requirements are 10 times greater for ensemble Nyström in comparison to standard Nyström (since $p = 10$ in this experiment), the space constraints are nonetheless quite reasonable. For instance, when working with 1M points, the ensemble Nyström method with ridge regression weights only required approximately 1% of the columns of $\mathbf{K}$ to achieve a percent error of 10%.

## 4.4 Summary

A key aspect to sampling-based matrix approximations is the manner in which we choose a subset of representative columns. In this chapter we have discussed both fixed and adaptive approaches to sampling columns of a matrix. We have seen that the approximation performance is significantly affected by the choice of sampling algorithm, and furthermore, that there is often a tradeoff between choosing a more informative set of columns and the efficiency of the sampling algorithm. Finally, we discussed an ensemble meta-algorithm for combining multiple matrix approximations that generates favorable matrix reconstructions and yet naturally fits within a distributed computing environment, thus making it quite efficient even in large-scale settings.

# Chapter 5

# Theoretical Analysis

The effectiveness of the Nyström method hinges on two key assumptions on the input matrix, **K**. First, we assume that a low-rank approximation to **K** can be effective for the task at hand. This assumption is often true empirically as demonstrated by the widespread use of Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) in practical applications. As expected, the Nyström method is not appropriate in cases where this assumption does not hold, which explains its poor performance in the experimental results of Fergus et al. (2009). In Section 5.1, we present theoretical results based on work in Kumar et al. (2009c) and Kumar et al. (2009a) that incorporate this low-rank assumption by comparing the quality of the Nyström approximation to the 'best' low-rank approximation, i.e., the approximation constructed from the top singular values and singular vectors of **K**. This work, related to work by Drineas and Mahoney (2005), provides performance bounds for the Nyström method as it is used in practice, i.e., using uniform sampling without replacement, and holds for both the standard Nyström method as well as the ensemble Nyström method discussed in Section 4.3.

The second crucial assumption involves the sampling-based nature of the algorithm, namely that an accurate low-rank approximation can be generated exclusively from information extracted from a small subset of $l \ll n$ columns of $\mathbf{K}$. This assumption is not generally true for all matrices. For instance, consider the extreme case of the $n \times n$ matrix described below:

$$\mathbf{K} = \begin{bmatrix} | & & | & | & & | \\ \mathbf{e}_1 & \ldots & \mathbf{e}_r & \mathbf{0} & \ldots & \mathbf{0} \\ | & & | & | & & | \end{bmatrix}, \tag{5.1}$$

where $\mathbf{e}_i$ is the $i$th column of the $n$ dimensional identity matrix and $\mathbf{0}$ is the $n$ dimensional zero vector. Although this matrix has rank $r$, it nonetheless cannot be well approximated by a random subset of $l$ columns unless this subset includes $\mathbf{e}_1, \ldots, \mathbf{e}_r$. In order to account for such pathological cases, previous theoretical bounds relied on sampling columns of $\mathbf{K}$ from a non-uniform distribution weighted precisely by the magnitude of the diagonal elements of $\mathbf{K}$, as discussed in Section 4.1. Indeed, these bounds give better guarantees for pathological cases. However, in practice, when working with real-world datasets, uniform sampling is more commonly used, e.g., Williams and Seeger (2000); Fowlkes et al. (2004); Platt (2004); Talwalkar et al. (2008), since diagonal sampling is more expensive and does not typically outperform uniform sampling, as discussed in Section 4.1. Hence the diagonal sampling bounds are not applicable in this setting. Furthermore, these bounds are typically loose for matrices in which the diagonal entries of the matrix are roughly of the same magnitude, as in the case of all kernel matrices generated from RBF kernels, for which the Nyström has been noted to work particularly well (Williams & Seeger, 2000). Adaptive techniques have also been proposed to handle these pathological cases, though they are not well studied theoretically and in practice are outperformed by uniform sampling given fixed time constraints, as shown in empirical studies in Zhang et al. (2008) and discussed in Section 4.2.

Hence, we propose to characterize the ability to extract information from a small subset of $l$ columns using the notion of matrix *coherence*, an alternative data-dependent measurement which we believe to be intrinsically related to the algorithm's performance. Recent work on compressed sensing and matrix completion, which also involve sampling-based approximations, have relied heavily on coherence assumptions (Donoho, 2006; Candès et al., 2006; Candès & Romberg, 2007). Coherence measures the extent to which the singular vectors of a matrix are correlated with the standard basis. Intuitively, if we work with sufficiently incoherent matrices, then we avoid pathological cases such as the one presented in (5.1), as we will show theoretically and empirically in Section 5.2. This research based on work in Talwalkar and Rostamizadeh (2010).

Finally, in Section 5.3, we address the question of how kernel approximation affects the performance of learning algorithms. There exists some previous work on this subject. Spectral clustering with perturbed data was studied in a restrictive setting with several assumption by Huang et al. (2008). In Fine and Scheinberg (2002), the authors address this question in terms of the impact on the value of the *objective function* to be optimized by the learning algorithm. However, we strive to take the question one step further and directly analyze the effect of an approximation in the kernel matrix on the *hypothesis* generated by several widely used kernel-based learning algorithms. Based on initial work in Cortes et al. (2010), we give stability bounds based on the norm of the kernel approximation for these algorithms, including SVMs, SVR, KRR, Kernel PCA and graph Laplacian-based regularization algorithms (Belkin et al., 2004). These bounds help determine the degree of approximation that can be tolerated in the estimation of the kernel matrix. Our analysis differs from previous applications of stability analysis as put forward by Bousquet and Elisseeff (2001). Instead of studying the effect of changing one training point, we study the effect of changing the kernel matrix. Our analysis is general and applies to arbitrary approximations of the kernel matrix. However, we also give a specific analysis of the Nyström low-rank approximation given the recent interest in this method and the successful applications of this algorithm to large-scale applications. We also discuss experimental results for Kernel Ridge Regression that support our theoretical analyses.

## 5.1 Nyström Analysis

We now present a theoretical analysis of the Nyström method based on work from Kumar et al. (2009a) for which we use as tools some results previously shown by Drineas and Mahoney (2005) and Kumar et al. (2009c). As in Kumar et al. (2009c), we shall use the following generalization of McDiarmid's concentration bound to sampling without replacement (Cortes et al., 2008).

**Theorem 5.1** *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 $i \in [1, l]$ 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 $\epsilon > 0$, the following inequality holds:*

$$\Pr\left[\phi - \mathbf{E}[\phi] \geq \epsilon\right] \leq \exp\left[\tfrac{-2\epsilon^2}{\alpha(l,u)c^2}\right], \tag{5.2}$$

*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 $\max_{ij} \sqrt{\mathbf{K}_{ii} + \mathbf{K}_{jj} - 2\mathbf{K}_{ij}}$.

### 5.1.1 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 5.2** *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 $l$:*

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

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

*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]:

$$\|\mathbf{K} - \widetilde{\mathbf{K}}\|_2 \leq \|\mathbf{K} - \mathbf{K}_k\|_2 + 2\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_2, \tag{5.3}$$

where $\mathbf{Z} = \sqrt{\frac{n}{l}}\,\mathbf{X}\mathbf{S}$. We then apply the McDiarmid-type inequality of Theorem 5.1 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 $\sqrt{\frac{n}{l}}\,\mathbf{X}\mathbf{S}'$. Let $\mathbf{z}$ and $\mathbf{z}'$ denote the only differing columns of $\mathbf{Z}$ and $\mathbf{Z}'$, then

$$|\phi(\mathbf{S}') - \phi(\mathbf{S})| \leq \|\mathbf{z}'\mathbf{z}'^\top - \mathbf{z}\mathbf{z}^\top\|_2 = \|(\mathbf{z}' - \mathbf{z})\mathbf{z}'^\top + \mathbf{z}(\mathbf{z}' - \mathbf{z})^\top\|_2 \tag{5.4}$$

$$\leq 2\|\mathbf{z}' - \mathbf{z}\|_2 \max\{\|\mathbf{z}\|_2, \|\mathbf{z}'\|_2\}. \tag{5.5}$$

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 $\mathbf{K}_{\max}^{\frac{1}{2}}$. This leads to the following inequality:

$$|\phi(\mathbf{S}') - \phi(\mathbf{S})| \leq \frac{2n}{l} d_{\max}^{\mathbf{K}} \mathbf{K}_{\max}^{\frac{1}{2}}. \tag{5.6}$$

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

$$\mathbf{E}[\Phi] = \mathbf{E}[\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_2] \leq \mathbf{E}[\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_F] \leq \frac{n}{\sqrt{l}}\mathbf{K}_{\max}, \tag{5.7}$$

where the last inequality follows Corollary 2 of Kumar et al. (2009c). The inequalities (5.6) and (5.7) combined with Theorem 5.1 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 & Mahoney, 2005):

$$\|\mathbf{K}-\widetilde{\mathbf{K}}\|_F^2 \le \|\mathbf{K}-\mathbf{K}_k\|_F^2 + \sqrt{64k}\,\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_F^2\, n\mathbf{K}_{ii}^{\max}. \tag{5.8}$$

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 5.1 yields the result of the theorem. □

### 5.1.2 Ensemble Nyström method

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

**Theorem 5.3** *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 $\boldsymbol{\mu}$ in the simplex $\Delta$ and $\widetilde{\mathbf{K}}^{ens} = \sum_{r=1}^{p} \mu_r \widetilde{\mathbf{K}}_r$:*

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

*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 5.2, the following holds:

$$\|\mathbf{K} - \widetilde{\mathbf{K}}^{ens}\|_2 = \Big\| \sum_{r=1}^{p} \mu_r(\mathbf{K} - \widetilde{\mathbf{K}}_r) \Big\|_2 \leq \sum_{r=1}^{p} \mu_r \|\mathbf{K} - \widetilde{\mathbf{K}}_r\|_2 \tag{5.9}$$

$$\leq \sum_{r=1}^{p} \mu_r \big( \|\mathbf{K} - \mathbf{K}_k\|_2 + 2\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}_r\mathbf{Z}_r^\top\|_2 \big) \tag{5.10}$$

$$= \|\mathbf{K} - \mathbf{K}_k\|_2 + 2\sum_{r=1}^{p} \mu_r \|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}_r\mathbf{Z}_r^\top\|_2. \tag{5.11}$$

We apply Theorem 5.1 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 (5.6) on the change to $\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}_r\mathbf{Z}_r^\top\|_2$, the following holds:

$$|\phi(S') - \phi(S)| \leq \frac{2n}{l} \mu_{\max} d_{\max}^{\mathbf{K}} \mathbf{K}_{\max}^{\frac{1}{2}}, \tag{5.12}$$

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 (5.7) for a single expert. Plugging in this upper bound and the Lipschitz bound (5.12) in Theorem 5.1 yields our 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 (5.8), we can write

$$\begin{aligned}\|\mathbf{K}-\widetilde{\mathbf{K}}^{ens}\|_F^2 &= \Big\|\sum_{r=1}^{p}\mu_r(\mathbf{K}-\widetilde{\mathbf{K}}_r)\Big\|_F^2 \leq \sum_{r=1}^{p}\mu_r\|\mathbf{K}-\widetilde{\mathbf{K}}_r\|_F^2 &(5.13)\\ &\leq \sum_{r=1}^{p}\mu_r\Big[\|\mathbf{K}-\mathbf{K}_k\|_F^2+\sqrt{64k}\,\|\mathbf{X}\mathbf{X}^\top-\mathbf{Z}_r\mathbf{Z}_r^\top\|_F\, n\mathbf{K}_{ii}^{\max}\Big]. &(5.14)\\ &= \|\mathbf{K}-\mathbf{K}_k\|_F^2+\sqrt{64k}\,\sum_{r=1}^{p}\mu_r\|\mathbf{X}\mathbf{X}^\top-\mathbf{Z}_r\mathbf{Z}_r^\top\|_F\, n\mathbf{K}_{ii}^{\max}. &(5.15)\end{aligned}$$

The result follows by the application of Theorem 5.1 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 5.3 are similar in form to those of Theorem 5.2. 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 $\mu_{\max}p^{\frac{1}{2}} = 1/\sqrt{p}$.

## 5.2 Coherence-based Bounds

The main contribution of this section is the connection made between matrix coherence and the Nyström method. Making use of related work in the compressed sensing and the matrix completion literature, we give a more refined analysis of the Nyström method as a function of matrix coherence. We also present extensive empirical results that strongly relate coherence to the performance of the Nyström method. The results in this section are based on work from Talwalkar and Rostamizadeh (2010).

### 5.2.1 Coherence

Although the Nyström method tends to work well in practice, the performance of this algorithm depends on the structure of the underlying matrix. We will show that the performance is related to the size of the entries of the singular vectors of $\mathbf{K}$, or the *coherence* of its singular vectors. We define $\mathbf{U}_r$ as the top $r$ singular vectors of $\mathbf{K}$, and denote the coherence of these singular vectors as $\mu(\mathbf{U}_r)$, which is adapted from Candès and Romberg (2007).

**Definition 5.1 (Coherence)** *The* coherence *of a matrix* $\mathbf{U}_r$ *with orthonormal columns is defined as:*

$$\mu(\mathbf{U}_r) = \sqrt{n} \max_{i,j} |{\mathbf{U}_r}_{(i)}^{(j)}| \ . \tag{5.16}$$

The coherence of $\mathbf{U}_r$ is lower bounded by 1, as is the case for the rank-1 matrix with all entries equal to $1/\sqrt{n}$, and upper bounded by $\sqrt{n}$, as is the case for the matrix of canonical basis vectors. As discussed in Candès and Recht (2009); Candès and Tao (2009), highly coherent matrices are difficult to randomly recover via matrix completion algorithms, and this same logic extends to the Nyström method. In contrast, incoherent matrices are much easier to successfully complete and to approximate via the Nyström method, as discussed in Section 5.2.2.

In order to provide some intuition, Candès and Recht (2009) give several classes of randomly generated matrices with low coherence. One such class of matrices is generated from uniform random orthonormal singular vectors and arbitrary singular values. For such a class they show that $\mu = O(\sqrt{\log n}\sqrt[4]{r})$ with high probability.[1] In what follows, we will show bounds on the number of points needed for reconstruction that become more favorable as coherence decreases. However, the bounds are useful for more generous values of coherence than given in the above example. We will also provide an empirical study of coherence for various real-world and synthetic examples.

### 5.2.2 Low-rank, low-coherence bounds

In this section, we make use of coherence to analyze the Nyström method when used with low-rank matrices. We note that although the bounds pre-

[1] For low-rank matrices, $\sqrt[4]{r}$ is quite small. Moreover, this $\sqrt[4]{r}$ factor only appears due to our use of the generally loose inequality $\mu^2 \leq \sqrt{r}\mu_1$, where $\mu_1$ is a slightly different notion of coherence used in the original bound in Candès and Recht (2009) for this class of matrices.

sented throughout this section hold for matrices of any rank $r$, they are only interesting when $r = o(\sqrt{n})$, and hence they are most applicable in the "low-rank" setting.

As discussed in Section 2.2, the Nyström method generates high quality low-rank approximations in cases where $\mathbf{K}$ has low-rank structure even if the matrix has full rank, i.e., $\mathbf{K} \approx \mathbf{K}_k$ for some $k \ll n$. Furthermore, as stated in Theorem 2.3, when $\mathbf{K}$ is actually a low-rank matrix, then the Nyström method can exactly recover the initial matrix. This theorem implies that if $\mathbf{K}$ has low-rank and $l \geq k \geq r$, then *there exists* a particular sampling such that $\text{rank}(\mathbf{W}) = \text{rank}(\mathbf{K})$ and the Nyström method can perfectly recover the full matrix. However, selecting a suitable set of $l$ columns from an $n \times n$ SPSD matrix can be an intractable combinatorial problem, and there exist matrices for which the probability of selecting such a subset uniformly at random is exponentially small, e.g., the rank-$r$ SPSD diagonal matrices discussed earlier. In contrast, a large class of SPSD matrices are much more incoherent, and for these matrices, we will next show that by choosing $l$ to be linear in $r$ and logarithmic in $n$ we can with very high probability guarantee that $\text{rank}(\mathbf{W}) = r$, and hence exactly recover the initial matrix.

### Probability of choosing a good subset

We start with a rank-$r$ Gram matrix, $\mathbf{K}$, and a fixed distribution, $\mathcal{D}$, over the columns of $\mathbf{K}$. Our goal is to calculate the probability of randomly choosing a subset of $l$ columns of $\mathbf{K}$ according to $\mathcal{D}$ such that $\text{rank}(\mathbf{W}) = r$. Recall that

$\mathbf{K} = \mathbf{X}^\top\mathbf{X}$, $\mathbf{X} = [\mathbf{X}_1 \quad \mathbf{X}_2]$ and $\mathbf{W} = \mathbf{X}_1^\top\mathbf{X}_1$. Then, by properties of SVD, we know that $\text{rank}(\mathbf{K}) = \text{rank}(\mathbf{X})$ and $\text{rank}(\mathbf{W}) = \text{rank}(\mathbf{X}_1)$. Hence, the probability of this desired event is equivalent to the probability of sampling $l$ columns of $\mathbf{X}$ according to $\mathcal{D}$ such that $\text{rank}(\mathbf{X}_1) = r$, as shown in (5.18). Next, we can write the thin SVD of $\mathbf{X}$ as $\mathbf{X} = \mathbf{U}_X\mathbf{\Sigma}_X\mathbf{V}_X^\top$, where $\mathbf{U}_X \in \mathbb{R}^{N\times r}$, $\mathbf{\Sigma}_X \in \mathbb{R}^{r\times r}$ and $\mathbf{V}_X \in \mathbb{R}^{n\times r}$. Since $\mathbf{U}_X$ contains orthonormal columns and $\mathbf{\Sigma}_X$ is invertible, we know that $\mathbf{\Sigma}_X^{-1}\mathbf{U}_X^\top\mathbf{X} = \mathbf{V}_X^\top$. Further, using the block representation of $\mathbf{X}$, we have

$$\mathbf{X}_1^\top\mathbf{U}_X\mathbf{\Sigma}_X^{-1} = \mathbf{V}_{X\,(1:l)}, \tag{5.17}$$

where $\mathbf{V}_{X\,(1:l)} \in \mathbb{R}^{l\times r}$ corresponds to the first $l$ rows of $\mathbf{V}_X$, i.e., the first $l$ components for each of the $r$ right singular vectors of $\mathbf{X}$. Since the left singular vectors of $\mathbf{X}$ span the columns $\mathbf{X}$ and hence of $\mathbf{X}_1$, we know that $\text{rank}(\mathbf{X}_1) = \text{rank}(\mathbf{X}_1^\top\mathbf{U}_X\mathbf{\Sigma}_X^{-1})$ and we obtain the equality of (5.19).

$$\begin{aligned}\Pr_{\mathcal{D}}[\text{rank}(\mathbf{W}) = r] &= \Pr_{\mathcal{D}}[\text{rank}(\mathbf{X}_1) = r] && (5.18)\\ &= \Pr_{\mathcal{D}}[\text{rank}(\mathbf{V}_{X\,(1:l)}) = r]. && (5.19)\end{aligned}$$

In the next section we calculate this probability for a specific distribution in terms of $l$ as well as a measure of the coherence of $\mathbf{V}_X$.

| Dataset | Type of data | # Points ($n$) | # Features ($d$) | Kernel |
|---|---|---|---|---|
| PIE | face images | 2731 | 2304 | linear |
| MNIST | digit images | 4000 | 784 | linear |
| Essential | proteins | 4728 | 16 | RBF |
| Abalone | abalones | 4177 | 8 | RBF |
| Dexter | bag of words | 2000 | 20000 | linear |
| Artificial | random features | 1000 | 20000 | linear |

Table 5.1: Description of the datasets used in our coherence experiments, including the type of data, the number of points ($n$), the number of features ($d$) and the choice of kernel (Sim et al., 2002; LeCun & Cortes, 1998; Gustafson et al., 2006; Asuncion & Newman, 2007).

### Sampling bound

Given the orthonormal matrix $\mathbf{U}_r$, we would like to find a choice of $l$ such that $\mathbf{V}_{X(1:l)}$ created by *uniform* sampling has rank $r$ with high probability. As pointed out in the previous section, a meaningful bound may not be possible for any $l < n$ if no assumption is made on $\mathbf{V}_r$. Here we adopt the assumption that $\mathbf{V}_r$ has low coherence, as defined in Definition 5.1. If we define $\mathbf{A} = \mathbf{V}_{X(1:l)}^\top \mathbf{V}_{X(1:l)}$, we then observe that by properties of SVD we have

$$\Pr\Big(\operatorname{rank}(\mathbf{V}_{X(1:l)}) = r\Big) = \Pr\Big(\operatorname{rank}(\mathbf{A}) = r\Big). \tag{5.20}$$

Next, we define $\sigma = \|\mathbf{A}\|_2$ and note that for $0 < c < 1/\sigma$, $c\mathbf{A}$ is an $r \times r$ SPSD matrix with singular values less than one. Furthermore, $\mathbf{I} - c\mathbf{A}$ is also SPSD with

$$\Pr\Big(\operatorname{rank}(\mathbf{A}) = r\Big) = \Pr\Big(\|c\mathbf{A} - \mathbf{I}\|_2 < 1\Big), \tag{5.21}$$

since $\|c\mathbf{A} - \mathbf{I}\| = 1$ implies that the nullspace of $\mathbf{A}$ is nonempty. Alternatively, if $c \geq 1/\sigma$, then

$$\Pr\Big(\text{rank}(\mathbf{A}) = r\Big) \geq \Pr\Big(\|c\mathbf{A} - \mathbf{I}\|_2 < 1\Big), \tag{5.22}$$

since, for large enough $c$, we could have $\|c\mathbf{A} - \mathbf{I}\|_2 \geq 1$ even if $\text{rank}(\mathbf{A}) = r$. Thus the inequality in (5.22) holds for any constant $c > 0$, i.e., the probability on the RHS of (5.22) serves as a lower bound to the probability of interest.

The probability on the RHS of (5.22) has been studied in previous compressive sampling literature. Specifically, Candès and Romberg (2007) makes use of a main lemma of Rudelson (1999) to derive Theorem 5.4, which provides us with our desired lower bound.

**Theorem 5.4 ((Candès & Romberg, 2007) Thm 1.2)** *Define* $\mathbf{V} \in \mathbb{R}^{n\times r}$ *such that* $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$ *and let* $\mathbf{V}_{(1:l)} \in \mathbb{R}^{l\times r}$ *be generated from* $\mathbf{V}$ *by sampling* $l$ *rows uniformly at random. Then, the following holds with probability at least* $1-\delta$*,*

$$\Big\|\frac{n}{l}\mathbf{V}_{(1:l)}^\top\mathbf{V}_{(1:l)} - \mathbf{I}\Big\| < \frac{1}{2}, \tag{5.23}$$

*for any* $l$ *that satisfies,*

$$l \geq r\mu^2(\mathbf{V})\max\big(C_1\log(r), C_2\log(3/\delta)\big), \tag{5.24}$$

*where* $C_1$ *and* $C_2$ *are positive constants.*

Note that our definition of coherence and statement of Theorem 5.4 are modified to account for the fact that $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$ as oppose to $n\mathbf{I}$, as in Candès and Romberg (2007). Also, $\mathbf{V}$ is not square as assumed in the original theorem, however it can be verified that the proof holds even for this case.

By making use of Theorem 5.4, we can now answer the question regarding the number of columns needed to sample from $\mathbf{K}$ in order to obtain an exact reconstruction via the Nyström method. Theorem 5.5 presents a bound on $l$ for matrix completion in terms of $\mu$.

**Theorem 5.5** *Let* $\mathbf{K} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top \in \mathbb{R}^{n \times n}$ *be a rank-r SPSD matrix, where* $(\mathbf{\Sigma}, \mathbf{U})$ *are matrices of its singular values and singular vectors. Then it suffices to sample* $l \geq r\mu^2(\mathbf{V}) \max\big(C_1 \log(r), C_2 \log(3/\delta)\big)$ *columns, where* $C_1$ *and* $C_2$ *are positive constants, to have with probability at least* $1 - \delta$*,*

$$\|\mathbf{K} - \widetilde{\mathbf{K}}_k\| = 0\,. \tag{5.25}$$

*Proof.* Theorem 2.3 states sufficient conditions for exact matrix completion. Equations (5.18) and (5.19) reduce these sufficient conditions to a condition on the rank of $\mathbf{V}_{X(1:l)}$. Equations (5.20) and (5.22) further reduce this problem to a similar problem previously studied in the context of compressed sensing. Finally, we use Theorem 5.4 to bound with high probability the RHS of (5.22).
□

### 5.2.3 Experiments

In this section we present a series of empirical results that show the empirical connection between matrix coherence and the performance of the Nyström method. We first perform experiments that corroborate the theoretical claims made in the previous section. We work with the six datasets detailed in Table 5.1 to illustrate the performance of the Nyström method for low-rank matrices, and then interpret these results in the context of the coherence of these datasets. We then present more general experimental results that connect matrix coherence to the Nyström method in the case of full rank matrices.

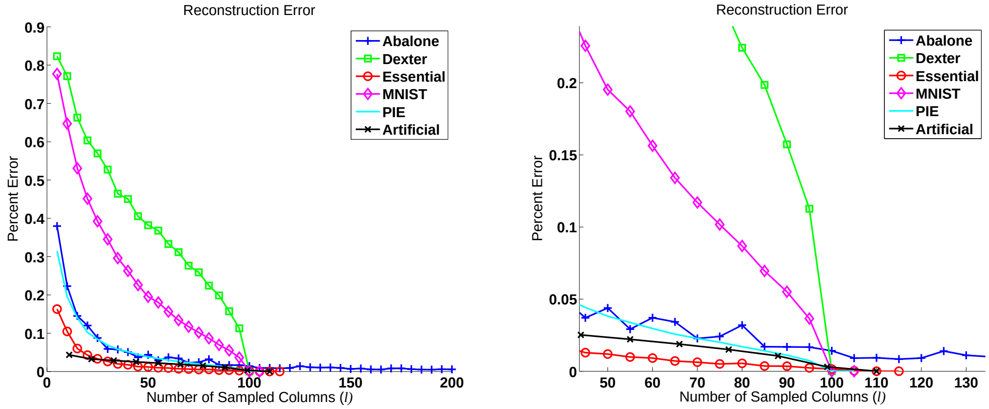


Figure 5.1: Mean percent error over 10 trials of Nyström approximations of rank 100 matrices. Left: Results for $l$ ranging from 5 to 200. Right: Detailed view of experimental results for $l$ ranging from 50 to 130.

### Reconstruction

In our first set of experiments we measure the accuracy of the Nyström approximation ($\widetilde{\mathbf{K}}_k$) for a variety of rank-$r$ matrices, with $r = 100$. For each of the six datasets listed above, we first constructed the optimal rank-$r$ approximation to each kernel matrix by reconstructing with the top $r$ singular values and singular vectors. Next, we performed the Nyström method for various values of $l$ to generate a series of approximations to our rank-$r$ matrix (note that we set $k = l$). For each approximation, we calculated the percent error of the Nyström approximation using the notion of percent error, which we have previously defined as follows:

$$\text{Percent error} = \frac{\|\mathbf{K} - \widetilde{\mathbf{K}}_k\|_F}{\|\mathbf{K}\|_F} \times 100.$$

The results of this experiment, averaged over 10 trials, are presented in Figure 5.1. The figure shows that for five of the six datasets, the Nyström method exactly reconstructs the initial rank $r$ matrix when the number of sampled columns ($l$) is equal or slightly larger than $r$. Note that this observation holds for each of the ten trials, since the mean error is zero for each of these datasets when $l \approx r$. In contrast, for the case of the Abalone dataset, we do not see convergence to zero percent error as $l$ surpasses $r$, and the percent error is non-zero even when $l = 2r$.

**Coherence of datasets**

In this set of experiments, we use the concept of coherence to explain these low-rank reconstruction results, namely that the Nyström method generates an exact matrix reconstruction for $l \approx r$ for five of the six datasets, but fails to do so for the Abalone dataset. As such, we first calculated the coherence of each of the six SPSD rank 100 matrices used in these experiments, using the definition of coherence from Definition 5.1. The left panel of Figure 5.2

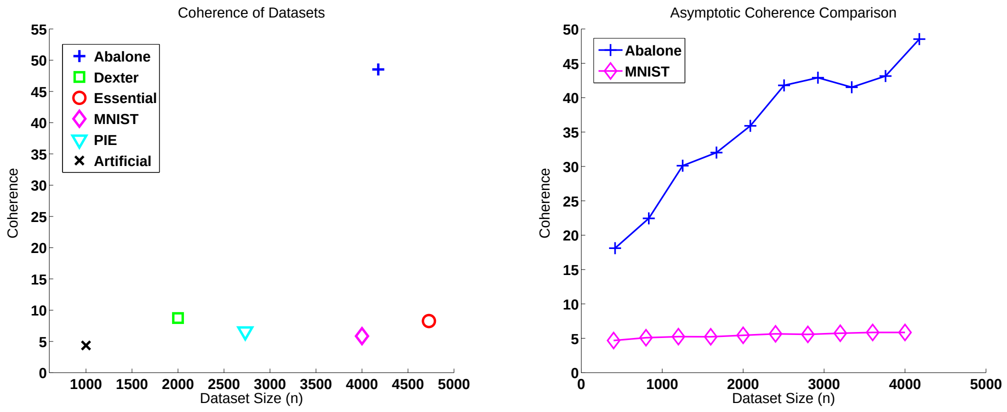


Figure 5.2: Coherence of Datasets. Left: Coherence of rank 100 SPSD matrices derived from datasets listed in Table 5.1. Right: Asymptotic growth of coherence for MNIST and Abalone datasets. Note that coherence values are means over ten trials.

shows the coherence of these matrices with respect to the number of points in the dataset. This plot illustrates the stark contrast between Abalone and the other five datasets in terms of coherence, and helps validate our theoretical connection between low-coherence matrices and the ability to generate exact reconstructions via the Nyström method.

Next, we performed an experiment in which we repeatedly subsampled the initial SPSD matrices to generate matrices with different dimensions, i.e., different values of $n$. For each value of $n$, we computed the coherence of the subsampled matrix, again using Definition 5.1. The right panel of Figure 5.2 shows the mean results over ten trials for both the MNIST and Abalone datasets. As illustrated by this plot, the coherence of the Abalone dataset grows much more quickly than that of the MNIST dataset. As illustrated by the orthogonal random model, we expect incoherent matrices to exhibit a slow rate of growth, i.e. $O(\sqrt{\log n}\sqrt[4]{r})$. The plots for the other four datasets (not shown) are comparable to the MNIST dataset. These results provide further intuition for why the Nyström method is able to perform exact reconstruction on all datasets except for Abalone.

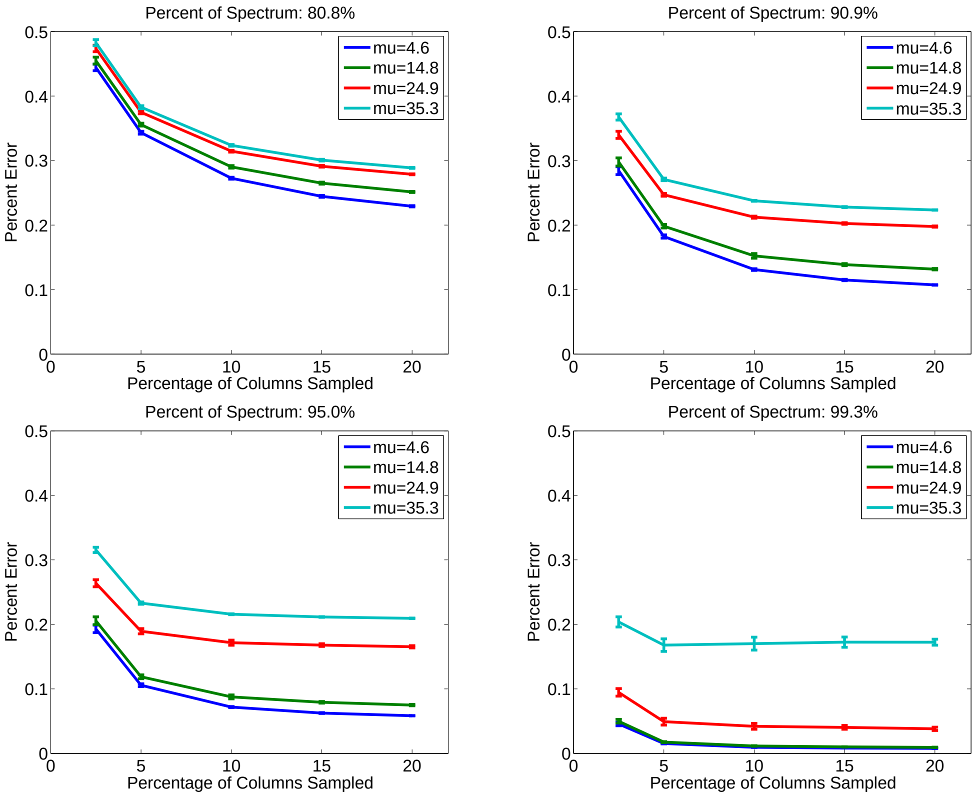


Figure 5.3: Coherence experiments with full rank synthetic datasets. Each plot corresponds to matrices with a fixed singular value decay rate (resulting in a fixed percentage of spectrum captured) and each line within a plot corresponds to the average results of 10 randomly generated matrices with the specified coherence.

### Full rank experiments

As previously discussed, the Nyström method hinges on two assumptions: good low-rank structure of the matrix and the ability to extract information from a small subset of $l$ columns of the input matrix. In this section, we analyze the effect of each of these assumptions on Nyström method performance on full-rank matrices, using matrix coherence as a quantification of the latter assumption. To do so, we devised a series of experiments using synthetic datasets that precisely control the effects of each of these parameters.

To control the low-rank structure of the matrix, we generated artificial datasets with exponentially decaying singular values with differing decay rates, i.e., for $i \in \{1, \ldots, n\}$ we defined the $i$th singular value as $\sigma_i = \exp(-i\eta)$, where $\eta$ controls the rate of decay. For a fixed value of $\eta$, we then measured the percentage of the spectrum captured by the top $k$ singular values as follows:

$$\text{Percent of Spectrum} = \frac{\sum_{i=1}^{k} \sigma_i}{\sum_{i=1}^{n} \sigma_i}. \tag{5.26}$$

To control coherence, we generated singular vectors with varying coherences by forcing the first singular vector to achieve our desired coherence and then using QR to generate a full orthogonal basis. The smallest values of $\mu$ used in our experiments correspond to randomly generated orthogonal matrices. We report the results of our experiments in Figure 5.3. For these experiments we set $n = 2000$ and $k = 50$. Each plot corresponds to matrices with a fixed singular value decay rate (resulting in a fixed percentage of spectrum captured) and each line within a plot corresponds to the average results of 10 randomly generated matrices with the specified coherence. Furthermore, results for each such matrix for a fixed percentage of sampled columns are the means over 5 random subsets of columns.

There are two main observations to be drawn from our experiments. First, as noted in previous work with the Nyström method, the Nyström method generates better approximations for matrices with better low-rank structure, i.e., matrices with a higher percentage of spectrum captured by the top $k$ singular values. Second, following the same pattern as in the low-rank setting, the Nyström method generates better approximations for lower coherence matrices, and hence, matrix coherence appears to effectively capture the degree to which information can be extracted from a subset of columns.

## 5.3 Kernel Stability

Up to this point, we have focused on reconstruction performance while analyzing the Nyström method. In this section we take a different approach, as we analyze the impact of kernel approximation on several common kernel-based learning algorithms: KRR, SVM, SVR, Kernel PCA and graph Laplacian-based regularization algorithms. Our stability analyses result in bounds on the hypotheses directly in terms of the quality of the kernel approximation. Some of the results in this section are based on work in Cortes et al. (2010).

In our analysis we assume that the kernel approximation is only used during training where the kernel approximation may serve to reduce resource requirements. At testing time the true kernel function is used. This scenario that we are considering is standard for the Nyström method and other approximations. We consider the standard supervised learning setting where the learning algorithm receives a sample of $n$ labeled points $S = ((x_1, y_1), \ldots, (x_n, y_n)) \in (X \times Y)^n$, where $X$ is the input space and $Y$ the set of labels, $Y = \mathbb{R}$ with $|y| \leq M$ in the regression case, and $Y = \{-1, +1\}$ in the classification case. We will also assume that $\mathbf{K}'$ is a symmetric, positive, and semi-definite (SPSD) approximation of the SPSD kernel matrix $\mathbf{K}$.[2] Hence, $\mathbf{K}$ and $\mathbf{K}'$ correspond to positive definite symmetric kernel functions $K(\cdot, \cdot)$ and $K'(\cdot, \cdot)$, respectively.

### 5.3.1 Kernel Ridge Regression

We first provide a stability analysis of Kernel Ridge Regression. The following is the dual optimization problem solved by KRR (Saunders et al., 1998):

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} + \boldsymbol{\alpha} \mathbf{K} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^\top \mathbf{y}, \tag{5.27}$$

where $\lambda = n\lambda_0 > 0$ is the ridge parameter. The problem admits the closed form solution $\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y}$. We denote by $h$ the hypothesis returned by Kernel Ridge Regression when using the exact kernel matrix.

**Proposition 5.1** *Let $h'$ denote the hypothesis returned by Kernel Ridge Re-*

[2] Note that we are using the notation $\mathbf{K}'$ to indicate an arbitrary SPSD approximation of $\mathbf{K}$. In contrast, throughout this thesis $\widetilde{\mathbf{K}}$ has referred to SPSD approximations generated from sampling-based algorithms.

*gression when using the approximate kernel matrix* $\mathbf{K}' \in \mathbb{R}^{n\times n}$*. Furthermore, define* $\kappa > 0$ *such that* $K(x,x) \leq \kappa$ *and* $K'(x,x) \leq \kappa$ *for all* $x \in X$*. This condition is verified with* $\kappa = 1$ *for Gaussian kernels for example. Then the following inequalities hold for all* $x \in X$*,*

$$|h'(x) - h(x)| \leq \frac{\kappa M}{\lambda_0^2 n}\|\mathbf{K}' - \mathbf{K}\|_2. \tag{5.28}$$

*Proof.* Let $\boldsymbol{\alpha}'$ denote the solution obtained using the approximate kernel matrix $\mathbf{K}'$. We can write

$$\begin{aligned} \boldsymbol{\alpha}' - \boldsymbol{\alpha} &= (\mathbf{K}' + \lambda\mathbf{I})^{-1}\mathbf{y} - (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y} & (5.29)\\ &= -\big[(\mathbf{K}' + \lambda\mathbf{I})^{-1}(\mathbf{K}' - \mathbf{K})(\mathbf{K} + \lambda\mathbf{I})^{-1}\big]\mathbf{y}, & (5.30) \end{aligned}$$

where we used the identity $\mathbf{M}'^{-1} - \mathbf{M}^{-1} = -\mathbf{M}'^{-1}(\mathbf{M}' - \mathbf{M})\mathbf{M}^{-1}$ valid for any invertible matrices $\mathbf{M}, \mathbf{M}'$. Thus, $\|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\|$ can be bounded as follows:

$$\begin{aligned} \|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\| &\leq \|(\mathbf{K}' + \lambda\mathbf{I})^{-1}\|_2\|\mathbf{K}' - \mathbf{K}\|_2\|(\mathbf{K} + \lambda\mathbf{I})^{-1}\|_2\|\mathbf{y}\| \\ &\leq \frac{\|\mathbf{K}' - \mathbf{K}\|_2\,\|\mathbf{y}\|}{\lambda_{\min}(\mathbf{K}' + \lambda\mathbf{I})\lambda_{\min}(\mathbf{K} + \lambda\mathbf{I})}, \end{aligned} \tag{5.31}$$

where $\lambda_{\min}(\mathbf{K}' + \lambda\mathbf{I})$ is the smallest singular value of $\mathbf{K}' + \lambda\mathbf{I}$ and $\lambda_{\min}(\mathbf{K} + \lambda\mathbf{I})$ the smallest singular value of $\mathbf{K} + \lambda\mathbf{I}$. The hypothesis $h$ derived with the exact kernel matrix is defined by $h(x) = \sum_{i=1}^n \alpha_i K(x, x_i) = \boldsymbol{\alpha}^\top \mathbf{k}_x$, where $\mathbf{k}_x = (K(x, x_1), \ldots, K(x, x_n))^\top$. By assumption, no approximation affects $\mathbf{k}_x$, thus the approximate hypothesis $h'$ is given by $h'(x) = \boldsymbol{\alpha}'^{\top} \mathbf{k}_x$ and

$$|h'(x) - h(x)| \leq \|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\| \|\mathbf{k}_x\| \leq \kappa\sqrt{n}\|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\|. \tag{5.32}$$

Using the bound on $\|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\|$ given by inequality (5.31), the fact that the singular values of $(\mathbf{K}' + \lambda\mathbf{I})$ and $(\mathbf{K} + \lambda\mathbf{I})$ are larger than or equal to $\lambda$ since $\mathbf{K}$ and $\mathbf{K}'$ are SPSD matrices, and $\|\mathbf{y}\| \leq \sqrt{n}M$ yields

$$\begin{aligned} |h'(x) - h(x)| &\leq \frac{\kappa n M \|\mathbf{K}' - \mathbf{K}\|_2}{\lambda_{\min}(\mathbf{K}' + \lambda\mathbf{I})\lambda_{\min}(\mathbf{K} + \lambda\mathbf{I})} \\ &\leq \frac{\kappa M}{\lambda_0^2 n}\|\mathbf{K}' - \mathbf{K}\|_2. \end{aligned}$$

□

The generalization bounds for KRR, e.g., stability bounds (Bousquet & Elisseeff, 2001), are of the form $R(h) \leq \widehat{R}(h) + O(1/\sqrt{n})$, where $R(h)$ denotes the generalization error and $\widehat{R}(h)$ the empirical error of a hypothesis $h$ with respect to the square loss. The proposition shows that $|h'(x) - h(x)|^2 = O(\|\mathbf{K}' - \mathbf{K}\|_2^2/\lambda_0^4 n^2)$. Thus, it suggests that the kernel approximation tolerated should be such that $\|\mathbf{K}' - \mathbf{K}\|_2^2/\lambda_0^4 n^2 \ll O(1/\sqrt{n})$, that is, such that $\|\mathbf{K}' - \mathbf{K}\|_2 \ll O(\lambda_0^2 n^{3/4})$.

Note that the main bound used in the proof of the theorem, inequality (5.31), is tight in the sense that it can be matched for some kernels $K$ and $K'$. Indeed, let $K$ and $K'$ be the kernel functions defined by $K(x, y) = \beta$ and $K'(x, y) = \beta'$ if $x = y$, $K'(x, y) = K(x, y) = 0$ otherwise, with $\beta, \beta' \geq 0$. Then, the corresponding kernel matrices for a sample $S$ are $\mathbf{K}=\beta\mathbf{I}$ and $\mathbf{K}'=\beta'\mathbf{I}$, and the dual parameter vectors are given by $\boldsymbol{\alpha}=\mathbf{y}/(\beta+\lambda)$ and $\boldsymbol{\alpha}'=\mathbf{y}/(\beta'+\lambda)$. Now, since $\lambda_{\min}(\mathbf{K}'+\lambda\mathbf{I})=\beta'+\lambda$ and $\lambda_{\min}(\mathbf{K}+\lambda\mathbf{I})=\beta+\lambda$, and $\|\mathbf{K}'-\mathbf{K}\|=\beta'-\beta$, the following equality holds:

$$\begin{aligned}\|\boldsymbol{\alpha}'-\boldsymbol{\alpha}\| &= \frac{|\beta'-\beta|}{(\beta'+\lambda)(\beta+\lambda)}\|\mathbf{y}\| && (5.33)\\ &= \frac{\|\mathbf{K}'-\mathbf{K}\|}{\lambda_{\min}(\mathbf{K}'+\lambda\mathbf{I})\lambda_{\min}(\mathbf{K}'+\lambda\mathbf{I})}\|\mathbf{y}\|. && (5.34)\end{aligned}$$

This limits significant improvements of the bound of Proposition 5.1 using similar techniques.

### 5.3.2 Support Vector Machines

This section analyzes the kernel stability of SVMs. For simplicity, we shall consider the case where the classification function sought has no offset. In practice, this corresponds to using a constant feature. Let $\Phi\colon X \to F$ denote a feature mapping from the input space $X$ to a Hilbert space $F$ corresponding to some kernel $K$. The hypothesis set we consider is thus $H=\{h\colon \exists\mathbf{w}\in F|\forall x\in X, h(x)=\mathbf{w}^\top\Phi(x)\}$.

The following is the standard primal optimization problem for SVMs:

$$\min_{\mathbf{w}} F_K(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C_0\widehat{R}_K(\mathbf{w}), \tag{5.35}$$

where $\widehat{R}_K(\mathbf{w})=\frac{1}{n}\sum_{i=1}^n L(y_i\mathbf{w}^\top\Phi(x_i))$ is the empirical error and $L(y_i\mathbf{w}^\top\Phi(x_i))=$

$\max(0, 1 - y_i \mathbf{w}^\top \Phi(x_i))$ is the hinge loss associated to the $i$th point.

In the following, we analyze the difference between the hypothesis $h$ returned by SVMs when trained on the sample $S$ of $n$ points and using a kernel $K$, versus the hypothesis $h'$ obtained when training on the same sample with the kernel $K'$. For a fixed $x \in X$, we shall compare more specifically $h(x)$ and $h'(x)$. Thus, we can work with the finite set $X_{n+1} = \{x_1, \ldots, x_n, x_{n+1}\}$, with $x_{n+1} = x$.

Different feature mappings $\Phi$ can be associated to the same kernel $K$. To compare the solutions $\mathbf{w}$ and $\mathbf{w}'$ of the optimization problems based on $F_K$ and $F_{K'}$, we can choose the feature mappings $\Phi$ and $\Phi'$ associated to $K$ and $K'$ such that they both map to $\mathbb{R}^{n+1}$ as follows. Let $\mathbf{K}_{n+1}$ denote the Gram matrix associated to $K$ and $\mathbf{K}'_{n+1}$ that of kernel $K'$ for the set of points $X_{n+1}$. Then for all $u \in X_{n+1}$, $\Phi$ and $\Phi'$ can be defined by

$$\Phi(u) = \mathbf{K}_{n+1}^{+1/2} \begin{bmatrix} K(x_1, u) \\ \vdots \\ K(x_{n+1}, u) \end{bmatrix} \tag{5.36}$$

$$\text{and} \quad \Phi'(u) = \mathbf{K}'^{+1/2}_{n+1} \begin{bmatrix} K'(x_1, u) \\ \vdots \\ K'(x_{n+1}, u) \end{bmatrix}, \tag{5.37}$$

where $\mathbf{K}_{n+1}^{+}$ denotes the pseudo-inverse of $\mathbf{K}_{n+1}$ and $\mathbf{K}'^{+}_{n+1}$ that of $\mathbf{K}'_{n+1}$. It is not hard to see then that for all $u, v \in X_{n+1}$, $K(u, v) = \Phi(u)^\top \Phi(v)$ and

$K'(u,v) = \Phi'(u)^\top \Phi'(v)$ (Schölkopf & Smola, 2002). Since the optimization problem depends only on the sample $S$, we can use the feature mappings just defined in the expression of $F_K$ and $F_{K'}$. This does not affect in any way the standard SVMs optimization problem.

Let $\mathbf{w} \in \mathbb{R}^{n+1}$ denote the minimizer of $F_K$ and $\mathbf{w}'$ that of $F_{K'}$. By definition, if we let $\Delta\mathbf{w}$ denote $\mathbf{w}' - \mathbf{w}$, for all $s \in [0,1]$, the following inequalities hold:

$$F_K(\mathbf{w}) \leq F_K(\mathbf{w} + s\Delta\mathbf{w}) \tag{5.38}$$

$$\text{and} \quad F_{K'}(\mathbf{w}') \leq F_{K'}(\mathbf{w}' - s\Delta\mathbf{w}). \tag{5.39}$$

Summing these two inequalities, rearranging terms, and using the identity $(\|\mathbf{w}+s\Delta\mathbf{w}\|^2-\|\mathbf{w}\|^2)+(\|\mathbf{w}'-s\Delta\mathbf{w}\|^2-\|\mathbf{w}'\|^2)=-2s(1-s)\|\Delta\mathbf{w}\|^2$, we obtain as in Bousquet and Elisseeff (2001):

$$\begin{aligned} s(1-s)\|\Delta\mathbf{w}\|^2 \leq C_0\Big[&\big(\widehat{R}_K(\mathbf{w}+s\Delta\mathbf{w}) - \widehat{R}_K(\mathbf{w})\big) \\ &+ \big(\widehat{R}_{K'}(\mathbf{w}'-s\Delta\mathbf{w}) - \widehat{R}_{K'}(\mathbf{w}')\big)\Big]. \end{aligned}$$

Note that $\mathbf{w} + s\Delta\mathbf{w} = s\mathbf{w}' + (1-s)\mathbf{w}$ and $\mathbf{w}' - s\Delta\mathbf{w} = s\mathbf{w} + (1-s)\mathbf{w}'$. Then, by the convexity of the hinge loss and thus $\widehat{R}_K$ and $\widehat{R}_{K'}$, the following inequalities hold:

$$\begin{aligned}
&\widehat{R}_K(\mathbf{w}+s\Delta\mathbf{w})-\widehat{R}_K(\mathbf{w})\leq s(\widehat{R}_K(\mathbf{w}')-\widehat{R}_K(\mathbf{w}))\\
&\widehat{R}_{K'}(\mathbf{w}'-s\Delta\mathbf{w})-\widehat{R}_{K'}(\mathbf{w}')\leq -s(\widehat{R}_{K'}(\mathbf{w}')-\widehat{R}_{K'}(\mathbf{w})).
\end{aligned}$$

Plugging in these inequalities on the left-hand side, simplifying by $s$ and taking the limit $s\rightarrow 0$ yields

$$\begin{aligned}
\|\Delta\mathbf{w}\|^2 &\leq C_0\Big[\big(\widehat{R}_K(\mathbf{w}')-\widehat{R}_{K'}(\mathbf{w}')\big)+\big(\widehat{R}_{K'}(\mathbf{w})-\widehat{R}_K(\mathbf{w})\big)\Big]\\
&=\frac{C_0}{n}\sum_{i=1}^{n}\bigg[\Big(L(y_i\mathbf{w}'^\top\Phi(x_i))-L(y_i\mathbf{w}'^\top\Phi'(x_i))\Big)\\
&\qquad+\Big(L(y_i\mathbf{w}^\top\Phi'(x_i))-L(y_i\mathbf{w}^\top\Phi(x_i))\Big)\bigg],
\end{aligned}$$

where the last inequality results from the definition of the empirical error. Since the hinge loss is 1-Lipschitz, we can bound the terms on the right-hand side as follows:

$$\begin{aligned}
\|\Delta\mathbf{w}\|^2 &\leq \frac{C_0}{n}\sum_{i=1}^{n}\Big[\|\mathbf{w}'\|\|\Phi'(x_i)-\Phi(x_i)\|\\
&\qquad+\|\mathbf{w}\|\|\Phi'(x_i)-\Phi(x_i)\|\Big] && (5.40)\\
&=\frac{C_0}{n}\sum_{i=1}^{n}(\|\mathbf{w}'\|+\|\mathbf{w}\|)\ \|\Phi'(x_i)-\Phi(x_i)\|. && (5.41)
\end{aligned}$$

Let $e_i$ denote the $i$th unit vector of $\mathbb{R}^{n+1}$, then $(K(x_1,x_i),\ldots,K(x_{n+1},x_i))^\top =$

$\mathbf{K}_{n+1}e_i$. Thus, in view of the definition of $\Phi$, for all $i \in [1, n+1]$,

$$\begin{aligned}\Phi(x_i) &= \mathbf{K}_{n+1}^{+1/2}[K(x_1, x_i), \ldots, K(x_n, x_i), K(x, x_i)]^\top \\ &= \mathbf{K}_{n+1}^{+1/2}\mathbf{K}_{n+1}e_i = \mathbf{K}_{n+1}^{1/2}e_i, \qquad (5.42)\end{aligned}$$

and similarly $\Phi'(x_i) = \mathbf{K}_{n+1}^{\prime 1/2}e_i$. $\mathbf{K}_{n+1}^{1/2}e_i$ is the $i$th column of $\mathbf{K}_{n+1}^{1/2}$ and similarly $\mathbf{K}'^{1/2}e_i$ the $i$th column of $\mathbf{K}_{n+1}^{\prime 1/2}$. Thus, (5.41) can be rewritten as

$$\|\mathbf{w}' - \mathbf{w}\|^2 \leq \frac{C_0}{n}\sum_{i=1}^{n}\big(\|\mathbf{w}'\| + \|\mathbf{w}\|\big)\ \|(\mathbf{K}_{n+1}^{\prime 1/2} - \mathbf{K}_{n+1}^{1/2})e_i\|.$$

As for the case of KRR, we shall assume that there exists $\kappa > 0$ such that $K(x, x) \leq \kappa$ and $K'(x, x) \leq \kappa$ for all $x \in X_{n+1}$. Now, since $\mathbf{w}$ can be written in terms of the dual variables $0 \leq \alpha_i \leq C$, $C = C_0/n$ as $\mathbf{w} = \sum_{i=1}^{n} \alpha_i K(x_i, \cdot)$, it can be bounded as $\|\mathbf{w}\| \leq nC_0/n\kappa^{1/2} = \kappa^{1/2}C_0$ and similarly $\|\mathbf{w}'\| \leq \kappa^{1/2}C_0$. Thus, we can write

$$\begin{aligned}\|\mathbf{w}' - \mathbf{w}\|^2 &\leq \frac{2C_0^2\kappa^{1/2}}{n}\sum_{i=1}^{n}\|(\mathbf{K}_{n+1}^{\prime 1/2} - \mathbf{K}_{n+1}^{1/2})e_i\| \\ &\leq \frac{2C_0^2\kappa^{1/2}}{n}\sum_{i=1}^{n}\|(\mathbf{K}_{n+1}^{\prime 1/2} - \mathbf{K}_{n+1}^{1/2})\|\|e_i\| \\ &= 2C_0^2\kappa^{1/2}\|\mathbf{K}_{n+1}^{\prime 1/2} - \mathbf{K}_{n+1}^{1/2}\|. \qquad (5.43)\end{aligned}$$

Let $\mathbf{K}$ denote the Gram matrix associated to $K$ and $\mathbf{K}'$ that of kernel $K'$ for the sample $S$. Then, using Lemma 5.1 the following result holds.

**Lemma 5.1** *Let* $\mathbf{M}$ *and* $\mathbf{M}'$ *be two* $n \times n$ *SPSD matrices. Then, the following bound holds for the difference of the square root matrices:* $\|\mathbf{M}'^{1/2} - \mathbf{M}^{1/2}\|_2 \leq \|\mathbf{M}' - \mathbf{M}\|_2^{1/2}$.

*Proof.* Since $\mathbf{M}' - \mathbf{M} \preceq \|\mathbf{M}' - \mathbf{M}\|_2 \mathbf{I}$ where $\mathbf{I}$ is the $n \times n$ identity matrix. Thus, $\mathbf{M}' \preceq \mathbf{M} + \|\mathbf{M}' - \mathbf{M}\|_2 \mathbf{I}$ and $\mathbf{M}'^{1/2} \preceq (\mathbf{M} + \lambda \mathbf{I})^{1/2}$, with $\lambda = \|\mathbf{M}' - \mathbf{M}\|_2$. Thus, $\lambda_{\max}(\mathbf{M}') \leq (\lambda_{\max}(\mathbf{M}) + \lambda)^{1/2} \leq \lambda_{\max}(\mathbf{M})^{1/2} + \lambda^{1/2}$, by sub-additivity of $\sqrt{\cdot}$. This shows that $\lambda_{\max}(\mathbf{M}') - \lambda_{\max}(\mathbf{M})^{1/2} \leq \lambda$ and by symmetry $\lambda_{\max}(\mathbf{M})^{1/2} - \lambda_{\max}(\mathbf{M}') \leq \lambda^{1/2}$, thus $\|\mathbf{M}'^{1/2} - \mathbf{M}^{1/2}\|_2 \leq \|\mathbf{M}' - \mathbf{M}\|_2^{1/2}$, which proves the statement of the lemma. □

**Proposition 5.2** *Let* $h'$ *denote the hypothesis returned by SVMs when using the approximate kernel matrix* $\mathbf{K}' \in \mathbb{R}^{n \times n}$. *Then, the following inequality holds for all* $x \in \mathcal{X}$*:*

$$|h'(x) - h(x)| \leq \sqrt{2}\kappa^{\frac{3}{4}} C_0 \|\mathbf{K}' - \mathbf{K}\|_2^{\frac{1}{4}} \Big[1 + \big[\tfrac{\|\mathbf{K}' - \mathbf{K}\|_2}{4\kappa}\big]^{\frac{1}{4}}\Big]. \tag{5.44}$$

*Proof.* In view of (5.42) and (5.43), the following holds:

$$
\begin{aligned}
&|h'(x) - h(x)| \\
&= \|\mathbf{w}'^{\top}\Phi'(x) - \mathbf{w}^{\top}\Phi(x)\| \\
&= \|(\mathbf{w}' - \mathbf{w})^{\top}\Phi'(x) + \mathbf{w}^{\top}(\Phi'(x) - \Phi(x))\| \\
&\leq \|\mathbf{w}' - \mathbf{w}\|\|\Phi'(x)\| + \|\mathbf{w}\|\|\Phi'(x) - \Phi(x)\| \\
&= \|\mathbf{w}' - \mathbf{w}\|\|\Phi'(x)\| + \|\mathbf{w}\|\|\Phi'(x_{n+1}) - \Phi(x_{n+1})\| \\
&\leq \Big(2C_0^2\kappa^{1/2}\|\mathbf{K}'^{1/2}_{n+1} - \mathbf{K}^{1/2}_{n+1}\|\Big)^{1/2}\kappa^{1/2} \\
&\qquad + \kappa^{1/2}C_0\|(\mathbf{K}'^{1/2}_{n+1} - \mathbf{K}^{1/2}_{n+1})e_{n+1}\| \\
&\leq \sqrt{2}\kappa^{3/4}C_0\|\mathbf{K}'^{1/2}_{n+1} - \mathbf{K}^{1/2}_{n+1}\|^{1/2} \\
&\qquad + \kappa^{1/2}C_0\|\mathbf{K}'^{1/2}_{n+1} - \mathbf{K}^{1/2}_{n+1}\|.
\end{aligned}
$$

Now, by Lemma 5.1, $\|\mathbf{K}'^{1/2}_{n+1} - \mathbf{K}^{1/2}_{n+1}\|_2 \leq \|\mathbf{K}'_{n+1} - \mathbf{K}_{n+1}\|_2^{1/2}$. By assumption, the kernel approximation is only used at training time so $K(x, x_i) = K'(x, x_i)$, for all $i \in [1, n]$, and since by definition $x = x_{n+1}$, the last rows or the last columns of the matrices $\mathbf{K}'_{n+1}$ and $\mathbf{K}_{n+1}$ coincide. Therefore, the matrix $\mathbf{K}'_{n+1} - \mathbf{K}_{n+1}$ coincides with the matrix $\mathbf{K}' - \mathbf{K}$ bordered with a zero-column and zero-row and $\|\mathbf{K}'^{1/2}_{n+1} - \mathbf{K}^{1/2}_{n+1}\|_2 \leq \|\mathbf{K}' - \mathbf{K}\|_2^{1/2}$. Thus,

$$
|h'(x) - h(x)| \leq \sqrt{2}\kappa^{3/4}C_0\|\mathbf{K}' - \mathbf{K}\|^{1/4} + \kappa^{1/2}C_0\|\mathbf{K}' - \mathbf{K}\|^{1/2}, \qquad (5.45)
$$

which is exactly the statement of the proposition. □

Since the hinge loss $l$ is 1-Lipschitz, Proposition 5.2 leads directly to the following bound on the pointwise difference of the hinge loss between the hypotheses $h'$ and $h$.

**Corollary 5.1** *Let $h'$ denote the hypothesis returned by SVMs when using the approximate kernel matrix $\mathbf{K}' \in \mathbb{R}^{n \times n}$. Then, the following inequality holds for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$:*

$$\big|L\big(yh'(x)\big) - L\big(yh(x)\big)\big| \leq \sqrt{2}\kappa^{\frac{3}{4}} C_0 \|\mathbf{K}' - \mathbf{K}\|_2^{\frac{1}{4}} \Big[1 + \big[\tfrac{\|\mathbf{K}'-\mathbf{K}\|_2}{4\kappa}\big]^{\frac{1}{4}}\Big]. \qquad (5.46)$$

The bounds we obtain for SVMs are weaker than our bound for KRR. This is due mainly to the different loss functions defining the optimization problems of these algorithms.

### 5.3.3 Support Vector Regression

This section analyzes the kernel stability of Support Vector Regression. As with the case of SVMs, we shall consider the case where the regression function sought has no offset. Let $\Phi\colon X \to F$ denote a feature mapping from the input space $X$ to a Hilbert space $F$ corresponding to some kernel $K$. The hypothesis set we consider is thus $H = \{h\colon \exists \mathbf{w} \in F | \forall x \in X, h(x) = \mathbf{w}^\top \Phi(x)\}$. The standard primal optimization problem for SVR is identical to that of SVMs as expressed in (5.35), except that the hinge loss is replaced by the $\epsilon$-insensitive loss, i.e., $L(y_i - \mathbf{w}^\top \Phi(x_i)) = \max(0, |y_i - \mathbf{w}^\top \Phi(x_i)| - \epsilon)$.

Our goal is to analyze the difference between the hypothesis $h$ returned by

SVR when trained on the sample $S$ of $n$ points and using a kernel $K$, versus the hypothesis $h'$ obtained when training on the same sample with the kernel $K'$. In the previous section, we presented perturbation bounds for SVMs. However, this analysis is in fact more general, as the proof technique from Section 5.3.2 holds for all optimization problems of the form described in (5.35) that use convex and Lipschitz loss functions. Hence, Corollary 5.2 details the kernel stability bound for SVR that follows directly from the analysis in Section 5.3.2 along with the fact that the $\epsilon$-insensitive loss function is convex and 1-Lipschitz.

**Corollary 5.2** *Let $h'$ denote the hypothesis returned by SVR when using the approximate kernel matrix $\mathbf{K}' \in \mathbb{R}^{n\times n}$. Then, the following inequality holds for all $x \in \mathcal{X}$:*

$$|h'(x) - h(x)| \leq \sqrt{2}\kappa^{\frac{3}{4}} C_0 \|\mathbf{K}' - \mathbf{K}\|_2^{\frac{1}{4}} \Big[1 + \big[\tfrac{\|\mathbf{K}'-\mathbf{K}\|_2}{4\kappa}\big]^{\frac{1}{4}}\Big]. \tag{5.47}$$

### 5.3.4 Graph Laplacian regularization algorithms

We next study the kernel stability of graph-Laplacian regularization algorithms such as that of Belkin et al. (2004). Given a connected weighted graph $G = (X, E)$ in which edge weights can be interpreted as similarities between vertices, the task consists of predicting the vertex labels of $u$ vertices using a labeled training sample $S$ of $n$ vertices. The input space $\mathcal{X}$ is thus reduced to the set of vertices, and a hypothesis $h\colon \mathcal{X} \to \mathbb{R}$ can be identified with the finite-dimensional vector $\mathbf{h}$ of its predictions $\mathbf{h}=[h(x_1),\ldots,h(x_{n+u})]^\top$. The hypothesis set $H$ can thus be identified with $\mathbb{R}^{n+u}$ here. Let $\mathbf{h}_S$ denote the restriction of $\mathbf{h}$ to the training points, $[h(x_1),\ldots,h(x_n)]^\top \in \mathbb{R}^n$, and similarly let $\mathbf{y}_S$ denote $[y_1,\ldots,y_n]^\top \in \mathbb{R}^n$. Then, the following is the optimization problem corresponding to this problem:

$$\begin{aligned} \min_{\mathbf{h}\in H} \quad & \mathbf{h}^\top \mathbf{L}\mathbf{h} + \frac{C_0}{n}(\mathbf{h}_S - \mathbf{y}_S)^\top(\mathbf{h}_S - \mathbf{y}_S) \\ \text{subject to} \quad & \mathbf{h}^\top \mathbf{1} = 0, \end{aligned} \tag{5.48}$$

where $\mathbf{L}$ is the graph Laplacian and $\mathbf{1}$ the column vector with all entries equal to 1. Thus, $\mathbf{h}^\top\mathbf{L}\mathbf{h} = \sum_{ij=1}^n w_{ij}(h(x_i)-h(x_j))^2$, for some weight matrix $(w_{ij})$. The label vector $\mathbf{y}$ is assumed to be centered, which implies that $\mathbf{1}^\top\mathbf{y} = 0$. Since the graph is connected, the singular value zero of the Laplacian has multiplicity one.

Define $\mathbf{I}_S \in \mathbb{R}^{(n+u)\times(n+u)}$ to be the diagonal matrix with $[\mathbf{I}_S]_{i,i} = 1$ if $i \le n$ and 0 otherwise. Maintaining the notation used in Belkin et al. (2004), we let $\mathbf{P}_H$ denote the projection on the hyperplane $H$ orthogonal to $\mathbf{1}$ and let $\mathbf{M} = \mathbf{P}_H\left(\frac{n}{C_0}\mathbf{L} + \mathbf{I}_S\right)$ and $\mathbf{M}' = \mathbf{P}_H\left(\frac{n}{C_0}\mathbf{L}' + \mathbf{I}_S\right)$. We denote by $\mathbf{h}$ the hypothesis returned by the algorithm when using the exact kernel matrix $\mathbf{L}$ and by $\mathbf{L}'$ an approximate graph Laplacian such that $\mathbf{h}^\top\mathbf{L}'\mathbf{h} = \sum_{ij=1}^n w'_{ij}(h(x_i)-h(x_j))^2$, based on matrix $(w'_{ij})$ instead of $(w_{ij})$. We shall assume that there exist $M>0$ such that $y_i \le M$ for $i \in [1,n]$.

**Proposition 5.3** *Let* $\mathbf{h}'$ *denote the hypothesis returned by the graph-Laplacian* *regularization algorithm when using an approximate Laplacian* $\mathbf{L}' \in \mathbb{R}^{n\times n}$*. Then, the following inequality holds:*

$$\|\mathbf{h}' - \mathbf{h}\| \leq \frac{n^{3/2}M/C_0}{(\frac{n}{C_0}\widehat{\lambda}_2 - 1)^2}\|\mathbf{L}' - \mathbf{L}\|, \tag{5.49}$$

*where* $\widehat{\lambda}_2 = \max\{\lambda_2, \lambda'_2\}$ *with* $\lambda_2$ *denoting the second smallest singular value of the kernel matrix* $\mathbf{L}$ *and* $\lambda'_2$ *the second smallest singular value of* $\mathbf{L}'$*.*

*Proof.* The closed-form solution of (5.48) is given by Belkin et al. (2004): $\mathbf{h} = \left(\mathbf{P}_H\left(\frac{n}{C_0}\mathbf{L} + \mathbf{I}_S\right)\right)^{-1}\mathbf{y}_S$. Thus, we can use that expression and the matrix identity for $(\mathbf{M}^{-1} - \mathbf{M}'^{-1})$ we already used in the analysis of KRR to write

$$\begin{aligned}
\|\mathbf{h} - \mathbf{h}'\| &= \|\mathbf{M}^{-1}\mathbf{y}_S - \mathbf{M}'^{-1}\mathbf{y}_S\| \\
&= \|(\mathbf{M}^{-1} - \mathbf{M}'^{-1})\mathbf{y}_S\| \\
&= \|-\mathbf{M}^{-1}(\mathbf{M} - \mathbf{M}')\mathbf{M}'^{-1}\mathbf{y}_S\| \\
&\leq \frac{n}{C_0}\|-\mathbf{M}^{-1}(\mathbf{L} - \mathbf{L}')\mathbf{M}'^{-1}\mathbf{y}_S\| \\
&\leq \frac{n}{C_0}\|\mathbf{M}^{-1}\|\;\|\mathbf{M}'^{-1}\|\;\|\mathbf{y}_S\|\;\|\mathbf{L}' - \mathbf{L}\|.
\end{aligned} \tag{5.50}$$

For any column matrix $\mathbf{z} \in \mathbb{R}^{(n+u)\times 1}$, by the triangle inequality and the projection property $\|\mathbf{P}_H\mathbf{z}\| \leq \|\mathbf{z}\|$, the following inequalities hold:

$$\begin{aligned}
\|\frac{n}{C_0}\mathbf{P}_H\mathbf{L}\| &= \|\frac{n}{C_0}\mathbf{P}_H\mathbf{L} + \mathbf{P}_H\mathbf{I}_S\mathbf{z} - P_H\mathbf{I}_S\mathbf{z}\| \\
&\leq \|\frac{n}{C_0}\mathbf{P}_H\mathbf{L} + \mathbf{P}_H\mathbf{I}_S\mathbf{z}\| + \|\mathbf{P}_H\mathbf{I}_S\mathbf{z}\| \\
&\leq \|\mathbf{P}_H\Big(\frac{n}{C_0}\mathbf{L} + \mathbf{I}_S\Big)\mathbf{z}\| + \|\mathbf{I}_S\mathbf{z}\|.
\end{aligned}$$

This yields the lower bound:

$$\begin{aligned}
\|\mathbf{M}\mathbf{z}\| &= \|\mathbf{P}_H\left(\frac{n}{C_0}\mathbf{L} + \mathbf{I}_S\right)\mathbf{z}\| \\
&\geq \frac{n}{C_0}\|\mathbf{P}_H\mathbf{L}\| - \|\mathbf{I}_S\mathbf{z}\| \\
&\geq \left(\frac{n}{C_0}\lambda_2 - 1\right)\|\mathbf{z}\|,
\end{aligned}$$

which gives the following upper bounds on $\|\mathbf{M}^{-1}\|$ and $\|\mathbf{M}'^{-1}\|$:

$$\|\mathbf{M}^{-1}\| \leq \frac{1}{\frac{n}{C_0}\lambda_2 - 1} \quad \text{and} \quad \|\mathbf{M}'^{-1}\| \leq \frac{1}{\frac{n}{C_0}\lambda_2' - 1}.$$

Plugging in these inequalities in (5.50) and using $\|\mathbf{y}_S\| \leq n^{1/2}M$ lead to

$$\|\mathbf{h} - \mathbf{h}'\| \leq \frac{n^{3/2}M/C_0}{(\frac{n}{C_0}\lambda_2 - 1)(\frac{n}{C_0}\lambda_2' - 1)}\|\mathbf{L}' - \mathbf{L}\|.$$

□

The generalization bounds for the graph-Laplacian algorithm are of the form $R(h) \leq \widehat{R}(h) + O(\frac{n}{(\frac{n}{C_0}\lambda_2 - 1)^2})$ (Belkin et al., 2004). In view of the bound given by the theorem, this suggests that the approximation tolerated should verify $\|\mathbf{L}' - \mathbf{L}\| \ll O(1/\sqrt{n})$.

### 5.3.5 Kernel Principal Component Analysis

Let $K$ be a PDS kernel defined over $\mathcal{X} \times \mathcal{X}$ and $\Phi \colon \mathcal{X} \to \mathbb{R}^N$ a feature mapping corresponding to $K$. Consider a zero-mean sample $S = (x_1, \ldots, x_n)$. Let $\mathbf{X}$ denote the matrix $(\Phi(x_1), \ldots, \Phi(x_n))$. The covariance matrix $\mathbf{C}$ and the kernel matrix $\mathbf{K}$ associated to $S$ are defined by $\mathbf{C} = \mathbf{X}\mathbf{X}^\top$ and $\mathbf{K} = \mathbf{X}^\top\mathbf{X}$.[3] Principal Component Analysis (PCA) is defined by the projection over the top $k$ singular vectors of $\mathbf{C}$. Since $\mathbf{X}$ admits the following thin singular value decomposition, $\mathbf{X} = \mathbf{V}\mathbf{\Lambda}\mathbf{U}^\top$, $\mathbf{C}$ and $\mathbf{K}$ can be rewritten as $\mathbf{C} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^\top$ and $\mathbf{K} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$, where $\mathbf{\Sigma} = \mathbf{\Lambda}^2$ is the diagonal matrix of the non-zero singular values of $\mathbf{C}$ and $\mathbf{K}$. $\mathbf{V}$ is thus the matrix of the singular vectors of $\mathbf{C}$.

Note that $\mathbf{V} = \mathbf{X}\mathbf{U}\mathbf{\Sigma}^{-1/2}$. Thus, the singular vector $\mathbf{v}$ of $\mathbf{C}$ associated to the singular value $\sigma$ coincides with $\frac{\mathbf{X}\mathbf{u}}{\sqrt{\sigma}}$, where $\mathbf{u}$ is the singular vector of $\mathbf{K}$ associated to $\sigma$. Given an arbitrary feature vector $\Phi(x)$, $x \in \mathcal{X}$, its projection over the singular vector $\mathbf{v}$ is thus defined by

$$\Phi(x)^\top \mathbf{v} = \Phi(x)^\top \frac{\mathbf{X}\mathbf{u}}{\sqrt{\sigma}} = \frac{\mathbf{k}_x^\top \mathbf{u}}{\sqrt{\sigma}}, \tag{5.51}$$

where $\mathbf{k}_x = (K(x_1, x), \ldots, K(x_n, x))^\top$. Thus, KPCA is fully defined by the top $k$ singular vectors of $\mathbf{K}$, $\mathbf{u}_1, \ldots, \mathbf{u}_k$, and the corresponding singular values.

[3] Note that in this section, $\mathbf{C}$ refers to the covariance matrix, and not the $n \times l$ submatrix used for sampling-based matrix approximations.

We now define $\mathbf{P}_{V_k}(\Phi(x))$ as the projection of $\Phi(x)$ onto the top $k$ singular vectors of $\mathbf{C}$, and observe that

$$\begin{aligned}\|P_{V_k}(\Phi(x))\|^2 = \sum_{i=1}^{k}(\Phi(x)^\top \mathbf{v}_i)^2 &= \sum_{i=1}^{k}\frac{\mathbf{k}_x^\top \mathbf{u}_i \mathbf{u}_i^\top \mathbf{k}_x}{\sigma_i} \\ &= \mathbf{k}_x^\top \mathbf{U}_k \mathbf{\Sigma}_k^{-1} \mathbf{U}_k^\top \mathbf{k}_x.\end{aligned}$$

In particular, for $j \in [1, n]$, $k_{x_j} = \mathbf{K}\mathbf{e}_j$, and we have:

$$\begin{aligned}\hat{\mathbf{E}}\big[\|P_{V_k}(\Phi(x))\|^2\big] &= \frac{1}{n}\sum_{j=1}^{n}\|P_{V_k}(\Phi(x_j))\|^2 = \frac{1}{n}\sum_{j=1}^{n}\mathbf{e}_j^\top \mathbf{K}\mathbf{U}_k\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^\top \mathbf{K}\mathbf{e}_j \\ &= \frac{1}{n}\sum_{j=1}^{n}\mathbf{e}_j^\top \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{U}_k^\top \mathbf{e}_j = \frac{1}{n}\sum_{i=1}^{k}\sigma_i,\end{aligned}$$

where $\hat{\mathbf{E}}$ is an expectation over the sample $S$. Let $\hat{R}(K) = \hat{\mathbf{E}}\big[\|\Phi(x)\|^2\big] - \hat{\mathbf{E}}\big[\|P_{V_k}(\Phi(x))\|^2\big]$ denote the average empirical residue of the kernel $K$. Following similar steps as above it can be shown that $\hat{\mathbf{E}}\big[\|\Phi(x)\|^2\big] = \frac{1}{n}\sum_{i=1}^{n}\sigma_i$, and so we have:

$$\hat{R}(K) = \frac{1}{n}\sum_{i>k}\sigma_i. \tag{5.52}$$

If the kernel $K$ is approximated with the PDS kernel $K'$, the difference of empirical residues is thus

$$|\hat{R}(K') - \hat{R}(K)| = \frac{1}{n}\Big|\sum_{i>k}\sigma_i(\mathbf{K}') - \sigma_i(\mathbf{K})\Big| \tag{5.53}$$

where $\sigma_i(\cdot)$ refers to the $i$th singular value of its argument. We now present

Proposition 5.4, which provides a bound on the effect of kernel perturbation on empirical residuals.

**Proposition 5.4** *The difference of empirical residuals of $K'$ and $K$ is bounded as follows:*

$$|\hat{R}(K') - \hat{R}(K)| \leq \Big(1 - \frac{k}{n}\Big) \|\mathbf{K}' - \mathbf{K}\|_2. \tag{5.54}$$

*Proof.* The result follows by combining (5.53) with Weyl's inequality (Golub & Loan, 1983), which states that $|\lambda_i(\mathbf{K}') - \lambda_i(\mathbf{K})| \leq \|\mathbf{K}' - \mathbf{K}\|_2$. □

Similar bounds exist for the Frobenius norm. In the remainder of this section, we show how Proposition 5.4 can be used to derive various bounds for applications of Kernel PCA.

### Expected residual

We first explore the effect of kernel perturbation on the expected residual. Let $\mathbf{V}_k^{opt}$ be the matrix of $k$ orthonormal vectors that minimize the average residue over all possible datapoints, i.e. $\mathbf{V}_k^{opt}$ is the minimizer of $\hat{R}(K)$ as $n \to \infty$. We then define $R(K)$ as the expected residue of the kernel, $\mathbf{E}_{\mathcal{X}}\left[\|\Phi(x)\|^2\right] - \mathbf{E}\left[\|P_{V_k^{opt}}(\Phi(x))\|^2\right]$. Previous work has focused on bounding $|R(K) - \hat{R}(K)|$ as summarized by Theorem 5.6, which holds under Assumption 5.1 (Zwald et al., 2004; Shawe-Taylor et al., 2005).

**Assumption 5.1 ((Zwald et al., 2004), Assumption 1)** *Let $D$ denote a distribution on $\mathcal{X}$ according to which $x_1, \ldots, x_n$ are sampled i.i.d. Define $K$ to* *be a positive definite function on $\mathcal{X}$ and $\mathcal{H}_k$ the associated reproducing kernel Hilbert space. We assume that:*

- *for all $x \in \mathcal{X}$, $K(x, \cdot)$ is $D$-measurable.*
- *there exists $\kappa > 0$ such that $K(x, x) \leq \kappa$ $D$-almost surely.*
- *$\mathcal{H}_k$ is separable.*

**Theorem 5.6 ((Zwald et al., 2004), Thm 4)** *Under Assumption 5.1, with probability at least $1 - \delta$,*

$$-\kappa\sqrt{\frac{\log(3/\delta)}{2n}} \leq R(K) - \hat{R}(K) \quad \leq \frac{2\sqrt{k}}{n}\sqrt{\sum_{i=1}^{n} K^2(x_i, x_i)} + 3\kappa\sqrt{\frac{\log(3/\delta)}{2n}}. \tag{5.55}$$

Since we are focusing on kernel approximation, we would like to analyze $|R(K) - \hat{R}(K')|$, or the additional residual error incurred by using a kernel approximation in addition to empirically estimating the optimal projection subspace from a sample of $n$ points. Corollary 5.3 provides a bound of this quantity, while Corollary 5.4 presents a simplified bound in the case of normalized kernels.

**Corollary 5.3** *Under Assumption 5.1 and with probability at least $1 - \delta$, the difference between the expected residual of $K$ and the empirical residual of $K'$*

*is bounded as follows:*

$$|R(K)-\hat{R}(K')| \leq \frac{2\sqrt{k}}{n}\sqrt{\sum_{i=1}^{n} K^2(x_i,x_i)}+3\kappa\sqrt{\frac{\log(3/\delta)}{2n}}+\Big(1-\frac{k}{n}\Big)\|\mathbf{K}'-\mathbf{K}\|_2. \tag{5.56}$$

*Proof.* Using the triangle inequality we have,

$$|R(K) - \hat{R}(K')| \leq |R(K) - \hat{R}(K)| + |\hat{R}(K') - \hat{R}(K)|\,. \tag{5.57}$$

We bound the first term on the RHS using Theorem 5.6 and bound the second term on the RHS using Proposition 5.4. □

**Corollary 5.4** *Assume that $K(\cdot,\cdot)$ is normalized, i.e., $K(x,x) = 1$ $D$-almost surely, as in the case of Gaussian kernels. Then under Assumption 5.1 and with probability at least $1-\delta$, the difference between the expected residual of $K$ and the empirical residual of $K'$ is bounded as follows:*

$$|R(K) - \hat{R}(K')| \;\leq\; 2\sqrt{\frac{k}{n}} + 3\kappa\sqrt{\frac{\log(3/\delta)}{2n}} + \Big(1-\frac{k}{n}\Big)\|\mathbf{K}' - \mathbf{K}\|_2. \tag{5.58}$$

### Subspace perturbation

In the previous analysis, we focused on residuals, which involved projecting the datapoints onto subspaces spanned by the singular vectors of the covariance matrix. We now focus on the effect of kernel perturbation on the subspaces themselves. We define $\mathbf{P}_{\hat{S}_k} = \mathbf{V}_k\mathbf{V}_k^\top$ and $\mathbf{P}_{S_k} = \mathbf{V}_k^{opt}\mathbf{V}_k^{opt\top}$ as the orthogonal projectors onto the subspaces spanned by the top $k$ singular vectors of the empirical and process covariance matrices, respectively, associated with $K(\cdot,\cdot)$. Similarly, we define $\mathbf{P}_{\hat{S}'_k}$ as the orthogonal projector onto the subspace spanned by the top $k$ singular vectors of the empirical covariance matrix associated with $K'(\cdot,\cdot)$. We would like to analyze $\|\mathbf{P}_{S_k} - \mathbf{P}_{\hat{S}'_k}\|_F$, or the difference in the approximate empirical subspace and the optimal process subspace. Drawing upon the work of Zwald and Blanchard (2005), as summarized by Theorems 5.7 and 5.8, we derive our desired bound on subspace perturbation in Corollary 5.5.

**Theorem 5.7 ((Zwald & Blanchard, 2005), Thm 3)** *Let $\mathbf{K}$ be a kernel matrix with simple nonzero singular values $\sigma_1 > \sigma_2 > \ldots$, and define $k$ as an integer such that $\sigma_k > 0$ and $\Delta_k = \frac{1}{2}(\sigma_k - \sigma_{k+1})$. Let $\mathbf{K}'$ be a perturbation of $\mathbf{K}$ such that $\mathbf{K}'$ is still positive semidefinite and $\|\mathbf{K} - \mathbf{K}'\|_F \le \Delta_k/2$. Then the following bound holds on the difference between the orthogonal projectors onto the subspaces spanned by the top $k$ singular vectors of the covariance matrices associated with $\mathbf{K}$ and $\mathbf{K}'$:*

$$\|P_{\hat{S}_k} - P_{\hat{S}'_k}\|_F \le \frac{\|\mathbf{K}' - \mathbf{K}\|_F}{\Delta_k} \tag{5.59}$$

**Theorem 5.8 ((Zwald & Blanchard, 2005), Thm 4)** *Denote the singular values of the kernel operator associated with $K(\cdot,\cdot)$ by $\bar{\sigma}_1 > \bar{\sigma}_2 > \ldots$, and define $k$ as an integer such that $\bar{\sigma}_k > 0$ and $\bar{\Delta}_k = \frac{1}{2}(\bar{\sigma}_k - \bar{\sigma}_{k+1})$. Assume that*

$\sup_{x \in \mathcal{X}} K(x,x) \leq \kappa$. *Define*

$$B_k = \frac{2\kappa}{\bar{\Delta}_k}\Big(1 + \sqrt{\frac{\log(1/\delta}{2}}\Big)\,. \tag{5.60}$$

*Then, with probability at least* $1-\delta$*, provided that* $n \geq B_k^2$*, the following bound holds on the difference between the orthogonal projectors onto the subspaces spanned by the top* $k$ *singular vectors of the empirical and process covariance matrices associated with* $K(\cdot,\cdot)$*:*

$$\|P_{S_k} - P_{\hat{S}_k}\|_F \leq \frac{B_k}{\sqrt{n}} \tag{5.61}$$

**Corollary 5.5** *Following the definitions and assumptions of Theorems 5.7 and 5.8, then, with probability at least* $1-\delta$*, the following bound holds on the difference between the orthogonal projectors onto the subspaces spanned by the top* $k$ *singular vectors of the empirical covariance matrix associated with* $K'(\cdot,\cdot)$ *and the process covariance matrix associated with* $K(\cdot,\cdot)$*:*

$$\|P_{S_k} - P_{\hat{S}'_k}\|_F \leq \frac{B_k}{\sqrt{n}} + \frac{\|\mathbf{K}' - \mathbf{K}\|_F}{\Delta_k} \tag{5.62}$$

*Proof.* Using the triangle inequality we have,

$$\|P_{S_k} - P_{\hat{S}'_k}\|_F \leq \|P_{S_k} - P_{\hat{S}_k}\|_F + \|P_{\hat{S}_k} - P_{\hat{S}'_k}\|_F\,. \tag{5.63}$$

Next, we bound the first term on the RHS using Theorem 5.8 and bound the second term on the RHS using Theorem 5.7. □

In summary, Corollaries 5.3 and 5.4 show the connections between expected residuals and the reconstruction error of the sample kernel matrix due to kernel perturbation. Similarly, Corollary 5.5 shows the connection between this reconstruction error and subspace perturbation. Together, these results generalize previous theory on expected residuals and subspace perturbation to also account for the effect of perturbations of the empirical kernel matrix.

### 5.3.6 Application to Nyström method

The previous section provided stability analyses for several common learning algorithms, studying the effect of using an approximate kernel matrix instead of the true one. The difference in hypothesis value is expressed simply in terms of the difference between the kernels measured by some norm. Although these bounds are general bounds that are independent of how the approximation is obtained (so long as $\mathbf{K}'$ remains SPSD), one relevant application of these bounds involves the Nyström method. As shown by Williams and Seeger (2000), later by Drineas and Mahoney (2005); Talwalkar et al. (2008); Zhang et al. (2008), low-rank approximations of the kernel matrix via the Nyström method can provide an effective technique for tackling large-scale data sets. However, all previous theoretical studies analyzing the performance of the Nyström method have focused on the quality of the low-rank approximations, rather than the performance of the kernel learning algorithms used in conjunction with these approximations. In this section, we show how we can leverage kernel stability analysis to present novel performance guarantees for the Nyström method in the context of kernel learning algorithms.

### Nyström Kernel Ridge Regression

The accuracy of low-rank Nyström approximations was discussed in Section 5.1. The following corollary, which is a simplified adaptation of Theorem 5.2, gives an upper bound on the norm-2 error of the Nyström approximation of the form $\|\mathbf{K}-\widetilde{\mathbf{K}}\|_2/\|\mathbf{K}\|_2 \leq \|\mathbf{K}-\mathbf{K}_k\|_2/\|\mathbf{K}\|_2 + O(1/\sqrt{l})$. We denote by $\mathbf{K}_{\max}$ the maximum diagonal entry of $\mathbf{K}$.

**Corollary 5.6** *Let $\widetilde{\mathbf{K}}$ denote the rank-k Nyström approximation of $\mathbf{K}$ based on l columns sampled uniformly at random with 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:*

$$\|\mathbf{K}-\widetilde{\mathbf{K}}\|_2 \leq \|\mathbf{K}-\mathbf{K}_k\|_2 + \tfrac{n}{\sqrt{l}}\mathbf{K}_{\max}\big(2+\log\tfrac{1}{\delta}\big).$$

Corollary 5.6 focuses on the quality of low-rank approximations. Combining the analysis from Section 5.3.1 with this corollary enables us to bound the relative performance of the kernel learning algorithms when the Nyström method is used as a means of scaling kernel learning algorithms. To illustrate this point, Theorem 5.9 uses Proposition 5.1 along with Corollary 5.6 to upper bound the relative performance of KRR as a function of the approximation accuracy of the Nyström method (a similar technique can be used to bound the error of the Nyström approximation when used with the other algorithms discussed in Section 5.3).

**Theorem 5.9** *Let $h'$ denote the hypothesis returned by Kernel Ridge Regression when using the approximate rank-$k$ kernel $\widetilde{\mathbf{K}} \in \mathbb{R}^{n \times n}$ generated using the Nyström method. Then, with probability at least $1-\delta$, the following inequality holds for all $x \in X$,*

$$|h'(x) - h(x)| \leq \frac{\kappa M}{\lambda_0^2 n}\Big[\|\mathbf{K} - \mathbf{K}_k\|_2 + \tfrac{n}{\sqrt{l}}\mathbf{K}_{\max}\big(2 + \log\tfrac{1}{\delta}\big)\Big].$$

We note that the experimental results of Figure 3.9 are useful in analyzing our theoretical analysis. In these experiments, we generated approximate kernel matrices using the Nyström method, and for approximations of varying quality we measured the perturbation of associated KRR hypotheses (see Section 3.2.2 for details of the experimental design). The experimental results suggest a linear relationship between kernel approximation and hypothesis perturbation, which corroborates the shape of the bound in Proposition 5.1.

## 5.4 Summary

In this section we presented a variety of analyses of the Nyström method in the context of machine learning. We first presented theoretical results comparing the quality of the Nyström approximation to the 'best' low-rank approximation, under sampling assumptions that are commonly used in practice, namely, uniform sampling without replacement. These bounds hold for both the standard Nyström method as well as the ensemble Nyström method. We then made a connection between matrix coherence and the performance of the Nyström method. We derived novel coherence-based bounds for the Nyström method in the low-rank setting, and presented empirical studies that convincingly demonstrate the ability of matrix coherence to measure the degree to which information can be extracted from a subset of columns both in the low-rank and full-rank settings. Finally, we addressed the issue of how kernel approximation affects the performance of learning algorithms. Our analysis is independent of how the approximation is obtained and simply expresses the change in hypothesis value in terms of the difference between the approximate kernel matrix and the true one measured by some norm. We also provided a specific analysis of the Nyström low-rank approximation in this context and discussed experimental results that support our theoretical analysis.

# Chapter 6

# Conclusion

We addressed the question of how machine learning algorithms, in particular kernel methods, can handle large-scale data. We focused on an attractive solution to this problem that involves sampling-based techniques to efficiently generate low-rank matrix approximations. In Chapter 2, we answered the question of what sampling-based approximation should be used. We discussed two common sampling-based methods, providing novel theoretical insights regarding their suitability for various applications and experimental results motivated by this theory. Our results show that one of these methods, the Nyström method, is superior in the context of large-scale learning. In Chapter 3, we focused on the applicability of sampling-based low-rank approximations for practical applications, showing the effectiveness of approximation techniques on a variety of problems. In particular, we presented the largest study to-date for manifold learning using the Nyström method to extract low-dimensional structure from high-dimensional data to effectively cluster face images. We also discussed the connection between low-rank matrices and the Woodbury approximation, reporting good empirical results for Kernel Ridge Regression and Kernel Logistic Regression using the Nyström method.

An important open question from these two chapters involves obtaining a better characterization of approximate low-dimensional embeddings, as this task appears to be closely related to the task of approximating spectral reconstruction, yet the empirical results are quite different. Additionally, a more in-depth study is required to understand the convergence properties of the Nyström Kernel Logistic Regression algorithm and to compare its performance relative to other optimization techniques, e.g., Gradient Descent, Conjugate Gradient, Iterative Scaling, Quasi-Newton, etc.

Next, in Chapter 4, we addressed a crucial aspect of these sampling based algorithms, namely, the method used to select a subset of columns. We focused our discussion on the Nyström method given its superior performance on large-scale tasks in Chapters 2 and 3. We first studied both fixed and adaptive sampling schemes, and showed that given fixed time constraints, uniform sampling works remarkably well. Next, we introduced a promising ensemble technique that can be easily parallelized and generates superior approximations, both in theory and in practice.

This area remains ripe for future work. A finer theoretical analysis of our Nyström adaptive sampling technique is required, perhaps inspired by work in (Deshpande et al., 2006). Additionally, new sampling distributions suggested in recent work have led to improved theoretical bounds for matrix projection reconstruction (Drineas et al., 2008; Mahoney & Drineas, 2009). Although these sampling techniques are inefficient to compute as they are derived from the singular vectors of $\mathbf{K}$ (in the case of a SPSD matrix), perhaps similar sampling distributions can be computed more efficiently and be used to generate superior low-rank approximations. Alternatively, sampling methods that account for the learning task that will use the resultant low-rank approximation could also lead to better performance (this idea has been studied by (Bach & Jordan, 2005) for the special case of the Incomplete Cholesky algorithm). Finally, in reference to the ensemble Nyström algorithm, an interesting avenue of future work involves the use of different types of base learners to generate an ensemble approximation. For instance, base learners could be generated using other (or a combination of) sampling schemes discussed in Sections 4.1 and 4.2, or even using a variety of low-rank approximation methods, e.g., a combination of Nyström and Column-sampling approximations.

Finally, in Section 5, we provided a variety of theoretical analyses of the Nyström method. We first presented general guarantees on approximation accuracy and then introduced coherence-based bounds. We also studied the effect of matrix approximation on actual kernel-based algorithms. There is room for improvement in the analysis of the Nyström method. The first set of bounds can likely be tightened by performing a direct analysis of the Nyström method, rather than using an indirect analysis based on approximation of matrix multiplication. The coherence-based bounds can be generalized to matrices with full-rank, and would be more practically relevant if there existed algorithms to efficiently estimate the coherence of a matrix to determine the applicability of the Nyström method on a case-by-case basis. In terms of kernel perturbation, future work involves analyzing additional kernel-based algorithms and tightening existing perturbation bounds, especially those for SVMs and SVR.

In summary, we have shown that sampling-based low-rank approximation is an effective tool that extends to large-scale applications the benefits of kernel-based algorithms, namely their empirical effectiveness and solid theoretical underpinnings.

# Bibliography

Achlioptas, D., & Mcsherry, F. (2007). Fast computation of low-rank matrix approximations. *Journal of the ACM*, *54*.

Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., & Mohri, M. (2007). OpenFST: A general and efficient weighted finite-state transducer library. *Conference on Implementation and Application of Automata*.

Arora, S., Hazan, E., & Kale, S. (2006). A fast random sampling algorithm for sparsifying matrices. *Approx-Random*.

Asuncion, A., & Newman, D. (2007). UCI machine learning repository. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Bach, F. R., & Jordan, M. I. (2002). Kernel Independent Component Analysis. *Journal of Machine Learning Research*, *3*, 1–48.

Bach, F. R., & Jordan, M. I. (2005). Predictive low-rank decomposition for kernel methods. *International Conference on Machine Learning*.

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

Balasubramanian, M., & Schwartz, E. L. (2002). The Isomap algorithm and topological stability. *Science*, *295*.

Belabbas, M., & Wolfe, P. J. (2009). On landmark selection and sampling in high-dimensional data analysis. `arXiv:0906.4582v1[stat.ML]`.

Belabbas, M. A., & Wolfe, P. J. (2009). 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*, 369–374.

Belkin, M., Matveeva, I., & Niyogi, P. (2004). Regularization and semi-supervised learning on large graphs. *Conference on Learning Theory*.

Belkin, M., & Niyogi, P. (2001). Laplacian Eigenmaps and spectral techniques for embedding and clustering. *Neural Information Processing Systems*.

Belkin, M., & Niyogi, P. (2006). Convergence of Laplacian Eigenmaps. *Neural Information Processing Systems*.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Conference on Learning Theory*.

Bousquet, O., & Elisseeff, A. (2001). Algorithmic stability and generalization performance. *Neural Information Processing Systems*.

Boutsidis, C., Mahoney, M. W., & Drineas, P. (2009). An improved approximation algorithm for the column subset selection problem. *Symposium on Discrete Algorithms*.

Candès, E. J., & Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, *9*, 717–772.

Candès, E. J., & Romberg, J. (2007). Sparsity and incoherence in compressive sampling. *Inverse Problems*, *23*, 969–986.

Candès, E. J., Romberg, J. K., & Tao, T. (2006). Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, *52*, 489–509.

Candès, E. J., & Tao, T. (2009). The power of convex relaxation: Near-optimal matrix completion. `arXiv:0903.1476v1[cs.IT]`.

Cawley, G., & Talbot, N. (2004). Miscellaneous MATLAB software. `http://theoval.cmp.uea.ac.uk/matlab/default.html#cholinc`.

Chang, E., Zhu, K., Wang, H., Bai, H., Li, J., Qiu, Z., & Cui, H. (2008). Parallelizing Support Vector Machines on distributed computers. *Neural Information Processing Systems*.

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.). (2006). *Semi-supervised learning*. Cambridge, MA: MIT Press.

Cortes, C., Mohri, M., Pechyony, D., & Rastogi, A. (2008). Stability of transductive regression algorithms. *International Conference on Machine Learning*.

Cortes, C., Mohri, M., & Talwalkar, A. (2010). On the impact of kernel approximation on learning accuracy. *Conference on Artificial Intelligence and Statistics*.

Cox, T. F., Cox, M. A. A., & Cox, T. F. (2000). *Multidimensional scaling*. Chapman & Hall/CRC. 2nd edition.

de Silva, V., & Tenenbaum, J. (2003). Global versus local methods in nonlinear dimensionality reduction. *Neural Information Processing Systems*.

Deshpande, A., Rademacher, L., Vempala, S., & Wang, G. (2006). Matrix approximation and projective clustering via volume sampling. *Symposium on Discrete Algorithms*.

Donoho, D. L. (2006). Compressed Sensing. *IEEE Transactions on Information Theory*, *52*, 1289–1306.

Donoho, D. L., & Grimes, C. (2003). Hessian Eigenmaps: locally linear embedding techniques for high dimensional data. *Proceedings of the National Academy of Sciences of the United States of America*, *100*, 5591–5596.

Drineas, P. (2008). Personal communication.

Drineas, P., Drinea, E., & Huggins, P. S. (2001). An experimental evaluation of a Monte-Carlo algorithm for SVD. *Panhellenic Conference on Informatics*.

Drineas, P., Kannan, R., & Mahoney, M. W. (2006a). Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal of Computing*, *36*.

Drineas, P., Kannan, R., & Mahoney, M. W. (2006b). Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, *36*.

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.

Drineas, P., Mahoney, M. W., & Muthukrishnan, S. (2008). Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, *30*, 844–881.

Fergus, R., Weiss, Y., & Torralba, A. (2009). Semi-supervised learning in gigantic image collections. *Neural Information Processing Systems*.

Fine, S., & Scheinberg, K. (2002). 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. *Transactions on Pattern Analysis and Machine Intelligence*, *26*, 214–225.

Frieze, A., Kannan, R., & Vempala, S. (1998). Fast Monte-Carlo algorithms for finding low-rank approximations. *Foundation of Computer Science*.

Ghahramani, Z. (1996). The kin datasets. `http://www.cs.toronto.edu/~delve/data/kin/desc.html`.

Golub, G., & Loan, C. V. (1983). *Matrix computations*. Baltimore: Johns Hopkins University Press. 2nd edition.

Goreinov, S. A., Tyrtyshnikov, E. E., & Zamarashkin, N. L. (1997). A theory of pseudoskeleton approximations. *Linear Algebra and Its Applications*, *261*, 1–21.

Gorrell, G. (2006). Generalized Hebbian algorithm for incremental Singular Value Decomposition in natural language processing. *European Chapter of the Association for Computational Linguistics*.

Gu, M., & Eisenstat, S. C. (1996). Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM Journal of Scientific Computing*, *17*, 848–869.

Gustafson, A., Snitkin, E., Parker, S., DeLisi, C., & Kasif, S. (2006). Towards the identification of essential genes using targeted genome sequencing and comparative analysis. *BMC:Genomics*, *7*, 265.

Halko, N., Martinsson, P. G., & Tropp, J. A. (2009). Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. `arXiv:0909.4061v1[math.NA]`.

Ham, J., Lee, D. D., Mika, S., & Schölkopf, B. (2004). A kernel view of the dimensionality reduction of manifolds. *International Conference on Machine Learning*.

Har-peled, S. (2006). Low-rank matrix approximation in linear time, manuscript.

Hastie, T., & Tibshirani, R. (1990). *Generalized additive models*. Chapman and Hall.

He, X., Yan, S., Hu, Y., & Niyogi, P. (2005). Face recognition using Laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 328–340.

Huang, L., Yan, D., Jordan, M., & Taft, N. (2008). Spectral clustering with perturbed data. *Neural Information Processing Systems*.

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

Joachims, T. (1999). Making large-scale Support Vector Machine learning practical. *Neural Information Processing Systems*.

Johnson, W. B., & Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, *26*, 189–206.

Karsmakers, P., Pelckmans, K., Suykens, J., & hamme, J. V. (2007). Fixed-size Kernel Logistic Regression for phoneme classification. *Interspeech*.

Keerthi, S. S., Duan, K. B., Shevade, S. K., & Poo, A. N. (2005). A fast dual algorithm for Kernel Logistic Regression. *Machine Learning*, *61*, 151–165.

Kumar, S., Mohri, M., & Talwalkar, A. (2009a). Ensemble Nyström method. *Neural Information Processing Systems*.

Kumar, S., Mohri, M., & Talwalkar, A. (2009b). On sampling-based approximate spectral decomposition. *International Conference on Machine Learning*.

Kumar, S., Mohri, M., & Talwalkar, A. (2009c). Sampling techniques for the Nyström method. *Conference on Artificial Intelligence and Statistics*.

Kumar, S., & Rowley, H. (2010). People Hopper. `http://googleresearch.blogspot.com/2010/03/hopping-on-face-manifold-via-people.html`.

LeCun, Y., & Cortes, C. (1998). The MNIST database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`.

Liberty, E. (2009). *Accelerated dense random projections*. Ph.D. thesis, computer science department, Yale University, New Haven, CT.

Littlestone, N., & Warmuth, M. K. (1994). The Weighted Majority algorithm. *Information and Computation*, *108*, 212–261.

Liu, R., Jain, V., & Zhang, H. (2006). Subsampling for efficient spectral mesh processing. *Computer Graphics International Conference*.

Liu, T., Moore, A. W., Gray, A. G., & Yang, K. (2004). An investigation of practical approximate nearest neighbor algorithms. *Neural Information Processing Systems*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*, 91–110.

Mahoney, M. W., & Drineas, P. (2009). CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, *106*, 697–702.

Mann, G., McDonald, R., Mohri, M., Silberman, N., & Walker, D. (2009). Efficient large-scale distributed training of conditional maximum entropy models. *Neural Information Processing Systems*.

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

Ouimet, M., & Bengio, Y. (2005). Greedy spectral embedding. *Artificial Intelligence and Statistics*.

Papadimitriou, C. H., Tamaki, H., Raghavan, P., & Vempala, S. (1998). Latent Semantic Indexing: a probabilistic analysis. *Principles of Database Systems*.

Platt, J. C. (1999). Fast training of Support Vector Machines using sequential minimal optimization. *Neural Information Processing Systems*.

Platt, J. C. (2004). Fast embedding of sparse similarity graphs. *Neural Information Processing Systems*.

Rokhlin, V., Szlam, A., & Tygert, M. (2009). A randomized algorithm for Principal Component Analysis. *SIAM Journal on Matrix Analysis and Applications*, *31*, 1100–1124.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by Locally Linear Embedding. *Science*, *290*.

Rudelson, M. (1999). Random vectors in the isotropic position. *Journal of Functional Analysis*, *164*, 60–72.

Rudelson, M., & Vershynin, R. (2007). Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM*, *54*, 21.

Ruston, A. F. (1964). Auerbachs theorem. *Mathematical Proceedings of the Cambridge Philosophical Society*, *56*, 476–480.

Saunders, C., Gammerman, A., & Vovk, V. (1998). Ridge Regression learning algorithm in dual variables. *International Conference on Machine Learning*.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels*. MIT Press: Cambridge, MA.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.

Shawe-Taylor, J., Williams, C. K. I., Cristianini, N., & Kandola, J. S. (2005). On the eigenspectrum of the Gram matrix and the generalization error of Kernel-PCA. *IEEE Transactions on Information Theory*, *51*, 2510–2522.

Sim, T., Baker, S., & Bsat, M. (2002). The CMU pose, illumination, and expression database. *Conference on Automatic Face and Gesture Recognition*.

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

Smola, A. J., & Schölkopf, B. (2000). Sparse Greedy Matrix Approximation for machine learning. *International Conference on Machine Learning*.

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

Talwalkar, A., Kumar, S., & Rowley, H. (2008). Large-scale manifold learning. *Conference on Vision and Pattern Recognition*.

Talwalkar, A., & Rostamizadeh, A. (2010). Matrix coherence and the Nyström method. `arXiv:1004.2008v1[cs.AI]`.

Tenenbaum, J., de Silva, V., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*.

Weinberger, K. Q., & Saul, L. K. (2006). An introduction to nonlinear dimensionality reduction by maximum variance unfolding. *AAAI Conference on Artificial Intelligence*.

Williams, C. K. I., & Seeger, M. (2000). Using the Nyström method to speed up kernel machines. *Neural Information Processing Systems.*

Zhang, K., & Kwok, J. T. (2009). Density-weighted Nyström method for computing large kernel eigensystems. *Neural Computation*, *21*, 121–146.

Zhang, K., Tsang, I., & Kwok, J. (2008). Improved Nyström low-rank approximation and error analysis. *International Conference on Machine Learning.*

Zwald, L., & Blanchard, G. (2005). On the convergence of eigenspaces in Kernel Principal Component Analysis. *Neural Information Processing Systems.*

Zwald, L., Bousquet, O., & Blanchard, G. (2004). Statistical properties of Kernel Principal Component Analysis. *Conference on Learning Theory.*