Provably Useful Kernel Matrix Approximation in Linear Time

Cameron Musco
MIT
cnmusco@mit.edu

Christopher Musco
MIT
cpmusco@mit.edu

June 9, 2016

Abstract

We give the first algorithms for kernel matrix approximation that run in time *linear in the number of data points* and output an approximation which gives provable guarantees when used in many downstream learning tasks, including kernel principal component analysis, kernel k-means clustering, kernel ridge regression, and kernel canonical correlation analysis.

Our methods require just $\tilde{O}(n \cdot k)$ kernel evaluations and $\tilde{O}(n \cdot k^2)$ additional runtime, where n is the number of training data points and k is a target rank or effective dimensionality parameter. These runtimes are significantly sub-linear in the size of the $n \times n$ kernel matrix and apply to *any kernel matrix*, without assuming regularity or incoherence conditions.

The algorithms are based on a *ridge leverage score* Nyström sampling scheme (RLS-Nyström) which was recently shown to yield strong kernel approximations, but which had no efficient implementation [AM15, RCR15, Wan16]. We address this shortcoming by introducing fast recursive sampling methods for RLS-Nyström, while at the same time proving extended approximation guarantees for this promising new method.

1 Introduction

The kernel method is an extremely popular learning tool, which implicitly maps data to a high-dimensional feature space before applying linear learning methods [SS02]. In this way, classical linear techniques such as support vector machines, ridge regression, principal component analysis, and k-means clustering can be used for nonlinear learning tasks – linear relationships in the high dimensional feature space correspond to nonlinear relationships in the original input space.

Unfortunately kernel learning is slow. Since points cannot be explicitly mapped to the kernel feature space, which is often infinite dimensional, the *kernel trick* is employed: an efficient kernel function is used to compute inner products between the high dimensional mappings of input points. Given n input points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ is formed where $\mathbf{K}_{i,j}$ contains the high dimensional inner product between every $\mathbf{x}_i$ and $\mathbf{x}_j$. All computations of a linear learning algorithm are then performed using the inner product information contained in $\mathbf{K}$, which inherently requires space (and computation time) *quadratic* in the size of the dataset.

1.1 Kernel approximation

Since this cost is prohibitive for large datasets, a large body of work focuses on quickly and space efficiently approximating the kernel matrix $\mathbf{K}$ with a *sketch* $\tilde{\mathbf{K}}$ [AMS01, BBV06, SS00, WS01, FS02, DM05, RR07, GM13]. While this approach has seen significant practical and theoretical success, the theory of kernel approximation is still lacking in two respects:

Problem 1: Usefulness While many algorithms compute a $\tilde{\mathbf{K}}$ that *provably approximates* $\mathbf{K}$, the approximation metric is rarely tied to downstream learning tasks. For example, several papers study algorithms for finding low-rank approximations or entry-wise approximations to $\mathbf{K}$. While qualitatively useful, these metrics do not imply that $\tilde{\mathbf{K}}$ can be used in place of $\mathbf{K}$ to *provably* approximate downstream computations.

Problem 2: Efficiency Amongst those methods that *do* guarantee useful kernel approximations [Bac13, AM15, RCR15, YPW15], none run provably quickly for all kernel matrices. Ideally, we want to avoid computing $\mathbf{K}$ entirely, running as fast as linear-time heuristic methods.

1.2 Our contributions

In this paper we address both issues by significantly pushing forward a recent line of research on *ridge leverage score based Nyström approximation*, a kernel approximation algorithm proposed by Alaoui and Mahoney [AM15] that we refer to as **RLS-Nyström**. One of many Nyström methods based on *non-uniform* subsampling of input points [DM05, ZTK08, KMT12, GM13, WZ13, LJS16], RLS-Nyström has shown promise: it outputs an approximate kernel $\tilde{\mathbf{K}}$ that is provably useful in kernel ridge regression and kernel canonical correlation analysis and can be represented in space linear in the number of data points n [AM15, RCR15, Wan16].

However, current methods for performing RLS-Nyström are slow. They require computing the so called ridge leverage scores of $\mathbf{K}$, which contain fine grained information about the $\mathbf{K}$'s spectrum. While these scores can be computed quickly for *some* matrices [AM15] it is not known how to do so for many natural instances [YPW15]. In the worst case, computing $\tilde{\mathbf{K}}$ could take $O(n^3)$ time.

Our main contribution is a recursive sampling scheme that computes the ridge leverage scores of any kernel matrix without forming the whole matrix. The key idea is that it is possible to compute

approximations to the ridge leverage scores by *uniformly* sampling a constant fraction of the input points. By recursively approximating the (still large) sampled kernel matrix, we can perform fast leverage score computation, running in time significantly sublinear in the size of $\mathbf{K}$.

Our methods are based on recent iterative sampling methods for spectral approximation and low-rank approximation of large matrices [CLM+15, CMM15]. However, our implementation is more involved as it must work with the kernelized dataset implicitly, without forming all of $\mathbf{K}$.

In addition to an efficient implementation, we provide a new analysis of RLS-Nyström which gives general approximation bounds for $\tilde{\mathbf{K}}$ that can be used to recover all prior results on downstream approximation guarantees. Our approach also yields new guarantees for kernel principal component analysis and kernel k-means clustering. The results are summarized in Table 1.

Table 1: We summarize the downstream guarantees offered by the kernel approximation $\tilde{\mathbf{K}}$ obtained from RLS-Nyström. Notice that for all problems the runtime required to compute $\tilde{\mathbf{K}}$ and the space required to store $\tilde{\mathbf{K}}$ depend linearly on the number of training data points n.

Application	Downstream Guarantee	Relevant Theorem	Space to store $\tilde{\mathbf{K}}$	Time to compute $\tilde{\mathbf{K}}$
Kernel Ridge Regression w/ Parameter λ	$(1+\epsilon)$ relative error risk bound	Thm 10	$\tilde{O}(\frac{nd_{\text{eff}}}{\epsilon})^{\dagger}$	$\tilde{O}(\frac{nd_{\text{eff}}^2}{\epsilon^2}) + \tilde{O}(\frac{nd_{\text{eff}}}{\epsilon})$ kernel evals.
Kernel k-means Clustering	$(1+\epsilon)$ relative error	Thm 11	$\tilde{O}(\frac{nk}{\epsilon})$	$\tilde{O}(\frac{nk^2}{\epsilon^2}) + \tilde{O}(\frac{nk}{\epsilon})$ kernel evals.
Rank k Kernel PCA	$(1+\epsilon)$ relative Frobenius norm error	Thm 12	$\tilde{O}(\frac{nk}{\epsilon})$	$\tilde{O}(\frac{nk^2}{\epsilon^2}) + \tilde{O}(\frac{nk}{\epsilon})$ kernel evals.
Kernel CCA w/ Regularization Params λ_x, λ_y	ϵ additive error to canonical correlation	Thm 13	$\tilde{O}(\frac{nd_{\text{eff}}^x + nd_{\text{eff}}^y}{\epsilon})$	$\tilde{O}(\frac{n(d_{\text{eff}}^x)^2 + n(d_{\text{eff}}^y)^2}{\epsilon^2}) +$ $\tilde{O}(\frac{nd_{\text{eff}}^x + nd_{\text{eff}}^y}{\epsilon})$ kernel evals.

* For simplicity, $\tilde{O}(\cdot)$ hides log factors in the failure probability, d_{eff}, and k.
† $d_{\text{eff}} = \text{tr}(\mathbf{K}(\mathbf{K}+\lambda\mathbf{I})^{-1})$ is the so-called "effective dimensionality" of the ridge regression [HTF02].

1.3 Prior Work

It was not previously clear that any method could provably approximate an arbitrary kernel matrix without at least examining every entry of the matrix, and thus requiring at least $O(n^2)$ time. All previous sub-quadratic time algorithms are either:

1. Based on uniform sampling or similar techniques for Nyström approximation that only work well under regularity and incoherence conditions on $\mathbf{K}$ [Git11, KMT12, AM15]. The extent to which these conditions can be assumed for practical data is debated (see [GM13]).

2. Based on random Fourier features, which are not currently known to give strong guarantees for downstream learning tasks in comparison to Nyström methods [RR07, And09, CP16, GPP16].

3. Apply to specific limited complexity kernels, e.g. constant degree polynomial kernels [ANW14].

1.4 Future work

Our primary focus is on advancing the theory behind the RLS-Nyström method and proving that it can be implemented in linear time in the number of training data points n. An important next step is to empirically test and develop practical implementations of our RLS-Nyström algorithms. We note that, aside from algorithmic considerations, preliminary experimental results on the effectiveness of RLS-Nyström for kernel approximation are available in [AM15], [RCR15], and [Wan16].

1.5 Paper organization

We first introduce the RLS-Nyström approximation scheme in Section 3, without considering algorithmic issues. We present and prove simple but powerful approximation metrics for $\tilde{\mathbf{K}}$ that allow it to be used in place of $\mathbf{K}$ for a variety of downstream learning tasks. In Section 4, we give algorithms for computing an RLS-Nyström approximation in linear time. Our problem specific results from Table 1 are proven in Appendix A using the approximation metrics from Section 3. Auxiliary lemmas are included in Appendix B and C.

2 Preliminaries

Consider an input space $\mathcal{X}$ and a positive semidefinite kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Let $\mathcal{F}$ be an associated reproducing kernel Hilbert space and $\phi : \mathcal{X} \to \mathcal{F}$ be a (typically non-linear) feature map such that for any $x, y \in \mathcal{X}$, $K(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{F}}$. Given a set of n input points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, define the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ by $\mathbf{K}_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$.

It will often be natural to consider the kernelized data matrix that generates $\mathbf{K}$. Informally, let $\mathbf{\Phi} \in \mathbb{R}^{n \times d'}$ be the matrix containing $\phi(\mathbf{x}_1), ..., \phi(\mathbf{x}_n)$ as its rows (note that d' may be infinite). $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$. While we will use $\mathbf{\Phi}$ for intuition, in our formal proofs we will replace it with any matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ satisfying $\mathbf{B}\mathbf{B}^T = \mathbf{K}$ (e.g. a Cholesky factor).

In our proofs, we will repeatedly make we will make use of the singular value decomposition, which allows us to write any rank r matrix $\mathbf{M} \in \mathbb{R}^{n \times d}$ as $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n \times r}$ and $\mathbf{V} \in \mathbb{R}^{d \times r}$ have orthogonal columns (the left and right singular vectors of $\mathbf{A}$), and $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$ is a positive diagonal matrix containing the singular values of $\mathbf{M}$: $\sigma_1(\mathbf{M}) \geq \sigma_2(\mathbf{M}) \geq \ldots \geq \sigma_r(\mathbf{M})$. The pseudoinverse of $\mathbf{M}$ is given by $\mathbf{M}^+ = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^\top$.

3 The RLS-Nyström method

We begin by describing the RLS-Nyström method of [AM15], which computes an efficiently representable approximate kernel matrix $\tilde{\mathbf{K}}$ by combining ridge leverage score sampling with the well studied Nyström method [WS01]. Without yet considering how to efficiently implement RLS-Nyström, we show that it gives several strong guarantees for approximating $\mathbf{K}$. Unlike ad-hoc guarantees (e.g. element-wise approximation or low-rank approximation of $\mathbf{K}$), these guarantees are *provably sufficient for use in many downstream maching learning applications.*

3.1 Nyström approximation

The Nyström method approximates $\mathbf{K}$ by subsampling input points. Given a sampling matrix $\mathbf{S} \in \mathbb{R}^{n \times s}$ with a single nonzero entry per column, the associated Nyström approximation is:

$$\tilde{\mathbf{K}} = \mathbf{KS}(\mathbf{S}^T\mathbf{KS})^+\mathbf{S}^T\mathbf{K} \tag{1}$$

where $^+$ denotes the pseudoinverse. $\tilde{\mathbf{K}}$ can be stored in $O(ns)$ space by separately storing $\mathbf{KS} \in \mathbb{R}^{n \times s}$ and $(\mathbf{SKS}^T)^+ \in \mathbb{R}^{s \times s}$. Furthermore, the factors can be computed using just $O(ns)$ kernel evaluations to form $\mathbf{KS}$ and $O(s^3)$ time to compute $(\mathbf{S}^T\mathbf{KS})^+$. Typically $s << n$ so these costs are significantly lower than the cost to form and store the full kernel matrix $\mathbf{K}$.

We view Nyström approximation as a low-rank approximation to the dataset in feature space. Recalling that $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$, $\mathbf{S}$ selects s kernelized data points $\mathbf{S}^T\mathbf{\Phi}$ and we approximate $\mathbf{\Phi}$ using its projection onto these points. Informally, let $\mathbf{P_S} \in \mathbb{R}^{d' \times d'}$ be the orthogonal projection onto the row span of $\mathbf{S}^T\mathbf{\Phi}$. We approximate $\tilde{\mathbf{\Phi}} = \mathbf{\Phi}\mathbf{P_S}$. We can write $\mathbf{P_S} = \mathbf{\Phi}^T\mathbf{S}(\mathbf{S}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{S})^+\mathbf{S}^T\mathbf{\Phi}$. Since it is an orthogonal projection, $\mathbf{P_S}\mathbf{P_S}^T = \mathbf{P_S}^2 = \mathbf{P_S}$ and so:

$$\begin{aligned}\tilde{\mathbf{K}} = \tilde{\mathbf{\Phi}}\tilde{\mathbf{\Phi}}^T = \mathbf{\Phi}\mathbf{P_S}^2\mathbf{\Phi}^T &= \mathbf{\Phi}\left(\mathbf{\Phi}^T\mathbf{S}(\mathbf{S}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{S})^+\mathbf{S}^T\mathbf{\Phi}\right)\mathbf{\Phi}^T \\ &= \mathbf{KS}(\mathbf{S}^T\mathbf{KS})^+\mathbf{S}^T\mathbf{K}.\end{aligned}$$

This recovers the standard Nyström approximation (1). Note that the above view is presented for intuition – we do not rigorously handle the infinite dimensionality of the feature space. However, the argument can be made formal by replacing $\mathbf{\Phi}$ with any $\mathbf{B} \in \mathbb{R}^{n \times n}$ satisfying $\mathbf{B}\mathbf{B}^T = \mathbf{K}$. Such a $\mathbf{B}$ is guaranteed to exist since $\mathbf{K}$ is positive semidefinite.

3.2 Ridge leverage scores

Classically, $\mathbf{S}$ is formed by sampling data points uniformly at random [WS01, BW09]. However, a large body of research focuses on non-uniform importance sampling strategies based off diagonal entries, column norms, leverage scores, and DPPs [DM05, KMT12, GM13, WZ13, LJS16]. These strategies give approximation bounds that do not require assumptions on $\mathbf{K}$.

Recent work shows that sampling points via their ridge leverage scores gives bounds on $\tilde{\mathbf{K}}$ that are provably useful for ridge regression and canonical correlation analysis [AM15, RCR15, Wan16].

Definition 1 (Ridge Leverage Scores)**.** *For any $\lambda > 0$ and any $\mathbf{B} \in \mathbb{R}^{n \times n}$ satisfying $\mathbf{B}\mathbf{B}^T = \mathbf{K}$, the λ-ridge leverage score of data point $\mathbf{x}_i$ with respect to the kernel matrix $\mathbf{K}$ is given by:*

$$l_i^\lambda(\mathbf{K}) = \mathbf{b}_i^T(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{I})^{-1}\mathbf{b}_i \tag{2}$$

where $\mathbf{b}_i^T \in \mathbb{R}^{1 \times n}$ is the i^{th} row of $\mathbf{B}$.

Above $\mathbf{I}$ refers to the $n \times n$ identity matrix. For ease of notation we will write $l_i^\lambda(\mathbf{K})$ simply as l_i^λ and include the argument only when referring to the ridge leverage scores of some kernel matrix other than $\mathbf{K}$. Additionally, note that we will typically replace λ with $\epsilon\lambda$ for some error parameter $\epsilon \in (0, 1)$, using the scores $l_i^{\epsilon\lambda}$. Finally, while $\mathbf{B}$ will be useful computationally, the value of l_i^λ *does not depend* on the choice of this matrix – any square root of $\mathbf{K}$ can be used. As noted in [AM15], $l_i^\lambda = \left(\mathbf{K}(\mathbf{K} + \lambda\mathbf{I})^{-1}\right)_{i,i}$. Informally, using $\mathbf{\Phi}$ in place of $\mathbf{B}$, the above definition is identical to the definition of ridge leverage scores given in [CMM15] for the rows of the kernelized dataset $\mathbf{\Phi}$.

3.3 The basic algorithm

The RLS-Nyström method is simple to describe. We include pseudocode as Algorithm 1 below, but again do not address how to efficiently implement each step of the algorithm until Section 4.

Algorithm 1 RLS-Nyström Sampling

input: $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, kernel matrix $\mathbf{K}$, ridge parameter $\lambda > 0$, $\epsilon, \delta \in (0,1)$
output: kernel approximation $\tilde{\mathbf{K}}$

1: Compute over-approximations to the $\epsilon\lambda$-ridge leverage scores of $\mathbf{x}_1, \ldots, \mathbf{x}_n$, $\tilde{l}_i^{\epsilon\lambda} \geq l_i^{\epsilon\lambda}$
2: Set oversampling parameter $q := \log(\sum \tilde{l}_i^{\epsilon\lambda}/\delta)$
3: Set $p_i := c \cdot \min\left\{q\tilde{l}_i^{\epsilon\lambda}, 1\right\}$ for sufficiently large constant c
4: Construct $\mathbf{S} \in \mathbb{R}^{n\times s}$ by sampling $\mathbf{x}_1, \ldots, \mathbf{x}_n$ each independently with probability p_i. In other words, for each i add a column to $\mathbf{S}$ with a 1 in position i with probability p_i
5: Form the Nyström approximation $\tilde{\mathbf{K}} := \mathbf{KS}(\mathbf{SKS})^{+}\mathbf{SK}$

Note that since points are sampled independently, s is actually a random variable in RLS-Nyström – when analyzing Algorithm 1 we will show that it is not too large with good probability. Also note that we do not form $\tilde{\mathbf{K}}$ explicitly, as this would take space and time quadratic in n. We simply return the $s \times s$ matrix $(\mathbf{SKS})^{+}$ along with $\mathbf{KS}$.

3.4 Accuracy Bounds

We first show that the RLS-Nyström method gives an *additive error kernel embedding*.

Theorem 2 (Additive Error Kernel Embedding)**.** *For any $\lambda > 0$ and $\epsilon, \delta \in (0,1)$, RLS-Nyström returns an $\mathbf{S} \in \mathbb{R}^{n\times s}$ such that with probability $1-\delta$, $s = O\left(\sum_i p_i\right)$ and the approximation $\tilde{\mathbf{K}}$ satisfies:*

$$\tilde{\mathbf{K}} \preceq \mathbf{K} \preceq \tilde{\mathbf{K}} + \epsilon\lambda\mathbf{I}. \tag{3}$$

When ridge scores are computed exactly, $\sum_i p_i = O\left(\frac{d_{\text{eff}}}{\epsilon}\log\frac{d_{\text{eff}}}{\delta\epsilon}\right)$ where $d_{\text{eff}} = \text{tr}(\mathbf{K}(\mathbf{K}+\lambda\mathbf{I})^{-1})$.

$\preceq$ denotes the standard Loewner matrix ordering on positive semi-definite matrices[1]. Note that (3) immediately implies that $\|\mathbf{K} - \tilde{\mathbf{K}}\|_2 \leq \epsilon\lambda$ and it is in fact equivalent to this well studied [GM13] spectral norm guarantee as long as we assume $\tilde{\mathbf{K}} \preceq \mathbf{K}$. (3) also appears directly in the streaming and sketching literature [GLPW15, CMM15].

Intuitively, Theorem 2 guarantees that the $\tilde{\mathbf{K}}$ produced by RLS-Nyström very well approximates the top of $\mathbf{K}$'s spectrum (i.e. any eigenvalues $> \lambda$) while allowing it to eliminate information about smaller eigenvalues, which are less important for many learning tasks. This guarantee is already sufficient for using $\tilde{\mathbf{K}}$ to approximately solve kernel ridge regression (Theorem 10).

Proof. It is clear from the view of Nyström approximation as a low-rank projection of the kernelized data (see Section 3.1) that $\tilde{\mathbf{K}} \preceq \mathbf{K}$. Formally, for any $\mathbf{B} \in \mathbb{R}^{n\times n}$ with $\mathbf{BB}^T = \mathbf{K}$:

$$\tilde{\mathbf{K}} = \mathbf{KS}(\mathbf{S}^T\mathbf{KS})^{+}\mathbf{S}^T\mathbf{K} = \mathbf{BP}_{\mathbf{S}}\mathbf{B}^T$$

[1] $\mathbf{M} \preceq \mathbf{N}$ means that $\mathbf{N} - \mathbf{M}$ is positive semidefinite.

where $\mathbf{P_S} = \mathbf{B}^T\mathbf{S}(\mathbf{S}^T\mathbf{B}\mathbf{B}^T\mathbf{S})^+\mathbf{S}^T\mathbf{B}$ is the orthogonal projection onto the row span of $\mathbf{S}^T\mathbf{B}$. Since $\mathbf{P_S}$ is a projection $\|\mathbf{P_S}\|_2 \leq 1$. So, for any $\mathbf{x} \in \mathbb{R}^n$:

$$\mathbf{x}^T\tilde{\mathbf{K}}\mathbf{x} = \mathbf{x}^T\mathbf{B}\mathbf{P_S}\mathbf{B}\mathbf{x} = \|\mathbf{P_S}\mathbf{B}\mathbf{x}\|_2^2 \leq \|\mathbf{B}\mathbf{x}\|_2^2 = \mathbf{x}^T\mathbf{K}\mathbf{x}$$

giving $\tilde{\mathbf{K}} \preceq \mathbf{K}$. So, it just remains to show $\mathbf{K} \preceq \tilde{\mathbf{K}} + \epsilon\lambda\mathbf{I}$.

It can be shown via a matrix Bernstein bound that if $\mathbf{S}$ samples rows of $\mathbf{B}$ by their ridge leverage scores, $\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B}$ will spectrally approximate $\mathbf{B}^T\mathbf{B}$ up to an additive ridge error (see Lemma 14 in Appendix B). We require a relatively weak corollary of this fact which appears in Appendix B as Corollary 15: with probability $1-\delta$, $s = O\left(\sum_i p_i\right)$ and there is some scaling factor C such that

$$\mathbf{B}^T\mathbf{B} \preceq C \cdot \mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I}. \tag{4}$$

Let $\bar{\mathbf{P}}_\mathbf{S} = \mathbf{I} - \mathbf{P_S}$ be the projection onto the complement of the row span of $\mathbf{S}^T\mathbf{B}$. By (4):

$$\bar{\mathbf{P}}_\mathbf{S}\mathbf{B}^T\mathbf{B}\bar{\mathbf{P}}_\mathbf{S} \preceq C \cdot \bar{\mathbf{P}}_\mathbf{S}\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B}\bar{\mathbf{P}}_\mathbf{S} + \epsilon\lambda\bar{\mathbf{P}}_\mathbf{S}\mathbf{I}\bar{\mathbf{P}}_\mathbf{S}. \tag{5}$$

Since $\bar{\mathbf{P}}_\mathbf{S}$ projects to the complement of the row span of $\mathbf{S}^T\mathbf{B}$, $\mathbf{S}^T\mathbf{B}\bar{\mathbf{P}}_\mathbf{S} = \mathbf{0}$. So (5) gives:

$$\bar{\mathbf{P}}_\mathbf{S}\mathbf{B}^T\mathbf{B}\bar{\mathbf{P}}_\mathbf{S} \preceq \mathbf{0} + \epsilon\lambda\bar{\mathbf{P}}_\mathbf{S}\mathbf{I}\bar{\mathbf{P}}_\mathbf{S} \preceq \epsilon\lambda\mathbf{I}.$$

In other notation, $\|\bar{\mathbf{P}}_\mathbf{S}\mathbf{B}^T\mathbf{B}\bar{\mathbf{P}}_\mathbf{S}\|_2 \leq \epsilon\lambda$. This in turn implies $\|\mathbf{B}\bar{\mathbf{P}}_\mathbf{S}\mathbf{B}^T\|_2 \leq \epsilon\lambda$ and hence:

$$\mathbf{B}\bar{\mathbf{P}}_\mathbf{S}\mathbf{B}^T = \mathbf{B}(\mathbf{I} - \mathbf{P_S})\mathbf{B}^T \preceq \epsilon\lambda\mathbf{I}.$$

Rearranging, and using $\mathbf{K} = \mathbf{B}\mathbf{B}^T$ and $\tilde{\mathbf{K}} = \mathbf{B}\mathbf{P_S}\mathbf{B}^T$ gives the theorem.

We conclude by noting that, if exact ridge leverage scores are used in Algorithm 1, $\sum_i p_i = \sum l_i^{\epsilon\lambda}\log(\sum l_i^{\epsilon\lambda}/\delta)$. Since the $\epsilon\lambda$ ridge leverage scores are the diagonal entries of $\mathbf{K}(\mathbf{K}+\epsilon\lambda\mathbf{I})^{-1}$,

$$\sum_i l_i^{\epsilon\lambda} = \operatorname{tr}(\mathbf{K}(\mathbf{K}+\epsilon\lambda\mathbf{I})^{-1}) \leq \frac{1}{\epsilon}\operatorname{tr}(\mathbf{K}(\mathbf{K}+\lambda\mathbf{I})^{-1}).$$

Accordingly, Theorem 2 guarantees that $\mathbf{S}$ contains just $O\left(\frac{d_{\text{eff}}}{\epsilon}\log\frac{d_{\text{eff}}}{\delta\epsilon}\right)$ samples. □

The additive error kernel embedding bound of Theorem 2 easily implies what's called a *projection-cost preserving* kernel embedding, which is presented next in Theorem 3. Since its recent introduction, projection-cost preservation has proven a powerful concept in the matrix sketching literature [FSS13, CEM+15, CMM15, BWZ16]. We hope that an explicit projection-cost preservation guarantee for kernels will lead to applications of RLS-Nyström beyond those considered in this paper.

Theorem 3 (Projection-Cost Preserving Kernel Embedding)**.** *Let* $\lambda = \frac{1}{k}\sum_{i=k+1}^n \sigma_i(\mathbf{K})$. *For any* $\epsilon, \delta \in (0,1)$, *RLS-Nyström returns an* $\mathbf{S} \in \mathbb{R}^{n\times s}$ *such that with probability* $1-\delta$, $s = O\left(\sum_i p_i\right)$ *and the approximation* $\tilde{\mathbf{K}} = \mathbf{K}\mathbf{S}(\mathbf{S}\mathbf{K}\mathbf{S})^+\mathbf{S}\mathbf{K}$ *satisfies, for any rank* k *orthogonal projection* $\mathbf{X}$ *and a positive constant* c *independent of* $\mathbf{X}$*:*

$$\operatorname{tr}(\mathbf{K} - \mathbf{X}\mathbf{K}\mathbf{X}) \leq \operatorname{tr}(\tilde{\mathbf{K}} - \mathbf{X}\tilde{\mathbf{K}}\mathbf{X}) + c \leq (1+\epsilon)\operatorname{tr}(\mathbf{K} - \mathbf{X}\mathbf{K}\mathbf{X}). \tag{6}$$

When ridge leverage scores are computed exactly, $\sum_i p_i = O\left(\frac{k}{\epsilon}\log\frac{k}{\delta\epsilon}\right)$.

Intuitively, Theorem 3 ensures that the distance from $\tilde{\mathbf{K}}$ to any low dimensional subspace closely approximates the distance from $\mathbf{K}$ to the subspace. Accordingly, $\tilde{\mathbf{K}}$ can be used in place of $\mathbf{K}$ to approximately solve low-rank approximation problems, both constrained (e.g. k-means clustering) and unconstrained (e.g. principal component analysis). See Theorems 11 and 12.

Proof. Set $c = \operatorname{tr}(\mathbf{K}) - \operatorname{tr}(\tilde{\mathbf{K}})$, which is ≥ 0 since $\tilde{\mathbf{K}} \preceq \mathbf{K}$ by Theorem 2. By linearity of trace:

$$\operatorname{tr}(\tilde{\mathbf{K}} - \mathbf{X}\tilde{\mathbf{K}}\mathbf{X}) + c = \operatorname{tr}(\mathbf{K}) - \operatorname{tr}(\mathbf{X}\tilde{\mathbf{K}}\mathbf{X}).$$

So to obtain (6) it suffices to show:

$$\operatorname{tr}(\mathbf{X}\mathbf{K}\mathbf{X}) - \epsilon \operatorname{tr}(\mathbf{K} - \mathbf{X}\mathbf{K}\mathbf{X}) \leq \operatorname{tr}(\mathbf{X}\tilde{\mathbf{K}}\mathbf{X}) \leq \operatorname{tr}(\mathbf{X}\mathbf{K}\mathbf{X}). \tag{7}$$

Since $\mathbf{X}$ is a rank k orthogonal projection we can write $\mathbf{X} = \mathbf{Q}\mathbf{Q}^T$ where $\mathbf{Q} \in \mathbb{R}^{n \times k}$ has orthonormal columns. Applying the cyclic property of the trace, and the spectral norm bound of Theorem 2:

$$\operatorname{tr}(\mathbf{X}\tilde{\mathbf{K}}\mathbf{X}) = \operatorname{tr}(\mathbf{Q}^T\tilde{\mathbf{K}}\mathbf{Q}) = \sum_{i=1}^{k} \mathbf{q}_i^T\tilde{\mathbf{K}}\mathbf{q}_i \leq \sum_{i=1}^{k} \mathbf{q}_i^T\mathbf{K}\mathbf{q}_i = \operatorname{tr}(\mathbf{Q}^T\mathbf{K}\mathbf{Q}) = \operatorname{tr}(\mathbf{X}\mathbf{K}\mathbf{X}).$$

This gives us the upper bound of (7). For the lower bound:

$$\operatorname{tr}(\mathbf{X}\tilde{\mathbf{K}}\mathbf{X}) = \sum_{i=1}^{k} \mathbf{q}_i^T\tilde{\mathbf{K}}\mathbf{q}_i \geq \sum_{i=1}^{k} \mathbf{q}_i^T\mathbf{K}\mathbf{q}_i - k\epsilon\lambda = \operatorname{tr}(\mathbf{X}\mathbf{K}\mathbf{X}) - k\epsilon\lambda. \tag{8}$$

Finally, $k\epsilon\lambda = \epsilon\sum_{i=k+1}^{n}\sigma_i(\mathbf{K}) \leq \epsilon\operatorname{tr}(\mathbf{K} - \mathbf{X}\mathbf{K}\mathbf{X})$ since $\operatorname{tr}(\mathbf{K}) = \sum_{i=1}^{n}\sigma_i(\mathbf{K})$ and $\operatorname{tr}(\mathbf{X}\mathbf{K}\mathbf{X}) \leq \sum_{i=1}^{k}\sigma_i(\mathbf{K})$ by the Eckart-Young theorem. Plugging into (8) gives (7), completing the proof.

Again we conclude by showing that s is not too large. As in the proof of Theorem 2, $s = O(\sum_i p_i)$ with probability $1-\delta$. When ridge leverage scores are computed exactly $\sum_i p_i = \sum l_i^{\epsilon\lambda}\log(\sum l_i^{\epsilon\lambda}/\delta)$.

$$\begin{aligned}
\sum_i l_i^{\epsilon\lambda} = \operatorname{tr}(\mathbf{K}(\mathbf{K}+\epsilon\lambda\mathbf{I})^{-1}) &\leq \frac{1}{\epsilon}\operatorname{tr}(\mathbf{K}(\mathbf{K}+\lambda\mathbf{I})^{-1}) \\
&= \frac{1}{\epsilon}\sum_{i=1}^{n}\frac{\sigma_i(\mathbf{K})}{\sigma_i(\mathbf{K}) + \frac{1}{k}\sum_{i=k+1}^{n}\sigma_i(\mathbf{K})} \\
&= \frac{1}{\epsilon}\left(\sum_{i=1}^{k}\frac{\sigma_i(\mathbf{K})}{\sigma_i(\mathbf{K}) + \frac{1}{k}\sum_{i=k+1}^{n}\sigma_i(\mathbf{K})} + \sum_{i=k+1}^{n}\frac{\sigma_i(\mathbf{K})}{\sigma_i(\mathbf{K}) + \frac{1}{k}\sum_{i=k+1}^{n}\sigma_i(\mathbf{K})}\right) \\
&\leq \frac{1}{\epsilon}\left(k + \sum_{i=k+1}^{n}\frac{\sigma_i(\mathbf{K})}{\frac{1}{k}\sum_{i=k+1}^{n}\sigma_i(\mathbf{K})}\right) = \frac{2k}{\epsilon}.
\end{aligned}$$

Accordingly, $\sum_i p_i = O\left(\frac{k}{\epsilon}\log\frac{k}{\delta\epsilon}\right)$ as desired. ∎

4 Efficient implementation of RLS-Nyström

Having established that RLS-Nyström can be used to well approximate $\mathbf{K}$, we next show how to very efficiently implement the high-level procedure described in Algorithm 1. We give two closely

related algorithms: The first applies when the ridge parameter λ is known – e.g. when applying Theorem 2 to approximating kernel ridge regression. The second applies when $\lambda = \frac{1}{k}\sum_{i=k+1}^{n}\sigma(\mathbf{K})$ for some rank parameter k – e.g. when applying Theorem 3 to kernel clustering and PCA. In this case we cannot directly compute λ, so it must be estimated.

Both algorithms are simple recursive methods, which sample uniformly from $\mathbf{K}$ and approximate the ridge leverage scores based on this sampling. In order to ensure accuracy, the uniform sample must be large – consisting of a constant fraction of the original data points. However, it is possible to recursively approximate the sample, leading to a runtime which is linear in n and polynomial in the sum of ridge leverage scores. Our recursive algorithms follow work in [CMM15] and [CLM+15] and should be thought of as adapting these earlier results to the more challenging kernel setting.

4.1 Ridge leverage score approximation via uniform sampling

We first show that uniform sampling a constant fraction of the data points yields good ridge leverage score estimates. We will focus first on the case when λ is known.

Lemma 4. *For any $\mathbf{B} \in \mathbb{R}^{n\times n}$ with $\mathbf{B}\mathbf{B}^T = \mathbf{K}$ and $\mathbf{S} \in \mathbb{R}^{n\times s}$ chosen by sampling each data point independently with probability $1/2$, let*

$$\tilde{l}_i^{\epsilon\lambda} = \mathbf{b}_i^T(\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{b}_i.$$

and $p_i = \min\{1, c\log(\sum \tilde{l}_i^{\epsilon\lambda}/\delta)\}$. Then with probability at least $1-\delta$:

1. $\tilde{l}_i^{\epsilon\lambda} \geq l_i^{\epsilon\lambda}$
2. $\sum_i p_i = O\left(c\log(\sum l_i^{\epsilon\lambda}/\delta)\cdot\sum_i l_i^{\epsilon\lambda}\right)$

The first condition ensures that we can form an RLS-Nyström approximation by sampling with the approximate ridge leverage scores $\tilde{l}_i^{\epsilon\lambda}$. The second ensures that this approximation will have up to constant factors the same size as if we used the true ridge leverage scores. Note that it is not obvious how to compute $\tilde{l}_i^{\epsilon\lambda} = \mathbf{b}_i^T(\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{b}_i$ without explicitly forming $\mathbf{B}$. We discuss how to do this in Section 4.2.

Proof. The first bound follows because $\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} \preceq \mathbf{B}^T\mathbf{B}$ so

$$\tilde{l}_i^{\epsilon\lambda} = \mathbf{b}_i^T(\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{b}_i \geq \mathbf{b}_i^T(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{b}_i = l_i^{\epsilon\lambda}.$$

So the challenge is showing the second bound. The proof closely follows proofs in [CLM+15] and [CMM15], and we refer the reader to Theorem 2 of [CLM+15] for details. We sketch the idea below.

We first argue that there exists a diagonal reweighting matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$, $\mathbf{0} \preceq \mathbf{W} \preceq \mathbf{I}$ such that for all i, $l_i^{\epsilon\lambda}(\mathbf{W}\mathbf{B}\mathbf{B}^T\mathbf{W}) \leq \alpha$ where $\alpha \stackrel{\text{def}}{=} \frac{1}{2}\cdot\frac{1}{c\log(\sum l_i^{\epsilon\lambda}/\delta)}$. This bound ensures that uniformly sampling rows with probability $1/2$ from the *reweighted kernel* $\mathbf{W}\mathbf{B}\mathbf{B}^T\mathbf{W}$ is a valid ridge leverage score sampling. Further, we can show $|\{i : \mathbf{W}_{i,i} < 1\}| = O\left(c\log(\sum l_i^{\epsilon\lambda}/\delta)\cdot\sum l_i^{\epsilon\lambda}\right)$ – that is, we do not need to reweight too many columns to achieve the ridge leverage score upper bound.

$\mathbf{W}$ is formed by iteratively considering any i with $l_i^{\epsilon\lambda}(\mathbf{W}\mathbf{B}\mathbf{B}^T\mathbf{W}) \geq \alpha$. Since $\epsilon\lambda > 0$, it is always possible to decrease the ridge leverage score to exactly α by decreasing $\mathbf{W}_{i,i}$ sufficiently.

It is clear from Definition 1 that decreasing the weight of one row of $\mathbf{B}$ will only increase the ridge leverage scores of other rows. So, any reweighted row will always have leverage score $\geq \alpha$.

Theorem 2 of [CLM+15] demonstrates rigorously that the leverage scores of these reweighted rows will in fact converge to α. Further, since $\mathbf{W} \preceq \mathbf{I}$, $\mathbf{WBB}^T\mathbf{W} \preceq \mathbf{BB}^T$ and it is not hard to show (see Lemma 16 in Appendix C) that:

$$\sum_i l_i^{\epsilon\lambda}(\mathbf{WBB}^T\mathbf{W}) \leq \sum_i l_i^{\epsilon\lambda}(\mathbf{BB}^T) \stackrel{\text{def}}{=} \sum_i l_i^{\epsilon\lambda}.$$

So the number of reweighted rows is at most $\frac{1}{\alpha}\sum_i l_i^{\epsilon\lambda} = O\left(c\log(\sum l_i^{\epsilon\lambda}/\delta)\cdot\sum l_i^{\epsilon\lambda}\right)$.

We can now bound $\sum_i p_i$. For any i that is reweighted by $\mathbf{W}$ we just trivially bound $p_i \leq 1$. Since $l_i^{\epsilon\lambda}(\mathbf{WBB}^T\mathbf{W}) \leq \frac{1}{2}\cdot\frac{1}{c\log(\sum l_i^{\epsilon\lambda}/\delta)}$ for all i, and since $\mathbf{S}$ samples each i with probability $1/2$, it is a valid ridge leverage score sampling, and by Lemma 14:

$$\frac{1}{2}(\mathbf{B}^T\mathbf{W}^2\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq (\mathbf{B}^T\mathbf{WSS}^T\mathbf{WB} + \epsilon\lambda\mathbf{I}) \preceq \frac{3}{2}(\mathbf{B}^T\mathbf{W}^2\mathbf{B} + \epsilon\lambda\mathbf{I}).$$

Hence:

$$\begin{aligned}\tilde{l}_i^{\epsilon\lambda} = \mathbf{b}_i^T(\mathbf{B}^T\mathbf{SS}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{b}_i &\leq \mathbf{b}_i^T(\mathbf{B}^T\mathbf{WSS}^T\mathbf{WB} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{b}_i \\ &\leq 2\mathbf{b}_i^T(\mathbf{B}^T\mathbf{W}^2\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{b}_i \\ &= 2l_i^{\epsilon\lambda}(\mathbf{WBB}^T\mathbf{W}).\end{aligned}$$

Again using the fact that $\mathbf{WBB}^T\mathbf{W} \preceq \mathbf{BB}^T$ and Lemma 16, $\sum_{\{i:\mathbf{W}_{i,i}=1\}}\tilde{l}_i^{\epsilon\lambda} \leq 2\sum_i l_i^{\epsilon\lambda}$. Overall:

$$\begin{aligned}\sum_i p_i &= \sum_{\{i:\mathbf{W}_{i,i}<1\}} p_i + \sum_{\{i:\mathbf{W}_{i,i}=1\}} p_i \\ &\leq |\{i:\mathbf{W}_{i,i}<1\}| + c\log\left(\sum l_i^{\epsilon\lambda}/\delta\right)\cdot\sum_i l_i^{\epsilon\lambda} = O\left(c\log\left(\sum l_i^{\epsilon\lambda}/\delta\right)\cdot\sum_i l_i^{\epsilon\lambda}\right).\end{aligned}$$

□

4.2 Efficient sampled ridge leverage score computation

In order to utilize Lemma 4 we must show how to efficiently compute

$$\tilde{l}_i^{\epsilon\lambda} = \mathbf{b}_i^T(\mathbf{B}^T\mathbf{SS}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{b}_i$$

without explicitly forming either $\mathbf{K}$ or $\mathbf{B}$. We show how to do this in the following lemma:

Lemma 5. *For any sampling matrix $\mathbf{S} \in \mathbb{R}^{n\times s}$, and any $\lambda, \epsilon > 0$:*

$$\tilde{l}_i^{\epsilon\lambda} \stackrel{\text{def}}{=} \mathbf{b}_i^T(\mathbf{B}^T\mathbf{SS}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{b}_i = \frac{1}{\epsilon\lambda}\left(\mathbf{K} - \mathbf{KS}\left(\mathbf{S}^T\mathbf{KS} + \epsilon\lambda\mathbf{I}\right)^{-1}\mathbf{S}^T\mathbf{K}\right)_{i,i}.$$

It follows that we can compute $\tilde{l}_i^{\epsilon\lambda}$ for all i in total time $O(ns^2)$ using just $O(ns)$ kernel evaluations.

Proof. Using the SVD write $\mathbf{S}^T\mathbf{B} = \bar{\mathbf{U}}\bar{\mathbf{\Sigma}}\bar{\mathbf{V}}^T$. $\bar{\mathbf{V}} \in \mathbb{R}^{n\times s}$ forms an orthonormal basis for the row span of $\mathbf{S}^T\mathbf{B}$. Let $\bar{\mathbf{V}}_\perp$ be span for the nullspace of $\mathbf{S}^T\mathbf{B}$. Then we can rewrite $\tilde{l}_i^{\epsilon\lambda}$ as:

$$\tilde{l}_i^{\epsilon\lambda} = \mathbf{b}_i^T\left(\mathbf{B}^T\mathbf{SS}^T\mathbf{B} + \epsilon\lambda\mathbf{I}\right)^{-1}\mathbf{b}_i = \mathbf{b}_i^T\left[\bar{\mathbf{V}}, \bar{\mathbf{V}}_\perp\right](\bar{\mathbf{\Sigma}}^2 + \epsilon\lambda\mathbf{I})^{-1}\left[\bar{\mathbf{V}}, \bar{\mathbf{V}}_\perp\right]^T\mathbf{b}_i.$$

Here we're abusing notation a bit by letting $\bar{\mathbf{\Sigma}}$ represent an $n \times n$ diagonal matrix whose first s entries are the singular values of $\mathbf{S}^T\mathbf{B}$ and whose remaining entries are all equal to 0. Now:

$$\tilde{l}_i^{\epsilon\lambda} = \mathbf{b}_i^T \left[\bar{\mathbf{V}}, \bar{\mathbf{V}}_\perp\right] (\bar{\mathbf{\Sigma}}^2 + \epsilon\lambda\mathbf{I})^{-1} \left[\bar{\mathbf{V}}, \bar{\mathbf{V}}_\perp\right]^T \mathbf{b}_i = \frac{1}{\epsilon\lambda}\mathbf{b}_i^T\bar{\mathbf{V}}_\perp^T\bar{\mathbf{V}}_\perp\mathbf{b}_i + \mathbf{b}_i^T\bar{\mathbf{V}}(\bar{\mathbf{\Sigma}}^2 + \epsilon\lambda\mathbf{I})^{-1}\bar{\mathbf{V}}^T\mathbf{b}_i^T \tag{9}$$

Focusing on the second term of (9),

$$\begin{aligned}\mathbf{b}_i^T\bar{\mathbf{V}}(\bar{\mathbf{\Sigma}}^2 + \epsilon\lambda\mathbf{I})^{-1}\bar{\mathbf{V}}^T\mathbf{b}_i &= \mathbf{b}_i^T\bar{\mathbf{V}}\frac{1}{\epsilon\lambda}\left(\mathbf{I} - \bar{\mathbf{\Sigma}}^2(\bar{\mathbf{\Sigma}}^2 + \epsilon\lambda\mathbf{I})^{-1}\right)\bar{\mathbf{V}}^T\mathbf{b}_i \\ &= \frac{1}{\epsilon\lambda}\mathbf{b}_i^T\bar{\mathbf{V}}\bar{\mathbf{V}}^T\mathbf{b}_i - \frac{1}{\epsilon\lambda}\mathbf{b}_i^T\bar{\mathbf{V}}\left(\bar{\mathbf{\Sigma}}^2(\bar{\mathbf{\Sigma}}^2 + \epsilon\lambda\mathbf{I})^{-1}\right)\bar{\mathbf{V}}^T\mathbf{b}_i \end{aligned} \tag{10}$$

Focusing on the second term of (10),

$$\begin{aligned}\mathbf{b}_i^T\bar{\mathbf{V}}\left(\bar{\mathbf{\Sigma}}^2(\bar{\mathbf{\Sigma}}^2 + \epsilon\lambda\mathbf{I})^{-1}\right)\bar{\mathbf{V}}^T\mathbf{b}_i &= \mathbf{b}_i^T\bar{\mathbf{V}}\bar{\mathbf{\Sigma}}\bar{\mathbf{U}}^T\bar{\mathbf{U}}(\bar{\mathbf{\Sigma}}^2 + \epsilon\lambda\mathbf{I})^{-1}\bar{\mathbf{U}}^T\bar{\mathbf{U}}\bar{\mathbf{\Sigma}}\bar{\mathbf{V}}^T\mathbf{b}_i^T \\ &= \mathbf{b}_i^T\mathbf{B}^T\mathbf{S}(\mathbf{S}^T\mathbf{K}\mathbf{S} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{S}^T\mathbf{B}\mathbf{b}_i.\end{aligned}$$

Substituting back into (10) and then (9), we conclude that:

$$\begin{aligned}\tilde{l}_i^{\epsilon\lambda} &= \frac{1}{\epsilon\lambda}\mathbf{b}_i^T\bar{\mathbf{V}}_\perp^T\bar{\mathbf{V}}_\perp\mathbf{b}_i + \frac{1}{\epsilon\lambda}\mathbf{b}_i^T\bar{\mathbf{V}}\bar{\mathbf{V}}^T\mathbf{b}_i - \frac{1}{\epsilon\lambda}\mathbf{b}_i\mathbf{B}^T\mathbf{S}(\mathbf{S}^T\mathbf{K}\mathbf{S} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{S}^T\mathbf{B}\mathbf{b}_i \\ &= \frac{1}{\epsilon\lambda}\mathbf{b}_i^T\mathbf{b}_i - \frac{1}{\epsilon\lambda}\mathbf{b}_i^T\mathbf{B}^T\mathbf{S}(\mathbf{S}^T\mathbf{K}\mathbf{S} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{S}^T\mathbf{B}\mathbf{b}_i \\ &= \frac{1}{\epsilon\lambda}\mathbf{K}_{i,i} - \frac{1}{\epsilon\lambda}\left(\mathbf{K}\mathbf{S}\left(\mathbf{S}^T\mathbf{K}\mathbf{S} + \epsilon\lambda\mathbf{I}\right)^{-1}\mathbf{S}^T\mathbf{K}\right)_{i,i}.\end{aligned}$$

We can compute $(\mathbf{S}^T\mathbf{K}\mathbf{S}+\epsilon\lambda\mathbf{I})^{-1}$ in $O(s^3) = O(ns^2)$ time and $O(s^2) = O(ns)$ kernel evaluations. Given this inverse, computing the diagonal entries of $\mathbf{K}\mathbf{S}\left(\mathbf{S}^T\mathbf{K}\mathbf{S} + \epsilon\lambda\mathbf{I}\right)^{-1}\mathbf{S}^T\mathbf{K}$ requires just $O(ns)$ kernel evaluations to form $\mathbf{K}\mathbf{S}$ and $O(ns^2)$ time to perform the necessary multiplications. Finally, computing the diagonal entries of $\mathbf{K}$ requires n additional kernel evaluations. □

4.3 Basic recursive RLS-Nyström algorithm

We are finally ready to combine the above results into an efficient recursive method for ridge leverage score approximation.

Algorithm 2 Recursive Ridge Leverage Score Approximation. Known λ.

input: $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, ridge parameter λ, $\epsilon, \delta \in (0, 1)$

output: weighted sampling matrix $\mathbf{S} \in \mathbb{R}^{n\times s}$ such that for any $\mathbf{B}$ with $\mathbf{B}\mathbf{B}^T = \mathbf{K}$, $\frac{1}{2}(\mathbf{B}^T\mathbf{B}+\epsilon\lambda\mathbf{I}) \preceq (\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq \frac{3}{2}(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})$.

1: Choose $\mathbf{S}_0$ by sampling each data point independently with probability $1/2$.

2: If $\mathbf{S}_0$ has $> O(1)$ columns, apply Algorithm 2 **recursively** to $\mathbf{S}_0^T\mathbf{K}\mathbf{S}_0$ with $\delta \leftarrow \delta/2$ to compute an $\mathbf{S}_1$ such that:

$$\frac{1}{2}(\mathbf{B}^T\mathbf{S}_0\mathbf{S}_0^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq (\mathbf{B}^T\mathbf{S}_1\mathbf{S}_1^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq \frac{3}{2}(\mathbf{B}^T\mathbf{S}_0\mathbf{S}_0^T\mathbf{B} + \epsilon\lambda\mathbf{I}).$$

3: $\tilde{l}_i^{\epsilon\lambda} := \frac{2}{\epsilon\lambda}\left(\mathbf{K} - \mathbf{K}\mathbf{S}_1\left(\mathbf{S}_1^T\mathbf{K}\mathbf{S}_1 + \epsilon\lambda\mathbf{I}\right)^{-1}\mathbf{S}_1^T\mathbf{K}\right)_{i,i}$ ▷ Equals $(\mathbf{B}(\mathbf{B}^T\mathbf{S}_1\mathbf{S}_1^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{B}^T)_{i,i}$

4: $p_i := \min\{1, \tilde{l}_i^{\epsilon\lambda} \cdot c\log(\sum \tilde{l}_i^{\epsilon\lambda}/\delta)\}$

5: **return** $\mathbf{S}$ chosen by sampling i with probability p_i and reweighting selected columns by $1/\sqrt{p_i}$.

Theorem 6. *Let $s = O\left(\log(\sum l_i^{\epsilon\lambda}/\delta) \cdot \sum l_i^{\epsilon\lambda}\right)$. Algorithm 2 performs $O(n \cdot s)$ kernel evaluations and runs in time $O(n \cdot s^2)$. With probability $1 - 2\delta$ it returns $\mathbf{S} \in \mathbb{R}^{n \times s}$ satisfying, for any $\mathbf{B}$ with $\mathbf{B}\mathbf{B}^T = \mathbf{K}$:*

$$\frac{1}{2}(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq (\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq \frac{3}{2}(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})$$

Proof. Assume by induction that after forming $\mathbf{S}_0$ via uniformly sampling, the recursive call to Algorithm 2 returns $\mathbf{S}_1$ satisfying:

$$\frac{1}{2}(\mathbf{B}^T\mathbf{S}_0\mathbf{S}_0^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq (\mathbf{B}^T\mathbf{S}_1\mathbf{S}_1^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq \frac{3}{2}(\mathbf{B}^T\mathbf{S}_0\mathbf{S}_0^T\mathbf{B} + \epsilon\lambda\mathbf{I}).$$

Then for all i, $\tilde{l}_i^{\epsilon\lambda}$ will be within a constant factor of the approximate leverage score computed using $\mathbf{S}_0$ instead of $\mathbf{S}_1$. So if we sample by these scores, by Lemmas 4 and 14 we have the desired bound. By a union bound, the failure probability of the recursive algorithm is bounded by $\delta + \delta/2 + \ldots = 2\delta$.

To evaluate total runtime, let n_i be the number of points passed to the i^{th} recursive call of Algorithm 2. Applying Lemma 5, the total runtime to compute $\mathbf{S}$, excluding the recursive computation of $\mathbf{S}_1$, is $O(n_i s_i)$ kernel evaluations and $O(n_i s_i^2 + s_i^3) = O(n_i s_i^2)$ where $s_i \leq n_i$ is the number of points sampled by $\mathbf{S}_1$. By induction,

$$s_i = O\left(\log(\sum l_i^{\epsilon\lambda}(\mathbf{S}_0^T\mathbf{B}\mathbf{B}^T\mathbf{S}_0)/(\delta/2^i)) \cdot \sum l_i^{\epsilon\lambda}(\mathbf{S}_0^T\mathbf{B}\mathbf{B}^T\mathbf{S}_0)\right) = O(i \cdot s)$$

since $\|\mathbf{S}_0\|_2 \leq 1$ so we can apply Lemma 17. Since $n_i = O(n/2^i)$ in expectation and with high probability, our total runtime is $O(ns^2 \cdot (1 + 1/2 + 2^2/4 + 3^2/8 + \ldots + \log^2 n/n)) = O(ns^2)$ by the fact that $\sum_{i=1}^{\infty} \frac{i^2}{2^i} = 6$. ☐

In the final stage of Algorithm 2, we can use $\mathbf{S}$ directly to form a Nyström approximation of $\mathbf{K}$. Note that in this stage, we can just take $\mathbf{S}$ to have 0, 1 entries – reweighting of sampled points will not change the Nyström approximation. Since Theorem 6 gives that:

$$\frac{1}{2}(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq (\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq \frac{3}{2}(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I}),$$

even if $\mathbf{S}$ is unweighted, there is some C such that $\mathbf{B}^T\mathbf{B} \preceq C \cdot \mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I}$ (see Corollary 15). This bound in turn implies that $\mathbf{S}$ satisfies the guarantees of RLS-Nyström (see Theorem 2).

Corollary 7 (Efficient RLS-Nyström for Fixed λ)**.** *Algorithm 2 implements RLS-Nyström for a fixed λ with $O(n \cdot s)$ kernel evaluations and $O(n \cdot s^2)$ computation time, where $s = O\left(\frac{d_{\text{eff}}}{\epsilon} \log \frac{d_{\text{eff}}}{\delta\epsilon}\right)$ for $d_{\text{eff}} = \text{tr}(\mathbf{K}(\mathbf{K} + \lambda\mathbf{I})^{-1})$.*

Proof. The runtime bound follows from Theorem 6 and the fact that it is possible to compute $\mathbf{K}\mathbf{S}$ using $O(n \cdot s)$ kernel evaluations and $(\mathbf{S}^T\mathbf{K}\mathbf{S})^+$ using $O(ns^2 + s^3) = O(n \cdot s^2)$ additional time. ☐

4.4 Recursive RLS-Nyström algorithm for fixed k

In order to obtain a rank k projection-cost preserving kernel embedding as in Theorem 3, we must set $\lambda = \frac{1}{k}\sum_{i=k+1}^{n} \sigma_i(\mathbf{K})$. Of course, it is not possible to compute λ directly – instead we will approximate it in conjunction with approximating the ridge leverage scores.

Algorithm 3 RECURSIVE RIDGE LEVERAGE SCORE APPROXIMATION. FIXED RANK k.

input: $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, rank parameter k, $\epsilon, \delta \in (0,1)$
output: sampling matrix $\mathbf{S} \in \mathbb{R}^{n \times s}$ such that for any $\mathbf{B}$ with $\mathbf{B}\mathbf{B}^T = \mathbf{K}$, for $\lambda = \frac{1}{k}\sum_{i=k+1}^{n} \sigma_i(\mathbf{K})$,

$$\frac{1}{2}(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq (\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq \frac{3}{2}(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I}).$$

1: Choose $\mathbf{S}_0$ by sampling each data point independently with probability $1/2$.
2: If $\mathbf{S}_0$ has $> O\left(ck\log(k/\epsilon\delta)/\epsilon\right)$ columns, apply Algorithm 3 **recursively** to $\mathbf{S}_0^T\mathbf{K}\mathbf{S}_0$ with $\delta \leftarrow \delta/2$ to compute an $\mathbf{S}_1$ such that for $\lambda' = \frac{1}{k}\sum_{i=k+1}^{n} \sigma_i(\mathbf{S}_0^T\mathbf{K}\mathbf{S}_0)$,

$$\frac{1}{2}(\mathbf{B}^T\mathbf{S}_0\mathbf{S}_0^T\mathbf{B} + \epsilon\lambda'\mathbf{I}) \preceq (\mathbf{B}^T\mathbf{S}_1\mathbf{S}_1^T\mathbf{B} + \epsilon\lambda'\mathbf{I}) \preceq \frac{3}{2}(\mathbf{B}^T\mathbf{S}_0\mathbf{S}_0^T\mathbf{B} + \epsilon\lambda'\mathbf{I}).$$

3: $\tilde{\lambda} := \frac{1}{k}\sum_{i=k+1}^{n} \sigma_i(\mathbf{S}_1^T\mathbf{K}\mathbf{S}_1)$ ▷ Approximate λ
4: $\tilde{l}_i^{\epsilon\lambda} := \frac{5}{\epsilon\lambda}\left(\mathbf{K} - \mathbf{K}\mathbf{S}_1\left(\mathbf{S}_1^T\mathbf{K}\mathbf{S}_1 + \epsilon\tilde{\lambda}\mathbf{I}\right)^{-1}\mathbf{S}_1^T\mathbf{K}\right)_{i,i}$ ▷ Equals $(\mathbf{B}(\mathbf{B}^T\mathbf{S}_1\mathbf{S}_1^T\mathbf{B} + \epsilon\tilde{\lambda}\mathbf{I})^{-1}\mathbf{B}^T)_{i,i}$
5: $p_i := \min\{1, \tilde{l}_i^{\lambda} \cdot c\log(k/\delta)\}$
6: **return** $\mathbf{S}$ chosen by sampling i with probability p_i and reweighting selected columns by $1/\sqrt{p_i}$.

Theorem 8. *Let $s = O\left(\frac{k\log(k/\epsilon\delta)}{\epsilon}\right)$. Algorithm 3 performs $O(n \cdot s)$ kernel evaluations and runs in time $O(n \cdot s^2)$. With probability $1 - 2\delta$ it returns $\mathbf{S} \in \mathbb{R}^{n \times s}$ satisfying, for any $\mathbf{B}$ with $\mathbf{B}\mathbf{B}^T = \mathbf{K}$:*

$$\frac{1}{2}(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq (\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq \frac{3}{2}(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \tag{11}$$

for $\lambda = \frac{1}{k}\sum_{i=k+1}^{n} \sigma_i(\mathbf{K})$.

Proof. Assume by induction that after forming $\mathbf{S}_0$ via uniformly sampling, the recursive call to Algorithm 3 returns $\mathbf{S}_1$ satisfying:

$$\frac{1}{2}(\mathbf{B}^T\mathbf{S}_0\mathbf{S}_0^T\mathbf{B} + \epsilon\lambda'\mathbf{I}) \preceq (\mathbf{B}^T\mathbf{S}_1\mathbf{S}_1^T\mathbf{B} + \epsilon\lambda'\mathbf{I}) \preceq \frac{3}{2}(\mathbf{B}^T\mathbf{S}_0\mathbf{S}_0^T\mathbf{B} + \epsilon\lambda'\mathbf{I}). \tag{12}$$

where $\lambda' = \frac{1}{k}\sum_{i=k+1}^{n} \sigma_i(\mathbf{S}_0^T\mathbf{K}\mathbf{S}_0)$. This implies that $\tilde{\lambda} = \frac{1}{k}\sum_{i=k+1}^{n} \sigma_i(\mathbf{S}_1^T\mathbf{K}\mathbf{S}_1)$ satisfies:

$$\frac{1}{2k}\left(\sum_{i=k+1}^{n} \sigma_i(\mathbf{S}_0^T\mathbf{K}\mathbf{S}_0) + k\epsilon\lambda'\right) \leq \tilde{\lambda} \leq \frac{3}{2k}\left(\sum_{i=k+1}^{n} \sigma_i(\mathbf{S}_0^T\mathbf{K}\mathbf{S}_0) + k\epsilon\lambda'\right)$$

$$\frac{1+\epsilon}{2}\lambda' \leq \tilde{\lambda} \leq \frac{3(1+\epsilon)}{2}\lambda'.$$

Combining with (12) and the fact that $\epsilon \in (0,1)$:

$$\frac{1}{4}(\mathbf{B}^T\mathbf{S}_0\mathbf{S}_0^T\mathbf{B} + \epsilon\lambda'\mathbf{I}) \preceq (\mathbf{B}^T\mathbf{S}_1\mathbf{S}_1^T\mathbf{B} + \epsilon\tilde{\lambda}\mathbf{I}) \preceq \frac{9}{2}(\mathbf{B}^T\mathbf{S}_0\mathbf{S}_0^T\mathbf{B} + \epsilon\lambda'\mathbf{I}).$$

So, for all i, $\tilde{l}_i^{\epsilon\lambda}$ (which is computed using $(\mathbf{B}^T\mathbf{S}_1\mathbf{S}_1^T\mathbf{B} + \epsilon\tilde{\lambda}\mathbf{I})$ in line 4 of Algorithm 3) is within a constant factor of the approximate leverage score computed using $\mathbf{S}_0$ instead of $\mathbf{S}_1$. If we sample by these scores, by Lemma 4 and Lemma 14:

$$\frac{1}{2}(\mathbf{B}^T\mathbf{B} + \epsilon\lambda'\mathbf{I}) \preceq (\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda'\mathbf{I}) \preceq \frac{3}{2}(\mathbf{B}^T\mathbf{B} + \epsilon\lambda'\mathbf{I})$$

which implies (11) since $\lambda' \leq \lambda$ since $\|\mathbf{S}_0\|_2 \leq 1$ so $\sigma_i(\mathbf{S}_0^T\mathbf{K}\mathbf{S}_0) \leq \sigma_i(\mathbf{K})$ for all i.

It just remains to show that we do not sample too many rows. This can be shown using a similar reweighting argument to that used in the known λ case in Lemma 4. See Lemma 13 of [CMM15] for a full proof. Roughly, when forming the reweighting matrix $\mathbf{W}$, decreasing $\mathbf{W}_{i,i}$ will decrease $\sum_{i=k+1}^{n}\sigma_i(\mathbf{W}\mathbf{B}\mathbf{B}^T\mathbf{W})$ and hence will decrease λ. However, it is not hard to show that the i^{th} ridge leverage score will still decrease. So we can find $\mathbf{W}$ giving a uniform ridge leverage score upper bound of α. Let $\lambda' = \sum_{i=k+1}^{n}\sigma_i(\mathbf{W}\mathbf{B}\mathbf{B}^T\mathbf{W})$.

Using the same argument as Lemma 4, we can bound the sum of estimated sampling probabilities by $\log(k/\epsilon\delta)\cdot\sum l_i^{\epsilon\lambda'}(\mathbf{W}\mathbf{B}\mathbf{B}^T\mathbf{W}) = O(k/\epsilon)$ by the argument in Theorem 3. The runtime and failure probability analysis is identical to that of Algorithm 2 – the only extra step is computing $\tilde{\lambda}$ which can be done in $O(s^3)$ time via an SVD of $\mathbf{S}_1^T\mathbf{K}\mathbf{S}_1$. □

As in Algorithm 2, in the final stage of Algorithm 3 we can use $\mathbf{S}$ directly to form a Nyström approximation of $\mathbf{K}$. Since Theorem 8 gives:

$$\frac{1}{2}(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq (\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I}) \preceq \frac{3}{2}(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})$$

even if $\mathbf{S}$ is unweighted, there is some C such that $\mathbf{B}^T\mathbf{B} \preceq C\cdot\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I}$ (see Corollary 15). This bound in turn implies that $\mathbf{S}$ satisfies the guarantees of RLS-Nyström (see Theorem 2).

Corollary 9 (Efficient RLS-Nyström for Fixed k)**.** *Algorithm 3 implements RLS-Nyström for* $\lambda = \frac{1}{k}\sum_{i=k+1}^{n}\sigma_i(\mathbf{K})$ *with* $O(n\cdot s)$ *kernel evaluations and* $O(n\cdot s^2)$ *computation time, where* $s = O\left(k\log(k/\epsilon\delta)/\epsilon\right)$.

Proof. The runtime bound follows from Theorem 8 and the fact that it is possible to compute $\mathbf{K}\mathbf{S}$ using $O(n\cdot s)$ kernel evaluations and $(\mathbf{S}^T\mathbf{K}\mathbf{S})^+$ using $O(ns^2 + s^3) = O(n\cdot s^2)$ additional time. □

Acknowledgements

We would like to thank Michael Mahoney for bringing the potential of ridge leverage scores to our attention and suggesting their possible approximation via iterative sampling schemes. We would also like to thank Michael Cohen for pointing out (and fixing) an error in our original manuscript and generally for his close collaboration in our work on leverage score sampling algorithms. Finally, thanks to Haim Avron for pointing our an error in our original analysis.

References

[AM15] Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 775–783, 2015.

[AMS01] Dimitris Achlioptas, Frank Mcsherry, and Bernhard Schölkopf. Sampling techniques for kernel methods. In *Advances in Neural Information Processing Systems 14 (NIPS)*, 2001.

[And09] Alexandr Andoni. *Nearest neighbor search: the old, the new, and the impossible.* PhD thesis, Massachusetts Institute of Technology, 2009.

[ANW14] Haim Avron, Huy Nguyen, and David Woodruff. Subspace embeddings for the polynomial kernel. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 2258–2266, 2014.

[Bac13] Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In *Proceedings of the 26th Annual Conference on Computational Learning Theory (COLT)*, 2013.

[BBV06] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. Kernels as features: On kernels, margins, and low-dimensional mappings. *Mach. Learn.*, 65(1):79–94, 2006.

[BMD09] Christos Boutsidis, Michael W. Mahoney, and Petros Drineas. Unsupervised feature selection for the k-means clustering problem. In *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 153–161, 2009.

[BW09] Mohamed-Ali Belabbas and Patrick J. Wolfe. Spectral methods in machine learning: New strategies for very large datasets. *Proceedings of the National Academy of Sciences of the USA*, 106:369–374, 2009.

[BWZ16] Christos Boutsidis, David P. Woodruff, and Peilin Zhong. Optimal principal component analysis in distributed and streaming models. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, 2016.

[CEM+15] Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 163–172, 2015.

[CLL+15] Shouyuan Chen, Yang Liu, Michael Lyu, Irwin King, and Shengyu Zhang. Fast relative-error approximation algorithm for ridge regression. In *Proceedings of the 31st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 201–210, 2015.

[CLM+15] Michael B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 6th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 181–190, 2015.

[CMM15] Michael B. Cohen, Cameron Musco, and Christopher Musco. Ridge leverage scores for low-rank approximation. *arXiv:1511.07263*, 2015.

[CMT10] Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the impact of kernel approximation on learning accuracy. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 113–120, 2010.

[CP16] Di Chen and Jeff M. Phillips. Relative error embeddings for the gaussian kernel distance. *arXiv:1602.05350*, 2016.

[DM05] Petros Drineas and Michael W Mahoney. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *The Journal of Machine Learning Research*, 6:2153–2175, 2005.

[FS02] Shai Fine and Katya Scheinberg. Efficient SVM training using low-rank kernel representations. *The Journal of Machine Learning Research*, 2:243–264, 2002.

[FSS13] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1434–1453, 2013.

[Git11] Alex Gittens. The spectral norm error of the naive nystrom extension. *arXiv:1110.5305*, 2011.

[GLPW15] Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent Directions: Simple and deterministic matrix sketching. *arXiv:1501.01711*, 2015.

[GM13] Alex Gittens and Michael Mahoney. Revisiting the Nyström method for improved large-scale machine learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 567–575, 2013. Full version at arXiv:1303.1849.

[GPP16] Mina Ghashami, Daniel Perry, and Jeff M. Phillips. Streaming kernel principal component analysis. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

[HTF02] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2002.

[KMT12] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the Nyström method. *The Journal of Machine Learning Research*, 13:981–1006, 2012.

[LJS16] Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Fast DPP sampling for nyström with application to kernel methods. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

[PD15] Saurabh Paul and Petros Drineas. Feature selection for ridge regression with provable guarantees. *arXiv:1506.05173*, 2015.

[RCR15] Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 1648–1656, 2015.

[RR07] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 1177–1184, 2007.

[SS00] Alex J Smola and Bernhard Schökopf. Sparse greedy matrix approximation for machine learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pages 911–918, 2000.

[SS02] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[SSM99] Bernhard Schölkopf, Alexander J. Smola, and Klaus-Robert Müller. Advances in kernel methods. chapter Kernel Principal Component Analysis, pages 327–352. MIT Press, 1999.

[Tro15] Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015.

[Wan16] Weiran Wang. On column selection in approximate kernel canonical correlation analysis. *arXiv:1602.02172*, 2016.

[Woo14] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2):1–157, 2014.

[WS01] Christopher Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 14 (NIPS)*, pages 682–688, 2001.

[WZ13] Shusen Wang and Zhihua Zhang. Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling. *The Journal of Machine Learning Research*, 14:2729–2769, 2013.

[YPW15] Yun Yang, Mert Pilanci, and Martin J Wainwright. Randomized sketches for kernels: Fast and optimal non-parametric regression. *Annals of Statistics*, 2015.

[YZ13] Martin Wainwright Yuchen Zhang, John Duchi. Divide and conquer kernel ridge regression. *Proceedings of the 26th Annual Conference on Computational Learning Theory (COLT)*, 2013.

[ZTK08] Kai Zhang, Ivor W. Tsang, and James T. Kwok. Improved nyström low-rank approximation and error analysis. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 1232–1239, 2008.

A Applications of kernel approximations

In this section we prove that the kernel approximation guarantees given by RLS-Nyström sampling are sufficient for many downstream kernel learning tasks. In other words, $\tilde{\mathbf{K}}$ can be used in place of $\mathbf{K}$ without sacrificing accuracy or statistical performance in the final computation.

A.1 Kernel ridge regression

We begin with a standard formulation of kernel ridge regression. Given a response space $\mathcal{Y}$, a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, and a regularization parameter $\lambda \in \mathbb{R}$, consider minimizing:

$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_{\mathcal{F}}^2.$$

We focus on the special case where $\mathcal{Y} = \mathbb{R}$ and ℓ is the square loss $(y_i - f(\mathbf{x}_i))^2$. By the representer theorem, f^* can be written as $f^* = \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i)$ [SS02]. Letting $\mathbf{y} \in \mathbb{R}^n$ contain the responses

$y_1, \ldots, y_n$ and $\boldsymbol{\alpha} \in \mathbb{R}^n$ contain the coefficients $\alpha_1, \ldots, \alpha_n$ for representing f^*, it is well known that

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y}.$$

Once $\boldsymbol{\alpha}$ is obtained, it can be used directly for prediction:

$$f^*(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}). \tag{13}$$

Naively, solving for $\boldsymbol{\alpha}$ exactly requires at least $O(n^2)$ time to compute $\mathbf{K}$, plus the cost of a direct or iterative matrix inversion algorithm. Accordingly, a lot of research has focused on how to approximate $\mathbf{K}$ for use in kernel ridge regression. Of particular interest are two recent papers, [Bac13] and [AM15], which show how to bound the statistical risk of constructing an estimator for ridge regression based on a subsampled approximation of $\mathbf{K}$.

In particular, both papers consider the fixed design scenario and seek to bound the expected in-sample prediction error under the assumption that y_i satisfies:

$$y_i = f^*(\mathbf{x}_i) + \eta_i$$

where the noise terms $\eta_1, \ldots, \eta_n$ are distributed as normal random variables with variance σ^2. Let $\mathbf{z}$ denote the vector containing $f^*(\mathbf{x}_1), \ldots, f^*(\mathbf{x}_n)$ and let $\boldsymbol{\eta}$ denote the vector containing $\eta_1, \ldots, \eta_n$. Following, [Bac13] and [AM15], consider the expected risk of our estimator for $\mathbf{z}$, $\hat{f}_{\mathbf{K}} = \mathbf{K}\boldsymbol{\alpha}$:

$$\begin{aligned}
\mathcal{R}(\hat{f}_{\mathbf{K}}) &= \mathop{\mathbb{E}}_{\boldsymbol{\eta}} \|\mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1}(\mathbf{z} + \boldsymbol{\eta}) - \mathbf{z}\|_2^2 \\
&= \| \left(\mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1} - \mathbf{I}\right) \mathbf{z}\|_2^2 + \mathop{\mathbb{E}}_{\boldsymbol{\eta}} \|\mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1}\boldsymbol{\eta}\|_2^2 \\
&= \lambda^2 \mathbf{z}^T (\mathbf{K} + \lambda \mathbf{I})^{-2}\mathbf{z} + \sigma^2 \operatorname{tr}(\mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1}) \\
&\overset{\text{def}}{=} \operatorname{bias}(\mathbf{K})^2 + \operatorname{variance}(\mathbf{K}).
\end{aligned}$$

We refer the reader to [AM15] and [Bac13] for the above derivation. Note that our λ parameter is scaled differently than the λ in those papers by a factor of n. [Bac13] also uses a more general noise model, which we avoid for simplicity but note could be handled with essentially the same proof.

Theorem 10 (Kernel Ridge Regression Risk Bound)**.** *Suppose $\tilde{\mathbf{K}}$ is computed using RLS-Nyström with parameters λ, ϵ, and δ. Let $\tilde{\boldsymbol{\alpha}} = (\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1}\mathbf{y}$ and let $\hat{f}_{\tilde{\mathbf{K}}} = \tilde{\mathbf{K}}\tilde{\boldsymbol{\alpha}}$ be our estimator for $\mathbf{z}$ computed using the approximate kernel. Then with probability $(1 - \delta)$:*

$$\mathcal{R}(\hat{f}_{\tilde{\mathbf{K}}}) \leq (1 + 3\epsilon)\mathcal{R}(\hat{f}_{\mathbf{K}}).$$

By Corollary 7, Algorithm 2 can compute $\tilde{\mathbf{K}}$ with just $O(n \cdot s)$ kernel evaluations and $O(n \cdot s^2)$ computation time, with $s = O\left(\frac{d_{\text{eff}}}{\epsilon} \log \frac{d_{\text{eff}}}{\delta\epsilon}\right)$ for $d_{\text{eff}} = \operatorname{tr}(\mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1})$.

In other words, replacing $\mathbf{K}$ with the approximation $\tilde{\mathbf{K}}$ is provably sufficient for obtaining a $(1+\epsilon)$ quality solution to the downstream task of ridge regression. To use the approximate solution for prediction, we just multiply any new input point $\phi(\mathbf{x})$ by the vector

$$\boldsymbol{\Phi}^T \mathbf{S}(\mathbf{S}^T \mathbf{K} \mathbf{S})^+ \mathbf{S}^T \mathbf{K} \boldsymbol{\alpha}$$

which can be done with only $\mathbf{s}$ additional kernel evaluations and $O(n \cdot s)$ time.

Proof. The rest of the proof follows that of Theorem 1 in [AM15]. First we show that:

$$\text{bias}(\tilde{\mathbf{K}}) \leq (1+\epsilon)\text{bias}(\mathbf{K}). \tag{14}$$

At first glance this might appear trivial as Theorem 2 easily implies that

$$(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1} \preceq (1+\epsilon)(\mathbf{K} + \lambda \mathbf{I})^{-1}$$

However, this statement *does not imply* that

$$(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-2} \preceq (1+\epsilon)^2(\mathbf{K} + \lambda \mathbf{I})^{-2}$$

since $(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1}$ and $(\mathbf{K} + \lambda \mathbf{I})^{-1}$ do no necessarily commute. We highlight the issue because the incorrect conclusion is used in several papers to prove risk bounds for approximating ridge regression. A rigorous proof requires a bit more work.

$$\begin{aligned}
\frac{1}{\lambda}\text{bias}(\tilde{\mathbf{K}}) &= \|(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1}\mathbf{z}\|_2 \\
&\leq \|(\mathbf{K} + \mathbf{I})^{-1}\mathbf{z}\|_2 + \|(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1}\mathbf{z} - (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{z}\|_2 && \text{(triangle inequality)} \\
&= \|(\mathbf{K} + \mathbf{I})^{-1}\mathbf{z}\|_2 + \|(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1}(\mathbf{K} - \tilde{\mathbf{K}})(\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{z}\|_2 \\
&\leq \|(\mathbf{K} + \mathbf{I})^{-1}\mathbf{z}\|_2 + \|(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1}(\mathbf{K} - \tilde{\mathbf{K}})\|_2\|(\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{z}\|_2 && \text{(submultiplicativity)} \\
&= \frac{1}{\lambda}\text{bias}(\mathbf{K})\left(1 + \|(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1}(\mathbf{K} - \tilde{\mathbf{K}})\|_2\right).
\end{aligned} \tag{15}$$

So we just need to bound $\|(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1}(\mathbf{K} - \tilde{\mathbf{K}})\|_2 \leq \epsilon$. First note that, by Theorem 2 ,

$$\mathbf{K} - \tilde{\mathbf{K}} \preceq \epsilon\lambda \mathbf{I}$$

and since $(\mathbf{K} - \tilde{\mathbf{K}})$ and $\mathbf{I}$ *commute*, it follows that

$$(\mathbf{K} - \tilde{\mathbf{K}})^2 \preceq \epsilon^2\lambda^2 \mathbf{I}. \tag{16}$$

Accordingly,

$$\begin{aligned}
\|(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1}(\mathbf{K} - \tilde{\mathbf{K}})\|_2^2 &= \|(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1}(\mathbf{K} - \tilde{\mathbf{K}})^2(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1}\|_2 \\
&\leq \epsilon^2\lambda^2\|(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-2}\|_2 \\
&\leq \epsilon^2\lambda^2\frac{1}{\lambda^2} = \epsilon^2.
\end{aligned}$$

So $\|(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1}(\mathbf{K} - \tilde{\mathbf{K}})\|_2 \leq \epsilon$ as desired and plugging into (15) we have shown (14), that $\text{bias}(\tilde{\mathbf{K}}) \leq (1+\epsilon)\text{bias}(\mathbf{K})$. We next show:

$$\text{variance}(\tilde{\mathbf{K}}) \leq \text{variance}(\mathbf{K}). \tag{17}$$

$\text{variance}(\tilde{\mathbf{K}}) = \sigma^2 \operatorname{tr}(\tilde{\mathbf{K}}(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1})$. $\operatorname{tr}(\tilde{\mathbf{K}}(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1})$ is just the sum of λ ridge leverage scores. So we have (17) by Lemma 16 where we show, for $\tilde{\mathbf{K}} \preceq \mathbf{K}$

$$\operatorname{tr}(\tilde{\mathbf{K}}(\tilde{\mathbf{K}} + \lambda \mathbf{I})^{-1}) \leq \operatorname{tr}(\mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1}).$$

Combining (17) and (14) we conclude that, for $\epsilon < 1$,

$$\mathcal{R}(\hat{f}_{\tilde{\mathbf{K}}}) \leq (1+\epsilon)^2\mathcal{R}(\hat{f}_{\mathbf{K}}) \leq (1+3\epsilon)\mathcal{R}(\hat{f}_{\mathbf{K}}).$$

□

A.1.1 Additional error bounds

Bounding risk in the fixed design setting is one particularly natural and popular way to evaluate the quality of $\tilde{\mathbf{K}}$ for use in ridge regression (it is also applied in [LJS16] and [PD15]). However, we remark that prior work has considered a number of alternative bounds which may be interesting to the reader. In particular, [CLL+15], [YPW15], and [YZ13] directly bound $\|\hat{f}_{\tilde{\mathbf{K}}} - f^*\|$. [CMT10] gives bounds on the in-sample hypothesis error $|\hat{f}_{\mathbf{K}}(\mathbf{x}_i) - \hat{f}_{\tilde{\mathbf{K}}}(\mathbf{x}_i)|$ that would also follow from the guarantee of RLS-Nyström. Finally, [RCR15] considers statistical settings beyond fixed design regression. They also employ ridge leverage scores to compute $\tilde{\mathbf{K}}$ so their bounds would immediately transfer over to our more efficient algorithm.

A.2 Kernel k-means

Kernel k-means clustering asks us to partition $\mathbf{x}_1, \ldots, \mathbf{x}_n$, into k cluster sets, $\{C_1, \ldots, C_k\}$. Let $\boldsymbol{\mu}_i = \frac{1}{|C_i|}\sum_{\mathbf{x}_j \in C_i} \phi(\mathbf{x}_j)$ be the centroid of the vectors in C_i after mapping to kernel space. The goal is to choose $\{C_1, \ldots, C_k\}$ which minimize the objective:

$$\sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} \|\phi(\mathbf{x}_j) - \boldsymbol{\mu}_i\|_{\mathcal{F}}^2 \tag{18}$$

It is well known that this optimization problem can be rewritten as a *constrained* low-rank approximation problem (see e.g. [BMD09] or [CEM+15]). In particular, for any clustering $C = \{C_1, \ldots, C_k\}$ we can define a rank k orthonormal matrix $\mathbf{C} \in \mathbb{R}^{n \times k}$ called the cluster indicator matrix for C. $\mathbf{C}_{i,j} = 1/\sqrt{|C_j|}$ if $\mathbf{x}_i$ is assigned to C_j and $\mathbf{C}_{i,j} = 0$ otherwise. $\mathbf{C}^T\mathbf{C} = \mathbf{I}$, so $\mathbf{C}\mathbf{C}^T$ is a rank k projection matrix. Furthermore, it's not hard to check that:

$$\sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} \|\phi(\mathbf{x}_j) - \boldsymbol{\mu}_i\|_{\mathcal{F}}^2 = \operatorname{tr}\left(\mathbf{K} - \mathbf{C}\mathbf{C}^T\mathbf{K}\mathbf{C}\mathbf{C}^T\right). \tag{19}$$

Informally, if we work with the kernalized data matrix $\boldsymbol{\Phi}$, (19) is equivalent to

$$\|\boldsymbol{\Phi} - \mathbf{C}\mathbf{C}^T\boldsymbol{\Phi}\|_F^2.$$

Regardless, it's clear that solving kernel k-means is equivalent to solving:

$$\min_{\mathbf{C} \in \mathcal{S}} \operatorname{tr}\left(\mathbf{K} - \mathbf{C}\mathbf{C}^T\mathbf{K}\mathbf{C}\mathbf{C}^T\right) \tag{20}$$

where $\mathcal{S}$ is the set of all rank k cluster indicator matrices. From this formulation, we easily obtain:

Theorem 11 (Kernel k-means Approximation Bound)**.** *Let $\tilde{\mathbf{K}}$ be computed by RLS-Nyström with $\lambda = \frac{1}{k}\sum_{i=k+1}^{n}\sigma_i(\mathbf{K})$ and $\epsilon, \delta < 1$. Let $\tilde{\mathbf{C}}^*$ be the optimal cluster indicator matrix for $\tilde{\mathbf{K}}$ and let $\tilde{\mathbf{C}}$ be an approximately optimal cluster indicator matrix satisfying:*

$$\operatorname{tr}\left(\tilde{\mathbf{K}} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^T\tilde{\mathbf{K}}\tilde{\mathbf{C}}\tilde{\mathbf{C}}^T\right) \leq (1+\gamma)\operatorname{tr}\left(\tilde{\mathbf{K}} - \tilde{\mathbf{C}}^*\tilde{\mathbf{C}}^{*T}\tilde{\mathbf{K}}\tilde{\mathbf{C}}^*\tilde{\mathbf{C}}^{*T}\right).$$

Then, if $\mathbf{C}^*$ is the optimal cluster indicator matrix for $\mathbf{K}$:

$$\operatorname{tr}\left(\mathbf{K} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^T\mathbf{K}\tilde{\mathbf{C}}\tilde{\mathbf{C}}^T\right) \leq (1+\gamma)(1+\epsilon)\operatorname{tr}\left(\mathbf{K} - \mathbf{C}^*\mathbf{C}^{*T}\mathbf{K}\mathbf{C}^*\mathbf{C}^{*T}\right)$$

By Corollary 9, Algorithm 3 can compute $\tilde{\mathbf{K}}$ with $O(n \cdot s)$ kernel evaluations and $O(n \cdot s^2)$ computation time, with $s = O\left(\frac{k}{\epsilon}\log\frac{k}{\delta\epsilon}\right)$.

In other words, if we find an optimal set of clusters for our approximate kernel matrix, those clusters will provide a $(1+\epsilon)$ approximation to the original kernel k-means problem. Furthermore, if we only solve the kernel k-means problem approximately on $\tilde{\mathbf{K}}$, i.e. with some approximation factor $(1+\gamma)$, we will do nearly as well on the original problem. This flexibility allows for the use of k-means approximation algorithms (since the problem is NP-hard to solve exactly).

Proof. The proof is almost immediate from our bounds on RLS-Nyström:

$$\begin{aligned}
\operatorname{tr}\left(\mathbf{K}-\tilde{\mathbf{C}}\tilde{\mathbf{C}}^T\mathbf{K}\tilde{\mathbf{C}}\tilde{\mathbf{C}}^T\right) &\leq \operatorname{tr}\left(\tilde{\mathbf{K}}-\tilde{\mathbf{C}}\tilde{\mathbf{C}}^T\tilde{\mathbf{K}}\tilde{\mathbf{C}}\tilde{\mathbf{C}}^T\right)+c && \text{(Theorem 2)}\\
&\leq (1+\gamma)\operatorname{tr}\left(\tilde{\mathbf{K}}-\tilde{\mathbf{C}}^*\tilde{\mathbf{C}}^{*T}\tilde{\mathbf{K}}\tilde{\mathbf{C}}^*\tilde{\mathbf{C}}^{*T}\right)+(1+\gamma)c && \text{(by assumption)}\\
&\leq (1+\gamma)\operatorname{tr}\left(\tilde{\mathbf{K}}-\mathbf{C}^*\mathbf{C}^{*T}\tilde{\mathbf{K}}\mathbf{C}^*\mathbf{C}^{*T}\right)+(1+\gamma)c && \text{(optimality of } \tilde{\mathbf{C}}^* \text{)}\\
&\leq (1+\gamma)\operatorname{tr}\left(\tilde{\mathbf{K}}-\mathbf{C}^*\mathbf{C}^{*T}\tilde{\mathbf{K}}\mathbf{C}^*\mathbf{C}^{*T}\right)+c && \text{(since } c\geq 0\text{)}\\
&\leq (1+\gamma)(1+\epsilon)\operatorname{tr}\left(\mathbf{K}-\tilde{\mathbf{C}}^*\mathbf{C}^{*T}\mathbf{K}\mathbf{C}^*\mathbf{C}^{*T}\right). && \text{(Theorem 2)}
\end{aligned}$$

□

A.3 Kernel principal component analysis

We consider the standard formulation of kernel principal component analysis (PCA) presented in [SSM99]. The goal is to find principal components *in the kernel space* $\mathcal{F}$ that capture as much variance in the kernelized data as possible. In particular, if we work informally with the kernelized data matrix $\mathbf{\Phi}$, we want to find a matrix $\mathbf{Z}_k$ containing k orthonormal columns such that:

$$\mathbf{\Phi}\mathbf{\Phi}^T-(\mathbf{\Phi}\mathbf{Z}_k\mathbf{Z}_k^T)(\mathbf{\Phi}\mathbf{Z}_k\mathbf{Z}_k^T)^T$$

is as small as possible. In other words, if we project $\mathbf{\Phi}$'s rows to the k dimensional subspace spanned by $\mathbf{V}_k$'s columns and then recompute our kernel, we want the approximate kernel to be close to the original.

We focus in particular on minimizing PCA error according to the metric:

$$\operatorname{tr}\left(\mathbf{\Phi}\mathbf{\Phi}^T-(\mathbf{\Phi}\mathbf{Z}_k\mathbf{Z}_k^T)(\mathbf{\Phi}\mathbf{Z}_k\mathbf{Z}_k^T)^T\right)=\|\mathbf{\Phi}-\mathbf{\Phi}\mathbf{Z}_k\mathbf{Z}_k^T\|_F^2 \tag{21}$$

which is standard in the literature [Woo14, ANW14]. As with f in kernel ridge regression, to solve this problem we cannot write down $\mathbf{Z}_k$ explicitly for most kernel functions. However, the optimal $\mathbf{Z}_k$ always lies in the column span of $\mathbf{\Phi}^T$, so we can implicitly represent it by constructing a matrix $\mathbf{X}\in\mathbb{R}^{n\times k}$ such that $\mathbf{\Phi}^T\mathbf{X}=\mathbf{Z}_k$. It is then easy to compute the projection of any new data vector onto the span of $\mathbf{Z}_k$ (the typical objective of principal component analysis) since we can multiply by $\mathbf{\Phi}^T\mathbf{X}$ using the kernel function.

By the Eckart-Young theorem the optimal $\mathbf{Z}_k$ contains the top k row principal components of $\mathbf{\Phi}$. Accordingly, if we write the singular value decomposition $\mathbf{\Phi}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ we want to set $\mathbf{X}=\mathbf{U}_k\mathbf{\Sigma}_k^{-1}$, which can be computed from the SVD of $\mathbf{K}=\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T$. $\mathbf{Z}_k$ will equal $\mathbf{V}_k$ and (21) reduces to:

$$\begin{aligned}
\operatorname{tr}(\mathbf{K}-\mathbf{\Phi}\mathbf{V}_k\mathbf{V}_k^T\mathbf{\Phi}) &= \operatorname{tr}(\mathbf{K}-\mathbf{V}_k\mathbf{V}_k^T\mathbf{K}) && \text{(cyclic property)}\\
&= \sum_{i=k+1}^{n}\sigma_i(\mathbf{K}) && (22)
\end{aligned}$$

Theorem 12 (Kernel PCA Approximation Bound)**.** *Suppose* $\tilde{\mathbf{K}}$ *is computed by RLS-Nyström with* $\lambda = \frac{1}{k}\sum_{i=k+1}^{n}\sigma_i(\mathbf{K})$ *and* $\epsilon, \delta < 1$*. Then from* $\tilde{\mathbf{K}}$ *we can compute a matrix* $\mathbf{X} \in \mathbb{R}^{s\times k}$ *such that if we set* $\mathbf{Z} = \mathbf{\Phi}^T\mathbf{S}\mathbf{X}$*,*

$$\|\mathbf{\Phi} - \mathbf{\Phi}\mathbf{Z}\mathbf{Z}^T\|_F^2 \leq (1+2\epsilon)\|\mathbf{\Phi} - \mathbf{\Phi}\mathbf{V}_k\mathbf{V}_k^T\|_F^2 = (1+2\epsilon)\sum_{i=k+1}^{n}\sigma_i(\mathbf{K}).$$

By Corollary 9, Algorithm 3 can compute $\tilde{\mathbf{K}}$ *with* $O(n\cdot s)$ *kernel evaluations and* $O(n\cdot s^2)$ *computation time, with* $s = O\left(\frac{k}{\epsilon}\log\frac{k}{\delta\epsilon}\right)$*.*

Note that $\mathbf{S}$ is the sampling matrix used to construct $\tilde{\mathbf{K}}$. $\mathbf{Z} = \mathbf{\Phi}^T\mathbf{S}\mathbf{X}$ can be applied to vectors (in order to project onto the approximate low-rank subspace) using only s kernel evaluations.

Proof. Re-parameterizing $\mathbf{Z}_k = \mathbf{\Phi}^T\mathbf{Y}$, we see that minimizing (21) is equivalent to minimizing

$$\mathrm{tr}(\mathbf{K} - \mathbf{K}\mathbf{Y}\mathbf{Y}^T\mathbf{K})$$

over $\mathbf{Y} \in \mathbb{R}^{n\times k}$ such that $(\mathbf{\Phi}^T\mathbf{Y})^T\mathbf{\Phi}^T\mathbf{Y} = \mathbf{Y}^T\mathbf{K}\mathbf{Y} = \mathbf{I}$. Then we re-parameterize again by writing $\mathbf{Y} = \mathbf{K}^{-1/2}\mathbf{W}$ where $\mathbf{W}$ is an $n\times k$ matrix with orthonormal columns. Using linearity and cyclic property of the trace, we can write:

$$\mathrm{tr}(\mathbf{K} - \mathbf{K}\mathbf{Y}\mathbf{Y}^T\mathbf{K}) = \mathrm{tr}(\mathbf{K}) - \mathrm{tr}(\mathbf{Y}^T\mathbf{K}\mathbf{K}\mathbf{Y}) = \mathrm{tr}(\mathbf{K}) - \mathrm{tr}(\mathbf{W}^T\mathbf{K}\mathbf{W}) = \mathrm{tr}(\mathbf{K}) - \mathrm{tr}(\mathbf{W}\mathbf{W}^T\mathbf{K}\mathbf{W}\mathbf{W}^T).$$

So, we have reduced our problem to a low-rank approximation problem that looks exactly like the k-means problem from Section A.2, except without constraints.

Accordingly, following the same argument as Theorem 11, if we find $\tilde{\mathbf{W}}$ minimizing

$$\mathrm{tr}(\tilde{\mathbf{K}}) - \mathrm{tr}(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T\tilde{\mathbf{K}}\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T),$$

then

$$\mathrm{tr}(\mathbf{K}) - \mathrm{tr}(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T\mathbf{K}\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T) \leq (1+\epsilon)\left[\min_{\mathbf{W}}\mathrm{tr}(\mathbf{K}) - \mathrm{tr}(\mathbf{W}\mathbf{W}^T\mathbf{K}\mathbf{W}\mathbf{W}^T)\right] = (1+\epsilon)\sum_{i=k+1}^{n}\sigma_i(\mathbf{K}).$$

$\tilde{\mathbf{W}}$ can be taken to equal the top k eigenvectors of $\tilde{\mathbf{K}}$, which can be found in $O(n\cdot s^2)$ time.

However, we are not quite done. Thanks to our re-parameterization this bound guarantees that $\mathbf{\Phi}^T\mathbf{K}^{-1/2}\tilde{\mathbf{W}}$ is a good set of approximate kernel principal components for $\mathbf{\Phi}$. Unfortunately, $\mathbf{\Phi}^T\mathbf{K}^{-1/2}\tilde{\mathbf{W}}$ cannot be represented efficiently (it requires computing $\mathbf{K}^{-1/2}$) and projecting new vectors to $\mathbf{\Phi}^T\mathbf{K}^{-1/2}\tilde{\mathbf{W}}$ would require n kernel evaluations to multiply by $\mathbf{\Phi}^T$.

Instead, recalling the definition of $\mathbf{P}_\mathbf{S} = \mathbf{\Phi}^T\mathbf{S}(\mathbf{S}^T\mathbf{K}^T\mathbf{S})^+\mathbf{S}^T\mathbf{\Phi}$ from Section 3.1, we suggest using the approximate principal components:

$$\mathbf{P}_\mathbf{S}\mathbf{\Phi}^T\tilde{\mathbf{K}}^{-1/2}\tilde{\mathbf{W}}.$$

Clearly $\mathbf{P}_\mathbf{S}\mathbf{\Phi}^T\tilde{\mathbf{K}}^{-1/2}\tilde{\mathbf{W}}$ is orthonormal because

$$\begin{aligned}(\mathbf{P}_\mathbf{S}\mathbf{\Phi}^T\tilde{\mathbf{K}}^{-1/2}\tilde{\mathbf{W}})^T\mathbf{P}_\mathbf{S}\mathbf{\Phi}^T\tilde{\mathbf{K}}^{-1/2}\tilde{\mathbf{W}} &= \tilde{\mathbf{W}}^T\tilde{\mathbf{K}}^{-1/2}\mathbf{\Phi}^T\mathbf{P}_\mathbf{S}\mathbf{\Phi}\tilde{\mathbf{K}}^{-1/2}\tilde{\mathbf{W}}\\ &= \tilde{\mathbf{W}}^T\mathbf{I}\tilde{\mathbf{W}} = \mathbf{I}\end{aligned}$$

and we will argue that it is offers nearly as a good of a solution as $\mathbf{\Phi}^T\mathbf{K}^{-1/2}\tilde{\mathbf{W}}$. Specifically, substituting into (21) gives a value of

$$\begin{aligned}\mathrm{tr}(\mathbf{K} - \mathbf{\Phi}\mathbf{P_S}\mathbf{\Phi}^T\tilde{\mathbf{K}}^{-1/2}\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T\tilde{\mathbf{K}}^{-1/2}\mathbf{\Phi}\mathbf{P_S}\mathbf{\Phi}^T) &= \mathrm{tr}(\mathbf{K}) - \mathrm{tr}(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T\tilde{\mathbf{K}}^{-1/2}\mathbf{\Phi}\mathbf{P_S}\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{P_S}\mathbf{\Phi}^T\tilde{\mathbf{K}}^{-1/2}) \\ &= \mathrm{tr}(\mathbf{K}) - \mathrm{tr}(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T\tilde{\mathbf{K}}^{-1/2}\tilde{\mathbf{K}}^2\tilde{\mathbf{K}}^{-1/2}) \\ &= \mathrm{tr}(\mathbf{K}) - \mathrm{tr}(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T\tilde{\mathbf{K}})\end{aligned}$$

Compare this to the value obtained from $\mathbf{\Phi}^T\mathbf{K}^{-1/2}\tilde{\mathbf{W}}$:

$$\begin{aligned}&\Big[\mathrm{tr}(\mathbf{K}) - \mathrm{tr}(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T\mathbf{K}\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T)\Big] - \Big[\mathrm{tr}(\mathbf{K}) - \mathrm{tr}(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T\tilde{\mathbf{K}}\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T)\Big] \\ &= \mathrm{tr}\left(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T(\mathbf{K} - \tilde{\mathbf{K}})\right) = \mathrm{tr}\left(\tilde{\mathbf{W}}^T(\mathbf{K} - \tilde{\mathbf{K}})\tilde{\mathbf{W}}\right) = \sum_{i=1}^{k}\tilde{\mathbf{w}}_i^T(\mathbf{K} - \tilde{\mathbf{K}})\tilde{\mathbf{w}}_i \leq k\epsilon\frac{1}{k}\sum_{i=k+1}^{n}\sigma_i(\mathbf{K}). \qquad (23)\end{aligned}$$

The last step follows from Theorem 2 which guarantees that $(\mathbf{K} - \tilde{\mathbf{K}}) \preceq \epsilon\lambda\mathbf{I}$. Recall that we set $\lambda = \frac{1}{k}\sum_{i=k+1}^{n}\sigma_i(\mathbf{K})$ and each column $\tilde{\mathbf{w}}_i$ of $\tilde{\mathbf{W}}$ has unit norm.

We conclude that the cost obtained by $\mathbf{P_S}\mathbf{\Phi}^T\tilde{\mathbf{K}}^{-1/2}\tilde{\mathbf{W}}$ is bounded by:

$$\begin{aligned}\mathrm{tr}(\mathbf{K} - \mathbf{\Phi}\mathbf{P_S}\mathbf{\Phi}^T\tilde{\mathbf{K}}^{-1/2}\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T\tilde{\mathbf{K}}^{-1/2}\mathbf{\Phi}\mathbf{P_S}\mathbf{\Phi}^T) &\leq \mathrm{tr}(\mathbf{K}) - \mathrm{tr}(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T\mathbf{K}\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T) + \epsilon\sum_{i=k+1}^{n}\sigma_i(\mathbf{K}) \\ &\leq (1+2\epsilon)\sum_{i=k+1}^{n}\sigma_i(\mathbf{K}).\end{aligned}$$

This gives the result. Notice that $\mathbf{P_S}\mathbf{\Phi}^T\tilde{\mathbf{K}}^{-1/2}\tilde{\mathbf{W}} = \mathbf{\Phi}^T\mathbf{S}(\mathbf{S}^T\mathbf{K}^T\mathbf{S})^+\mathbf{S}^T\mathbf{\Phi}\mathbf{\Phi}^T\tilde{\mathbf{K}}^{-1/2}\tilde{\mathbf{W}}$ so, if we set

$$\mathbf{X} = (\mathbf{S}^T\mathbf{K}^T\mathbf{S})^+\mathbf{S}^T\tilde{\mathbf{K}}^{1/2}\tilde{\mathbf{W}}$$

our solution can be represented as $\mathbf{Z} = \mathbf{\Phi}^T\mathbf{S}\mathbf{X}$ as desired. □

A.4 Kernel canonical correlation analysis

We briefly discuss a final application to canonical correlation analysis (CCA) that follows from applying our additive error kernel embedding guarantee (Theorem 2) to recent work in [Wan16].

Consider n pairs of input points $(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n) \in (\mathcal{X}, \mathcal{Y})$ along with two positive semidefinite kernels, $K_x : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $K_y : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. Let $\mathcal{F}_x$ and $\mathcal{F}_y$ and $\phi_x : \mathcal{X} \to \mathcal{F}_x$ and $\phi_y : \mathcal{Y} \to \mathcal{F}_y$ be the Hilbert spaces and feature maps associated with these kernels. Let $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_y$ denote the kernelized $\mathcal{X}$ and $\mathcal{Y}$ inputs respectively and $\mathbf{K}_x$ and $\mathbf{K}_y$ denote the associated kernel matrices.

We consider standard regularized kernel CCA, following the presentation in [Wan16]. The goal is to compute coefficient vectors $\boldsymbol{\alpha}^x$ and $\boldsymbol{\alpha}^y$ such that $\mathbf{f}_x^* = \sum_{i=1}^{n}\boldsymbol{\alpha}_i^x\phi_x(\mathbf{x}_i)$ and $\mathbf{f}_y^* = \sum_{i=1}^{n}\boldsymbol{\alpha}_i^y\phi_y(\mathbf{y}_i)$ satisfy:

$$(\mathbf{f}_x^*, \mathbf{f}_y^*) = \underset{\mathbf{f}_x\in\mathcal{F}_x, \mathbf{f}_y\in\mathcal{F}_y}{\arg\max}\ \mathbf{f}_x^T\mathbf{\Phi}_x^T\mathbf{\Phi}_y\mathbf{f}_y^*$$

$$\text{subject to}$$

$$\mathbf{f}_x^T\mathbf{\Phi}_x^T\mathbf{\Phi}_x\mathbf{f}_x + \lambda_x\|\mathbf{f}_x\|_{\mathcal{F}_x}^2 = 1$$

$$\mathbf{f}_y^T\mathbf{\Phi}_y^T\mathbf{\Phi}_y\mathbf{f}_y + \lambda_y\|\mathbf{f}_y\|_{\mathcal{F}_y}^2 = 1$$

In [Wan16], the kernelized points are centered to their means. For simplicity we ignore centering, but note that [Wan16] shows how bounds for the uncentered problem carry over to the centered one.

It can be shown that $\boldsymbol{\alpha}^x = (\mathbf{K}_x + \lambda_x \mathbf{I})^{-1}\boldsymbol{\beta}^x$ and $\boldsymbol{\alpha}^y = (\mathbf{K}_y + \lambda_y \mathbf{I})^{-1}\boldsymbol{\beta}^y$ where $\boldsymbol{\beta}^x$ and $\boldsymbol{\beta}^y$ are the top left and right singular vectors respectively of

$$\mathbf{T} = (\mathbf{K}_x + \lambda_x \mathbf{I})^{-1}\mathbf{K}_x\mathbf{K}_y(\mathbf{K}_y + \lambda_y \mathbf{I})^{-1}.$$

The optimum value of the above program will be equal to $\sigma_1(\mathbf{T})$.

[Wan16] shows that if $\tilde{\mathbf{K}}_x$ and $\tilde{\mathbf{K}}_y$ are ϵ additive error kernel embeddings with parameters λ_x and λ_y, then if $\tilde{\boldsymbol{\alpha}}^x$ and $\tilde{\boldsymbol{\alpha}}^y$ are computed using these approximations, the achieved objective function value will be within ϵ of optimal (see their Lemma 1 and Theorem 1). So we have:

Theorem 13 (Kernel CCA Approximation Bound)**.** *Suppose $\tilde{\mathbf{K}}_x$ and $\tilde{\mathbf{K}}_y$ are computed by RLS-Nyström with parameters ϵ, δ, λ_x and λ_y. If we solve for $\tilde{\boldsymbol{\alpha}}^x$ and $\tilde{\boldsymbol{\alpha}}^y$, the approximate canonical correlation will be within an additive ϵ of the true canonical correlation $\sigma_1(\mathbf{T})$.*

By Corollary 7, Algorithm 2 can compute $\tilde{\mathbf{K}}_x$ and $\tilde{\mathbf{K}}_y$ with $O(ns_x + ns_y)$ kernel evaluations and $O(ns_x^2 + ns_y^2)$ computation time, with $s_x = O\left(\frac{d^x_{eff}}{\epsilon}\log\frac{d^x_{eff}}{\delta\epsilon}\right)$ for $d^x_{eff} = \operatorname{tr}(\mathbf{K}_x(\mathbf{K}_x + \lambda_x\mathbf{I})^{-1})$ and $s_y = O\left(\frac{d^y_{eff}}{\epsilon}\log\frac{d^y_{eff}}{\delta\epsilon}\right)$ for $d^y_{eff} = \operatorname{tr}(\mathbf{K}_y(\mathbf{K}_y + \lambda_y\mathbf{I})^{-1})$.

B Ridge leverage score sampling bounds

Lemma 14. *For any $\lambda > 0$ and $\epsilon, \delta \in (0,1)$, given ridge leverage score approximations $\tilde{l}_i^{\epsilon\lambda} \geq l_i^{\epsilon\lambda}$ for all i, let $p_i = \min\left\{\tilde{l}_i^{\epsilon\lambda} \cdot c\log(\sum l_i^{\epsilon\lambda}/\delta), 1\right\}$ for some sufficiently large constant c. Let $\mathbf{S} \in \mathbb{R}^{n\times s}$ be selected by sampling $\mathbf{x}_1, \ldots, \mathbf{x}_n$ each independently with probability p_i and rescaling selected columns by $1/\sqrt{p_i}$. With probability at least $1-\delta$, $s = O(\sum_i p_i)$ and*

$$\frac{1}{2}\mathbf{B}^T\mathbf{B} - \frac{1}{2}\epsilon\lambda\mathbf{I} \preceq \mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} \preceq \frac{3}{2}\mathbf{B}^T\mathbf{B} + \frac{1}{2}\epsilon\lambda\mathbf{I}, \tag{24}$$

Proof. Let $\mathbf{B} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ be the singular value decomposition of $\mathbf{B}$. By Definition 1:

$$\begin{aligned} l_i^{\epsilon\lambda} = \mathbf{b}_i^T\left(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I}\right)^{-1}\mathbf{b}_i &= \mathbf{b}_i^T\left(\mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^T + \epsilon\lambda\mathbf{V}\mathbf{V}^T\right)^{-1}\mathbf{b}_i \\ &= \mathbf{b}_i^T\left(\mathbf{V}\bar{\boldsymbol{\Sigma}}^2\mathbf{V}^T\right)^{-1}\mathbf{b}_i \\ &= \mathbf{b}_i^T\left(\mathbf{V}\bar{\boldsymbol{\Sigma}}^{-2}\mathbf{V}^T\right)\mathbf{b}_i, \end{aligned}$$

where $\bar{\boldsymbol{\Sigma}}_{i,i}^2 = \sigma_i^2(\mathbf{B}) + \epsilon\lambda$. For each $i \in 1, \ldots, n$ define the matrix valued random variable:

$$\mathbf{X}_i = \begin{cases} \left(\frac{1}{p_i} - 1\right)\bar{\boldsymbol{\Sigma}}^{-1}\mathbf{V}^T\mathbf{b}_i\mathbf{b}_i^T\mathbf{V}\bar{\boldsymbol{\Sigma}}^{-1} & \text{with probability } p_i \\ -\bar{\boldsymbol{\Sigma}}^{-1}\mathbf{V}^T\mathbf{b}_i\mathbf{b}_i^T\mathbf{V}\bar{\boldsymbol{\Sigma}}^{-1} & \text{with probability } (1-p_i) \end{cases}$$

Let $\mathbf{Y} = \sum_i \mathbf{X}_i$. We have $\mathbb{E}\,\mathbf{Y} = \mathbf{0}$. Furthermore, $\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} = \mathbf{V}\bar{\boldsymbol{\Sigma}}\mathbf{Y}\bar{\boldsymbol{\Sigma}}\mathbf{V}^T + \mathbf{B}^T\mathbf{B}$. Showing $\|\mathbf{Y}\|_2 \leq \frac{1}{2}$ gives $-\frac{1}{2}\mathbf{I} \preceq \mathbf{Y} \preceq \frac{1}{2}\mathbf{I}$, and since $\mathbf{V}\bar{\boldsymbol{\Sigma}}^2\mathbf{V}^T = \mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I}$ would give:

$$\frac{1}{2}\mathbf{B}^T\mathbf{B} - \frac{1}{2}\epsilon\lambda\mathbf{I} \preceq \mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} \preceq \frac{3}{2}\mathbf{B}^T\mathbf{B} + \frac{1}{2}\epsilon\lambda\mathbf{I},$$

giving the Lemma.

To prove that $\|\mathbf{Y}\|_2$ is small we use a stable rank matrix Bernstein inequality [Tro15]. If $p_i = 1$ (i.e. $\tilde{l}_i^{\epsilon\lambda} \cdot c\log(\sum l_i^{\epsilon\lambda}/\delta) \geq 1$) then $\mathbf{X}_i = \mathbf{0}$ so $\|\mathbf{X}_i\|_2 = 0$. Otherwise, we use the fact that

$$\frac{1}{\tilde{l}_i^{\epsilon\lambda}}\mathbf{b}_i\mathbf{b}_i^T \preceq \frac{1}{l_i^{\epsilon\lambda}}\mathbf{b}_i\mathbf{b}_i^T \preceq \mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I}. \tag{25}$$

This follows since we can write any $\mathbf{x} \in \mathbb{R}^n$ as $\mathbf{x} = (\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1/2}\mathbf{y}$ for some $\mathbf{y} \in \mathbb{R}^n$. We can then write:

$$\begin{aligned}\mathbf{x}^T\mathbf{b}_i\mathbf{b}_i^T\mathbf{x} &= \mathbf{y}^T(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1/2}\mathbf{b}_i\mathbf{b}_i^T(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1/2}\mathbf{y} \\ &\leq \|\mathbf{y}\|_2^2 \cdot \|(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1/2}\mathbf{b}_i\mathbf{b}_i^T(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1/2}\|_2.\end{aligned}$$

Since it is rank 1, we have

$$\begin{aligned}\|(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1/2}\mathbf{b}_i\mathbf{b}_i^T(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1/2}\|_2 &= \operatorname{tr}\left((\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1/2}\mathbf{b}_i\mathbf{b}_i^T(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1/2}\right) \\ &= \mathbf{b}_i^T(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{-1}\mathbf{b}_i = l_i^{\epsilon\lambda}\end{aligned} \tag{26}$$

where in the last step we use the cyclic property of the trace. Writing $\mathbf{y} = (\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})^{1/2}\mathbf{x}$ and plugging back into (26) we thus have:

$$\mathbf{x}^T\mathbf{b}_i\mathbf{b}_i^T\mathbf{x} \leq \|\mathbf{y}\|_2^2 \cdot l_i^{\epsilon\lambda} = \mathbf{x}^T(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I})\mathbf{x} \cdot l_i^{\epsilon\lambda}.$$

Rearranging gives (25). With this bound in place we have:

$$\frac{1}{\tilde{l}_i^{\epsilon\lambda}} \cdot \bar{\mathbf{\Sigma}}^{-1}\mathbf{V}^T\mathbf{b}_i\mathbf{b}_i^T\mathbf{V}\bar{\mathbf{\Sigma}}^{-1} \preceq \bar{\mathbf{\Sigma}}^{-1}\mathbf{V}^T\left(\mathbf{B}^T\mathbf{B} + \epsilon\lambda\mathbf{I}\right)\mathbf{V}\bar{\mathbf{\Sigma}}^{-1} = \mathbf{I}.$$

So we have

$$\mathbf{X}_i \preceq \frac{1}{p_i}\bar{\mathbf{\Sigma}}^{-1}\mathbf{V}^T\mathbf{b}_i\mathbf{b}_i^T\mathbf{V}\bar{\mathbf{\Sigma}}^{-1} \preceq \frac{\tilde{l}_i^{\epsilon\lambda}}{p_i}\mathbf{I} = \frac{1}{c\log\left(\sum l_i^{\epsilon\lambda}/\delta\right)}\mathbf{I}.$$

Next we bound the variance of $\mathbf{Y}$.

$$\begin{aligned}\mathbb{E}(\mathbf{Y}^2) = \sum \mathbb{E}(\mathbf{X}_i^2) &\preceq \sum\left[p_i\left(\frac{1}{p_i} - 1\right)^2 + (1 - p_i)\right] \cdot \bar{\mathbf{\Sigma}}^{-1}\mathbf{V}^T\mathbf{b}_i\mathbf{b}_i^T\mathbf{V}\bar{\mathbf{\Sigma}}^{-2}\mathbf{V}^T\mathbf{b}_i\mathbf{b}_i^T\mathbf{V}\bar{\mathbf{\Sigma}}^{-1} \\ &\preceq \sum\frac{1}{p_i} \cdot l_i^{\epsilon\lambda} \cdot \bar{\mathbf{\Sigma}}^{-1}\mathbf{V}^T\mathbf{b}_i\mathbf{b}_i^T\mathbf{V}\bar{\mathbf{\Sigma}}^{-1} \preceq \frac{1}{c\log\left(\sum l_i^{\epsilon\lambda}/\delta\right)}\bar{\mathbf{\Sigma}}^{-1}\mathbf{V}^T\mathbf{B}^T\mathbf{B}\mathbf{V}\bar{\mathbf{\Sigma}}^{-1} \\ &\preceq \frac{1}{c\log\left(\sum l_i^{\epsilon\lambda}/\delta\right)}\mathbf{\Sigma}^2\bar{\mathbf{\Sigma}}^{-2} \preceq \frac{1}{c\log\left(\sum l_i^{\epsilon\lambda}/\delta\right)}\mathbf{D}.\end{aligned} \tag{27}$$

where $\mathbf{D}_{1,1} = 1$ and $\mathbf{D}_{i,i} = (\mathbf{\Sigma}^2\bar{\mathbf{\Sigma}}^{-2})_{i,i} = \frac{\sigma_i^2(\mathbf{B})}{\sigma_i^2(\mathbf{B})+\epsilon\lambda}$ for all $i \geq 2$. By the stable rank matrix Bernstein inequality given in Theorem 7.3.1 of [Tro15]

$$\mathbb{P}\left[\|\mathbf{Y}\| \geq \frac{1}{2}\right] \leq \frac{4\operatorname{tr}(\mathbf{D})}{\|\mathbf{D}\|_2}e^{\left(\frac{1}{4c\log(\sum l_i^{\epsilon\lambda}/\delta)}(\|\mathbf{D}\|_2+1/6)\right)^{-1/8}}. \tag{28}$$

Clearly $\|\mathbf{D}\|_2 = 1$. Additionally,

$$\text{tr}(\mathbf{D}) \leq 1 + \text{tr}(\mathbf{\Sigma}^2\bar{\mathbf{\Sigma}}^{-2}) = 1 + \text{tr}\left(\mathbf{K}(\mathbf{K}+\epsilon\lambda\mathbf{I})^{-1}\right) = 1 + \sum_i l_i^{\epsilon\lambda}.$$

Plugging into (28), we see that

$$\mathbb{P}\left[\|\mathbf{Y}\| \geq \frac{1}{2}\right] \leq 4\left(1 + \sum_i l_i^{\epsilon\lambda}\right) \cdot e^{-\Theta\left(c\log(\sum l_i^{\epsilon\lambda}/\delta)\right)} \leq \delta/2,$$

if we choose the constant c large enough. Note that here we make the extremely mild assumption that $\sum_i l_i^{\epsilon\lambda} \geq 1$. If not, we can simply use a smaller λ that allows this condition to be true, and will have $s = O(1)$. So we have established (29).

All that remains to show is that, the sample size s is bounded with high probability. If $p_i = 1$, we always sample i so there is no variance in s. Let $S \subseteq [1, ..., n]$ be the set of indices with $p_i < 1$. The expected number of points sampled from S is $\sum_{i\in S} p_i = c\log(\sum l_i^{\epsilon\lambda}/\delta) \cdot \sum_{i\in S} \tilde{l}_i^{\epsilon\lambda}$. Assume without loss of generality that $\sum_{i\in S} \tilde{l}_i^{\epsilon\lambda} \geq 1$ – otherwise can just increase our leverage score estimates and increase the expected sample size by at most 1. Then, by a standard Chernoff bound, with probability at least $1 - \delta/2$, $O\left(\log(\sum l_i^{\epsilon\lambda}/\delta) \cdot \sum_i \tilde{l}_i^{\epsilon\lambda}\right)$ points are sampled from S. Union bounding over failure probabilities gives the lemma. ☐

The above Lemma yields an easy Corollary which is all we need for $\mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B}$:

Corollary 15. *For any $\lambda > 0$ and $\epsilon, \delta \in (0, 1)$, given ridge leverage score approximations $\tilde{l}_i^{\epsilon\lambda} \geq l_i^{\epsilon\lambda}$ for all i, let $p_i = \min\left\{\tilde{l}_i^{\epsilon\lambda} \cdot c\log(\sum l_i^{\epsilon\lambda}/\delta), 1\right\}$ for some sufficiently large constant c. Let $\mathbf{S} \in \mathbb{R}^{n\times s}$ be selected by sampling $x_1, ..., x_n$ each independently with probability p_i. With probability at least $1 - \delta$, $s = O(\sum_i p_i)$ and the exists some scaling factor C such that*

$$\mathbf{B}^T\mathbf{B} \preceq C \cdot \mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I} \tag{29}$$

Proof. By Lemma 14, if we set $C' = \frac{1}{\min_i p_i}$ we have:

$$\frac{1}{2}\mathbf{B}^T\mathbf{B} - \frac{1}{2}\epsilon\lambda\mathbf{I} \preceq C' \cdot \mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B}$$
$$\mathbf{B}^T\mathbf{B} \preceq 2C' \cdot \mathbf{B}^T\mathbf{S}\mathbf{S}^T\mathbf{B} + \epsilon\lambda\mathbf{I}$$

which gives the corollary by setting $C = 2C'$. ☐

C Additional proofs

Lemma 16. *For any $\mathbf{K}, \mathbf{K}' \in \mathbb{R}^{n\times n}$, $\mathbf{K}' \preceq \mathbf{K}$,*

$$\sum_{i=1}^n l_i^{\lambda}(\mathbf{K}') \leq \sum_{i=1}^n l_i^{\lambda}(\mathbf{K}).$$

Proof. As discussed in [AM15], $l_i^\lambda = \left(\mathbf{K}(\mathbf{K}+\lambda\mathbf{I})^{-1}\right)_{i,i}$ so

$$\sum_{i=1}^{n} l_i^\lambda(\mathbf{K}) = \operatorname{tr}\left(\mathbf{K}(\mathbf{K}+\lambda\mathbf{I})^{-1}\right) = \sum_{i=1}^{n} \frac{\sigma_i(\mathbf{K})}{\sigma_i(\mathbf{K})+\lambda}.$$

$\mathbf{K}' \preceq \mathbf{K}$ implies that $\sigma_i(\mathbf{K}') \leq \sigma_i(\mathbf{K})$ for all i. So:

$$\sum_{i=1}^{n} l_i^\lambda(\mathbf{K}') = \sum_{i=1}^{n} \frac{\sigma_i(\mathbf{K}')}{\sigma_i(\mathbf{K}')+\lambda} \leq \sum_{i=1}^{n} \frac{\sigma_i(\mathbf{K})}{\sigma_i(\mathbf{K})+\lambda} = \sum_{i=1}^{n} l_i^\lambda(\mathbf{K})$$

giving the lemma. □

Lemma 16 generalizes to the case when $\mathbf{K}'$ and $\mathbf{K}$ have different sizes and each of $\mathbf{K}'$'s singular values is bounded above by the corresponding singular value of $\mathbf{K}$. We give a proof for a specific case that we will require:

Lemma 17. *For any* $\mathbf{K} \in \mathbb{R}^{n\times n}$ *and* $\mathbf{S} \in \mathbb{R}^{n\times s}$ *with* $\|\mathbf{S}\|_2 \leq 1$. *If* $\mathbf{K}' = \mathbf{S}^T\mathbf{K}\mathbf{S}$ *then,*

$$\sum_{i=1}^{n} l_i^\lambda(\mathbf{K}') \leq \sum_{i=1}^{n} l_i^\lambda(\mathbf{K}).$$

Proof. Define $\bar{\mathbf{S}} \in \mathbb{R}^{n\times n}$ as $\bar{\mathbf{S}} = [\mathbf{S}, \mathbf{0}_{n\times n-s}]$. Clearly $\|\bar{\mathbf{S}}\|_2 \leq 1$ and so $\bar{\mathbf{S}}^T\mathbf{K}\bar{\mathbf{S}} \preceq \mathbf{K}$. Additionally, for $i \in [1,...,s]$, $l_i^\lambda(\bar{\mathbf{S}}^T\mathbf{K}\bar{\mathbf{S}}) = l_i^\lambda(\mathbf{K}')$ and for $i \in [s+1,...,n]$, $l_i^\lambda(\bar{\mathbf{S}}^T\mathbf{K}\bar{\mathbf{S}}) = 0$. So $\sum_{i=1}^n l_i^\lambda(\mathbf{K}') = \sum_{i=1}^n l_i^\lambda(\bar{\mathbf{S}}^T\mathbf{K}\bar{\mathbf{S}})$ and the lemma follows by applying Lemma 16. □