Recursive Sampling for the Nyström Method

Cameron Musco
MIT EECS
cnmusco@mit.edu

Christopher Musco
MIT EECS
cpmusco@mit.edu

Abstract

We give the first algorithm for kernel Nyström approximation that runs in linear time in the number of training points and is provably accurate for all kernel matrices, without dependence on regularity or incoherence conditions. The algorithm projects the kernel onto a set of \( s \) landmark points sampled by their ridge leverage scores, requiring just \( O(ns) \) kernel evaluations and \( O(ns^2) \) additional runtime. While leverage score sampling has long been known to give strong theoretical guarantees for Nyström approximation, by employing a fast recursive sampling scheme, our algorithm is the first to make the approach scalable. Empirically we show that it finds more accurate kernel approximations in less time than popular techniques such as classic Nyström approximation and the random Fourier features method.

1 Introduction

The kernel method is a powerful for applying linear learning algorithms (SVMs, linear regression, etc.) to nonlinear problems. The key idea is to map data to a higher dimensional kernel feature space, where linear relationships correspond to nonlinear relationships in the original data.

Typically this mapping is implicit. A kernel function is used to compute inner products in the high-dimensional kernel space, without ever actually mapping original data points to the space. Given \( n \) data points \( x_1, \ldots, x_n \), the \( n \times n \) kernel matrix \( K \) is formed where \( K_{i,j} \) contains the high-dimensional inner product between \( x_i \) and \( x_j \), as computed by the kernel function. All computations required by a linear learning method are performed using the inner product information in \( K \).

Unfortunately, the transition from linear to nonlinear comes at a high cost. Just generating the entries of \( K \) requires \( \Theta(n^2) \) time, which is prohibitive for large datasets.

1.1 Kernel approximation

A large body of work seeks to accelerate kernel methods by finding a compressed, often low-rank, approximation \( \tilde{K} \) to the true kernel matrix \( K \). Techniques include random sampling and embedding [AMS01, BBV06, ANW14], random Fourier feature methods for shift invariant kernels [RR07, RR09, LSS13], and incomplete Cholesky factorization [FS02, BJ02].

One of the most popular techniques is the Nyström method, which constructs \( \tilde{K} \) using a subset of “landmark” data points [WS01]. Once \( s \) data points are selected, \( \tilde{K} \) (in factored form) takes just \( O(ns) \) kernel evaluations and \( O(s^3) \) additional time to compute, requires \( O(ns) \) space to store, and can be manipulated quickly in downstream applications. E.g., inverting \( \tilde{K} \) takes \( O(ns^2) \) time.

The Nyström method performs well in practice [YLM+12, GM13, TRVR16], is widely implemented [HFH+09, PVG+11, IBM14], and is used in a number of applications under different names such as “landmark isomap” [DST03] and “landmark MDS” [Pla05]. In the classic variant, landmark points are selected uniformly at random. However, significant research seeks to improve performance via data-
dependent sampling that selects landmarks which more closely approximate the full kernel matrix than uniformly sampled landmarks [SS00, DM05, ZTK08, BW09, KMT12, WZ13, GM13, LJS16]. Theoretical work has converged on leverage score based approaches, as they give the strongest provable guarantees for both kernel approximation [DMM08, GM13] and statistical performance in downstream applications [AM15, RCR15, Wan16]. Leverage scores capture how important an individual data point is in composing the span of the kernel matrix.

Unfortunately, these scores are prohibitively expensive to compute. All known approximation schemes require Ω(n²) time or only run quickly under strong conditions on K – e.g. good conditioning or data “incoherence” [DMIMW12, GM13, AM15, CLV16]. Hence, leverage score-based approaches remain largely in the domain of theory, with limited practical impact [KMT12, LBKL15, YPW15].

1.2 Our contributions

In this work, we close the gap between strong approximation bounds and efficiency: we present a new Nyström algorithm based on recursive leverage score sampling which achieves the “best of both worlds”: it produces kernel approximations provably matching the high accuracy of leverage score methods while only requiring O(ns) kernel evaluations and O(ns²) runtime for s landmark points.

Theoretically, this runtime is surprising. In the typical case when s ≪ n, the algorithm evaluates just a small subset of K, ignoring most of the kernel space inner products. Yet its performance guarantees hold for general kernels, requiring no assumptions on coherence or regularity.

Empirically, the runtime’s linear dependence on n means that our method is the first leverage score algorithm that can compete with the most commonly implemented techniques, including the classic uniform sampling Nyström method and random Fourier features sampling [RR07]. Since our algorithm obtains higher quality samples, we show experimentally that it outperforms these methods on benchmark datasets – it can obtain as accurate a kernel approximation in significantly less time. Our approximations also have lower rank, so they can be stored in less space and processed more quickly in downstream learning tasks.

1.3 Paper outline

Our recursive sampling algorithm is built on top of a Nyström scheme of Alaoui and Mahoney that samples landmark points based on their ridge leverage scores [AM15]. After reviewing preliminaries in Section 2, in Section 3 we analyze this scheme, which we refer to as RLS-Nyström. To simplify prior work, which studies the statistical performance of RLS-Nyström for specific kernel learning tasks [AM15, RCR15, Wan16], we prove a strong, application independent approximation guarantee: for any λ, if ̂K is constructed with s = Θ(λeff log d') samples1, where d' = tr(K(K + 1I)⁻¹) is the so-called “λ-effective dimensionality” of K, then with high probability, ∥K − ̂K∥₂ ≤ λ.

In Appendix E, we show that this guarantee implies bounds on the statistical performance of RLS-Nyström for kernel ridge regression and canonical correlation analysis. We also use it to prove new results on the performance of RLS-Nyström for kernel rank-k PCA and k-means clustering – in both cases just O(k log k) samples are required to obtain a solution with good accuracy.

After affirming the favorable theoretical properties of RLS-Nyström, in Section 4 we show that its runtime can be significantly improved using a recursive sampling approach. Intuitively our algorithm is simple. We show how to approximate the kernel ridge leverage scores using a uniform sample of 1/2 of our input points. While the subsampled kernel matrix still has a prohibitive n²/4 entries, we can recursively approximate it, using our same sampling algorithm. If our final Nyström approximation will use s landmarks, the recursive approximation only needs rank O(s), which lets us estimate the ridge leverage scores of the original kernel matrix in just O(ns²) time. Since n is cut in half at each level of recursion, our total runtime is\[ O\left(ns^2 + \frac{ns^2}{2} + \frac{ns^2}{4} + ...\right) = O(ns^2), \]significantly improving upon the method of [AM15], which takes Θ(n³) time in the worst case.

Our approach builds on recent work on iterative sampling methods for approximate linear algebra [CLMT15, CMM17]. While the analysis in the kernel setting is technical, our final algorithm is

---

1This is within a log factor of the best possible for any low-rank approximation with error λ.
simple and easy to implement. We present and test a parameter-free variation of Recursive RLS-Nyström in Section 5, confirming superior performance compared to existing methods.

2 Preliminaries

Consider an input space $\mathcal{X}$ and a positive semidefinite kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Let $\mathcal{F}$ be an associated reproducing kernel Hilbert space and $\phi : \mathcal{X} \to \mathcal{F}$ be a (typically nonlinear) 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 $x_1, \ldots, x_n \in \mathcal{X}$, define the kernel matrix $K \in \mathbb{R}^{n \times n}$ by $K_{i,j} = K(x_i, x_j)$.

It is often natural to consider the kernelized data matrix that generates $K$. Informally, let $\Phi \in \mathbb{R}^{n \times d'}$ be the matrix containing $\phi(x_1), \ldots, \phi(x_n)$ as its rows (note that $d'$ may be infinite). $K = \Phi \Phi^T$.

While we use $\Phi$ for intuition, in our formal proofs we replace it with any matrix $B \in \mathbb{R}^{n \times n}$ satisfying $BB^T = K$ (e.g. a Cholesky factor). Such a $B$ is guaranteed to exist since $K$ is positive semidefinite.

We repeatedly use the singular value decomposition, which allows us to write any rank $r$ matrix $M \in \mathbb{R}^{n \times d}$ as $M = U \Sigma V^T$, where $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{d \times r}$ have orthogonal columns (the left and right singular vectors of $M$), and $\Sigma \in \mathbb{R}^{r \times r}$ is a positive diagonal matrix containing the singular values: $\sigma_1(M) \geq \sigma_2(M) \geq \ldots \geq \sigma_r(M) > 0$. $M$’s pseudoinverse is given by $M^+ = V \Sigma^{-1} U^T$.

2.1 Nyström approximation

The Nyström method selects a subset of “landmark” points and uses them to construct a low-rank approximation to $K$. Given a matrix $S \in \mathbb{R}^{n \times s}$ that has a single entry in each column equal to 1 so that $K_S$ is a subset of $s$ columns from $K$, the associated Nyström approximation is:

$$\tilde{K} = K_S (S^T K_S)^+ S^T K.$$  \hfill (1)

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

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

$$\tilde{K} = \tilde{\Phi} \tilde{\Phi}^T = \Phi P_S^2 \Phi^T = \Phi (\Phi^T S (S^T \Phi \Phi^T S)^+ S^T \Phi) \Phi^T = K_S (S^T K_S)^+ S^T K.$$  \hfill (2)

This recovers the standard Nyström approximation (1).

3 The RLS-Nyström method

We now introduce the RLS-Nyström method, which uses ridge leverage score sampling to select landmark data points, and discuss its strong approximation guarantees for any kernel matrix $K$.

3.1 Ridge leverage scores

In classical Nyström approximation (1), $S$ is formed by sampling data points uniformly at random. Uniform sampling can work in practice, but it only gives theoretical guarantees under strong regularity or incoherence assumptions on $K$ [Git11]. It will fail for many natural kernel matrices where the relative “importance” of points is not uniform across the dataset.

For example, imagine a dataset where points fall into several clusters, but one of the clusters is much larger than the rest. Uniform sampling will tend to oversample landmarks from the large cluster while undersampling or possibly missing smaller but still important clusters. Approximation of $K$ and learning performance (e.g. classification accuracy) will decline as a result.
We denote 

As is standard in leverage score methods, we don’t directly select landmarks to be the points with the highest scores. Instead, we sample each point with probability proportional to \( l_i^\lambda \). Accordingly, the number of landmarks selected, which controls \( \tilde{K} \)’s rank, is a random variable with expectation equal to the sum of the \( \lambda \)-ridge leverage scores. To ensure compact kernel approximations, we want this sum to be small. Immediately from Definition 1, we have:

**Fact 2.** \[ \sum_{i=1}^n l_i^\lambda(\mathbf{K}) = \text{tr}(\mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1}). \]

We denote \( d^\lambda_{\text{eff}} \equiv \text{tr}(\mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1}) \). \( d^\lambda_{\text{eff}} \) is a natural quantity, referred to as the “effective dimension” or “degrees of freedom” for a ridge regression problem on \( \mathbf{K} \) with regularization \( \lambda \) [HTF02, Zha06]. \( d^\lambda_{\text{eff}} \) increases monotonically as \( \lambda \) decreases. For any fixed \( \lambda \) it is essentially the smallest possible rank achievable for \( \tilde{K} \) satisfying the approximation guarantee given by RLS-Nyström: \( \| \mathbf{K} - \tilde{K} \|_2 < \lambda \).
3.3 The basic sampling algorithm

We can now introduce the RLS-Nyström method as Algorithm 1. We allow sampling each point by any probability greater than \( l_i^\lambda \), which is useful later when we compute the scores approximately. Oversampling landmarks can only improve \( \tilde{K} \)'s accuracy. It could cause us to take more samples, but we will always ensure that the sum of our approximate ridge leverage scores is not too large.

\textbf{Algorithm 1} RLS-NYSTRÖM SAMPLING

\begin{algorithm}
\textbf{input}: \( x_1, \ldots, x_n \in \mathcal{X} \), kernel matrix \( K \), ridge parameter \( \lambda > 0 \), failure probability \( \delta \in (0, 1/8) \)
\begin{enumerate}
\item Compute an over-approximation, \( \tilde{l}_i^\lambda > l_i^\lambda \) for the \( \lambda \)-ridge leverage score of each \( x_1, \ldots, x_n \)
\item Set \( p_i := \min \left\{ 1, \tilde{l}_i^\lambda \cdot 16 \log (\sum \tilde{l}_i^\lambda / \delta) \right\} \).
\item Construct \( S \in \mathbb{R}^{n \times s} \) by sampling \( x_1, \ldots, x_n \) each independently with probability \( p_i \). In other words, for each \( i \) add a column to \( S \) with a 1 in position \( i \) with probability \( p_i \).
\item \textbf{return} the Nyström factors \( KS \in \mathbb{R}^{n \times s} \) and \( (S^T KS)^+ \in \mathbb{R}^{s \times s} \).
\end{enumerate}
\end{algorithm}

3.4 Accuracy bounds

We show that RLS-Nyström produces \( \tilde{K} \) which spectrally approximates \( K \) up to a small additive error. This is the strongest type of approximation offered by any known Nyström method [GM13]. It guarantees provable accuracy when \( \tilde{K} \) is used in place of \( K \) in many learning applications [CMT10].

\textbf{Theorem 3} (Spectral error approximation). For any \( \lambda > 0 \) and \( \delta \in (0, 1/8) \), Algorithm 1 returns \( S \in \mathbb{R}^{n \times s} \) such that with probability \( 1 - \delta, s \leq 2 \sum p_i \) and \( \tilde{K} = KS(S^T KS)^+ S^T K \) satisfies:
\[
\tilde{K} \preceq K \preceq \tilde{K} + \lambda I. \tag{5}
\]

When ridge scores are computed exactly, \( \sum p_i = O \left( d_{\text{eff}}^\lambda \log (d_{\text{eff}}^\lambda / \delta) \right) \).

\( \preceq \) denotes the Loewner ordering: \( M \preceq N \) means that \( N - M \) is positive semidefinite. Note that (5) immediately implies the well studied (see e.g. [GM13]) spectral norm guarantee, \( \| K - \tilde{K} \|_2 \leq \lambda \).

Intuitively, Theorem 3 guarantees that \( \tilde{K} \) well approximates the top of \( K \)'s spectrum (i.e. any eigenvalues > \( \lambda \)) while losing information about smaller, less important eigenvalues. Due to space constraints, we defer the proof to Appendix A. It relies on the view of Nyström approximation as a low-rank projection of the kernelized data (see Section 2.1) and we use an intrinsic dimension matrix Bernstein bound to show accuracy of the sampled approximation.

Often the regularization parameter \( \lambda \) is specified for a learning task, and for near optimal performance on this task, we set the approximation factor in Theorem 3 to \( \epsilon \lambda \). In this case we have:

\textbf{Corollary 4} (Tighter spectral error approximation). For any \( \lambda > 0 \) and \( \delta \in (0, 1/8) \), Algorithm 1 run with ridge parameter \( \epsilon \lambda \) returns \( S \in \mathbb{R}^{n \times s} \) such that with probability \( 1 - \delta, s = O \left( \frac{d_{\text{eff}}^\lambda}{\epsilon} \log (\frac{d_{\text{eff}}^\lambda}{\delta \epsilon}) \right) \) and \( \tilde{K} = KS(S^T KS)^+ S^T K \) satisfies \( \tilde{K} \preceq K \preceq \tilde{K} + \epsilon \lambda I. \)

\textbf{Proof}. This follows from Theorem 3 by noting \( d_{\text{eff}}^\lambda \leq \frac{d_{\text{eff}}^\lambda}{\epsilon} \) since \( (K + \epsilon \lambda I)^{-1} \succeq \frac{1}{\epsilon} (K + \lambda I)^{-1} \).

Corollary 4 suffices to prove that \( \tilde{K} \) can be used in place of \( K \) without sacrificing performance on kernel ridge regression and canonical correlation tasks [AM15, Wan16]. We also use it to prove a projection-cost preservation guarantee (Theorem 12, Appendix B), which gives approximation bounds for kernel PCA and \( k \)-means clustering. Projection-cost preservation has proven a powerful concept in the matrix sketching literature [FSS13, CEM+15, CMM17, BWZ16, CW17] and we hope that extending the guarantee to kernels leads to applications beyond those considered in this work.

Our results on downstream learning bounds that can be derived from Theorem 3 are summarized in Table 1. Details can be found in Appendices B and E.
4 Recursive sampling for efficient RLS-Nyström

Having established strong approximation guarantees for RLS-Nyström, it remains to provide an efficient implementation. Specifically, Step 1 of Algorithm 1 naively requires $\Theta(n^3)$ time. We show that significant acceleration is possible using a recursive sampling approach.

4.1 Ridge leverage score approximation via uniform sampling

The key is to estimate the leverage scores by computing (3) approximately, using a uniform sample of the data points. To ensure accuracy, the sample must be large – a constant fraction of the points. Our fast runtimes are achieved by recursively approximating this large sample. In Appendix F we prove:

Lemma 5. For any $B \in \mathbb{R}^{n \times n}$ with $BB^T = K$ and $S \in \mathbb{R}^{n \times s}$ chosen by sampling each data point independently with probability $1/2$, let $l_i^\lambda = b_i^T(B'B + \lambda I)^{-1} b_i$ and $p_i = \min\{1, 16l_i^\lambda \log(\sum_i l_i^\lambda / \delta)\}$ for any $\delta \in (0, 1/8)$. Then with probability at least $1 - \delta$:

1) $l_i^\lambda \geq \tilde{l}_i^\lambda$ for all $i$
2) $\sum_i p_i \leq 64 \sum_i l_i^\lambda \log(\sum_i l_i^\lambda / \delta)$.

The first condition ensures that the approximate scores $\tilde{l}_i^\lambda$ suffice for use in Algorithm 1. The second ensures that the Nyström approximation obtained will not have too many sampled landmarks.

Naively computing $\tilde{l}_i^\lambda$ in Lemma 5 involves explicitly forming $B$, requiring $\Omega(n^2)$ time (e.g. $\Theta(n^3)$ via Cholesky decomposition). Fortunately, the following formula (proof in Appx. F) avoids this cost:

Lemma 6. For any sampling matrix $S \in \mathbb{R}^{n \times s}$, and any $\lambda > 0$:

$$l_i^\lambda \equiv b_i^T(B'B + \lambda I)^{-1} b_i = \frac{1}{\lambda} \left( K - KS TKS + \lambda I \right)^{-1} S^T K_{i,i}$$

It follows that we can compute all $\tilde{l}_i^\lambda$ for all $i$ in $O(ns^2)$ time using just $O(ns)$ kernel evaluations, to compute $KS$ and the diagonal of $K$.

4.2 Recursive RLS-Nyström

We apply Lemmas 5 and 6 to give an efficient recursive implementation of RLS-Nyström, Algorithm 2. We show that the output of this algorithm, $\tilde{K}$, is sampled according to approximate ridge leverage scores for $K$ and thus satisfies the approximation guarantee of Theorem 3.

Theorem 7 (Main Result). Let $S \in \mathbb{R}^{n \times s}$ be computed by Algorithm 2. With probability $1 - 3\delta$, $s \leq 384 \cdot d_{\text{eff}} \log(d_{\text{eff}} / \delta)$, $S$ is sampled by overestimates of the $\lambda$-ridge leverage scores of $K$, and thus by Theorem 3, the Nyström approximation $\tilde{K} = KS (S^T KS)^+ S^T K$ satisfies:

$$\tilde{K} \preceq K \preceq \tilde{K} + \lambda I.$$
Algorithm 2 RecursiveRLS-Nyström.

**Input:** $x_1, \ldots, x_m \in \mathcal{X}$, kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, ridge $\lambda > 0$, failure prob. $\delta \in (0, 1/32)$

**Output:** weighted sampling matrix $S \in \mathbb{R}^{m \times s}$

1. if $m \leq 192 \log(1/\delta)$ then

2. return $S := I_{m \times m}$

3. end if

4. Let $\hat{S}$ be a random subset of $\{1, \ldots, m\}$, with each $i$ included independently with probability $\frac{1}{2}$.

5. Let $\hat{X} = \{x_{i_1}, x_{i_2}, \ldots, x_{i_{\hat{S}}}\}$ for $i_j \in \hat{S}$ be the data sample corresponding to $\hat{S}$.

6. Let $\hat{S} = [e_{i_1}, e_{i_2}, \ldots, e_{i_{\hat{S}}}]$ be the sampling matrix corresponding to $\hat{S}$.

7. Set $\hat{l}_i := \frac{1}{2} \left( K - K\hat{S}(\hat{S}^T K\hat{S} + \lambda I)^{-1} \hat{S}^T K \right)_{i,i}$ for each $i \in \{1, \ldots, m\}$.

8. Set $p_i := \min(1, \frac{1}{\sqrt{\hat{l}_i}} \cdot 16 \log(\sum \hat{l}_i^2/\delta))$ for each $i \in \{1, \ldots, m\}$.

9. Initially set weighted sampling matrix $S$ to be empty. For each $i \in \{1, \ldots, m\}$, with probability $p_i$, append the column $\frac{1}{\sqrt{p_i}} e_i$ onto $S$.

10. return $S$.

Note that in Algorithm 2 the columns of $S$ are weighted by $1/\sqrt{p_i}$. The Nyström approximation $\tilde{K} = K\hat{S}(\hat{S}^T K\hat{S}) + \hat{S}^T K$ is not affected by column weights (see derivation in Section 2.1). However, the weighting is necessary when the output is used in recursive calls (i.e. when $\hat{S}$ is used in Step 6).

We prove Theorem 7 via the following intermediate result:

**Theorem 8.** For any inputs $x_1, \ldots, x_m$, $K$, $\lambda > 0$ and $\delta \in (0, 1/32)$, let $K$ be the kernel matrix for $x_1, \ldots, x_m$ and kernel function $K$ and let $d_{\text{eff}}(K)$ be the effective dimension of $K$ with parameter $\lambda$. With probability $(1 - 3\delta)$, RecursiveRLS-Nyström outputs $S$ with $s$ columns that satisfies:

$$\frac{1}{2} (B^T B + \lambda I) \preceq (B^T S S^T B + \lambda I) \preceq \frac{3}{2} (B^T B + \lambda I) \quad \text{for any } B \text{ with } B B^T = K.$$  \hfill (6)

Additionally, $s \leq s_{\text{max}}(d_{\text{eff}}(K), \delta)$ where $s_{\text{max}}(w, z) \overset{\text{def}}{=} 384 \cdot (w + 1) \log((w + 1)/z)$. The algorithm uses $c_1 s_{\text{max}}(d_{\text{eff}}(K), \delta)$ kernel evaluations and $\leq c_2 s_{\text{max}}(d_{\text{eff}}(K), \delta)^2$ additional computation time where $c_1$ and $c_2$ are fixed universal constants.

Theorem 8 is proved via an inductive argument, given in Appendix C. Roughly, consider in Step 6 of Algorithm 2, setting $\hat{S} := \hat{S}$ instead of $\tilde{S} \cdot \hat{S}$. By Lemma 5 and the formula in Lemma 6, the leverage score approximations $\hat{l}_i$ computed in Step 7 would be good approximations to the true leverage scores, and $\hat{S}$ would satisfy Theorem 8 by a standard matrix Bernstein bound (see Lemma 9). However, if we set $\hat{S} := \tilde{S}$, it will have $n/2$ columns in expectation, and the computation in Step 7 will be expensive – requiring roughly $O(n^3)$ time. By recursively calling Algorithm 8 and applying Theorem 8 inductively, we obtain $\tilde{S}$ satisfying with high probability:

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

This guarantee ensures that when we use $\hat{S} := \tilde{S} \cdot \hat{S}$ in place of $\hat{S}$ in Step 7, the leverage score estimates are changed only by a constant factor. Thus, sampling by these estimates, still gives us the desired guarantee (6). Further, $\hat{S}$ and therefore $\hat{S}$ has just $O(s_{\text{max}}(d_{\text{eff}}(K), \delta))$ columns, so Step 7 can be performed very efficiently, within the stated runtime bounds.

With Theorem 8 we can easily prove our main result, Theorem 7.

**Proof of Theorem 7.** In our proof of Theorem 3 in Appendix A.1, we show that if

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


We split categorical features into binary indicatory features and mean center and normalize features to

\[ \mathbf{K} \preceq \mathbf{K} \preceq \mathbf{K} + \lambda \mathbf{I}. \]

The runtime bounds also follow nearly directly from Theorem 8. In particular, we have established that \( O(n s_{\text{max}}(d_{\text{eff}}(\mathbf{K}, \delta))) \) kernel evaluations and \( O(n s_{\text{max}}(d_{\text{eff}}(\mathbf{K}, \delta)^2)) \) additional runtime are required by \textsc{RecursiveRLSNyström}. We only needed the upper bound to prove Theorem 8, but along the way actually show that in a successful run of \textsc{RecursiveRLSNyström}, \( \mathbf{S} \) has \( \Theta(d_{\text{eff}}(\mathbf{K}) \log(d_{\text{eff}}(\mathbf{K})/\delta)) \) columns. Additionally, we may assume that \( d_{\text{eff}}(\mathbf{K}) \geq 1/2 \). If it is not, then \( s_{\text{eff}} \) is not hard to check (see proof of Lemma 20) that \( \lambda \) must be \( \geq ||\mathbf{K}|| \). If this is the case, the guarantee of Theorem 7 is vacuous: any Nyström approximation \( \mathbf{K} \) satisfies \( \mathbf{K} \preceq \mathbf{K} \preceq \mathbf{K} + \lambda \mathbf{I} \). With \( d_{\text{eff}}(\mathbf{K}) \geq 1/2 \), \( s_{\text{eff}}(d_{\text{eff}}(\mathbf{K})/\delta) \) and thus \( s \) are \( \Theta(s_{\text{max}}(d_{\text{eff}}(\mathbf{K}), \delta)) \) so we conclude that Theorem 7 uses \( O(ns) \) kernel evaluations and \( O(ns^2) \) additional runtime.

\[ 5 \quad \text{Empirical evaluation} \]

We conclude with an empirical evaluation of our recursive RLS-Nyström method. We use a variant of Algorithm 2 where, instead of choosing a regularization parameter \( \lambda \), the user sets a sample size \( s \) and \( \lambda \) is automatically determined such that \( s = \Theta(d_{\text{eff}} \cdot \log(d_{\text{eff}}/\delta)) \). This variant is practically appealing as it essentially yields the best possible approximation to \( \mathbf{K} \) for a fixed sample budget. Pseudocode and proofs of correctness are included in Appendix D.

\[ 5.1 \quad \text{Performance of Recursive RLS-Nyström for kernel approximation} \]

We evaluate RLS-Nyström on the \textsc{YearPredictionMSD}, \textsc{Covertype}, \textsc{Cod-RNA}, and \textsc{Adult} datasets downloaded from the UCI ML Repository \cite{Lichman2013} and \cite{Huda2006}. These datasets contain 515345, 581012, 331152, and 48842 data points respectively. We compare against the classic Nyström method with uniform sampling \cite{Williams2001} and the random Fourier features method \cite{Rahimi2008}. Due to the large size of the datasets, prior leverage score based Nyström approaches \cite{Deng2012, Gammerman2013, Albermann2015}, which require at least \( \Omega(n^2) \) time are infeasible, and thus not included in our tests.

We split categorical features into binary indicatory features and mean center and normalize features to have variance 1. We use a Gaussian kernel for all tests, with the width parameter \( \sigma \) selected via cross validation on regression and classification tasks. \( ||\mathbf{K} - \tilde{\mathbf{K}}||_2 \) is used to measure approximation error. Since this quantity is prohibitively expensive to compute directly (it requires building the full kernel matrix \( \mathbf{K} \)), the error is estimated using a random subset of 20,000 data points and repeated trials.

![Image](image_url)

\[ (a) \text{Adult} \quad (b) \text{Covertype} \quad (c) \text{Cod-RNA} \quad (d) \text{YearPredictionMSD} \]

Figure 2: For a given number of samples, Recursive RLS-Nyström yields approximations with lower error, measured by \( ||\mathbf{K} - \tilde{\mathbf{K}}||_2 \). Error is plotted on a logarithmic scale, averaged over 10 trials.

Figure 2 confirms that Recursive RLS-Nyström consistently obtains substantially better kernel approximation error than the other methods. As we can see in Figure 3, with the exception of \textsc{YearPredictionMSD}, the better quality of the landmarks obtained with Recursive RLS-Nyström also translates into runtime improvements. While the cost per sample is higher for our method at \( O(nd + ns) \) time versus \( O(nd + s^2) \) for uniform Nyström and \( O(nd) \) for random Fourier features, since RLS-Nyström requires fewer samples it more quickly obtains \( \tilde{\mathbf{K}} \) with a given accuracy. \( \tilde{\mathbf{K}} \) will also have lower rank, which can accelerate processing in downstream applications.
Figure 3: Recursive RLS-Nyström obtains a fixed level of approximation faster than uniform sampling, only underperforming on YearPredictionMSD. Results for random Fourier features are not shown: while the method is faster, it never obtained high enough accuracy to be directly comparable.

In Appendix G, we show that that runtime of RLS-Nyström can be further accelerated, via a heuristic approach that under-samples landmarks at each level of recursion. This approach brings the per sample cost down to approximately that of random Fourier features and uniform Nyström while nearly maintaining the same approximation quality. Results are shown in Figure 4.

For datasets such as Covertype in which Recursive RLS-Nyström performs significantly better than uniform sampling, so does the accelerated method (see Figure 4b). However, the performance of the accelerated method does not degrade when leverage scores are relatively uniform – it still offers the best runtime to approximation quality tradeoff (Figure 4c).

We note further runtime optimizations may be possible. Subsequent work extends fast ridge leverage score methods to distributed and streaming environments [CLV17]. Empirical evaluation of these techniques could lead to even more scalable, high accuracy Nyström methods.

5.2 Additional Empirical Results

In Appendix G we verify the usefulness of our kernel approximations in downstream learning tasks. While full kernel methods do not scale to our large datasets, Recursive RLS-Nyström does since its runtime depends linearly on $n$. For example, on YearPredictionMSD the method requires 307 sec. (averaged over 5 trials) to build a 2,000 landmark Nyström approximation for 463,716 training points. Ridge regression using the approximate kernel then requires 208 sec for a total of 515 sec. These runtime are comparable to those of the very fast random Fourier features method, which underperforms RLS-Nyström in terms of regression and classification accuracy.

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


