

A Survey of Fast Kernel Sketching Algorithms

Course Project Report for EE381K-6 Estimation Theory (Spring 2016)

Shanshan Wu EID: sw33323

May 6, 2016

Abstract

Kernel based algorithms suffer from quadratic runtime for computing kernel matrices, which makes these algorithms hard to scale in the large scale setting. One common way of avoiding quadratic runtime is to compute a sketch matrix and use it in place of the original kernel matrix. In this survey we compare two categories of pass-efficient randomized algorithms for sketching kernel matrices: random features method and Nyström method. Comparisons are performed in terms of their theoretical bounds and also their empirical performance. In particular, we implement three sketching methods in Matlab, and find that leverage-score based Nyström method outperforms the other methods. On the other hand, method based on random features has the advantage of taking only one pass over the data, and hence can be used in the streaming setting. We conclude this survey by proposing a future research direction towards improving the efficiency-accuracy tradeoff for random features methods.

1 Introduction

Kernel methods such as support vector machine and kernel ridge regression are widely used in machine learning and data analysis. The central idea is commonly referred as the "kernel trick", which allows us to implicitly work with high-dimensional (or even infinite dimensional) feature mapping by only dealing with pairwise inner products between data points. A kernel matrix K is n -by- n positive semi-definite (n is the number of data points), with each entry being the inner product between two data points in the mapped space. Computing K needs quadratic runtime, so all kernel based algorithms require at least quadratic time complexity. This partially hinders the use of kernel methods in large-scale settings. In this survey we compare two types of pass-efficient kernel sketching methods that produce an approximation of the kernel matrix in sub-quadratic time. Here "pass-efficient" means that these algorithms only require a few constant number of passes over the input data. Pass-efficient algorithms are desirable in the setting when data are stored in disk, since disk I/O usually dominates the total runtime.

Before introducing the kernel approximation methods, we first present a framework of standard kernel based learning problems, and one specific example called Kernel Ridge Regression. Consider the following learning problem: given n data points $(x_1, y_1), \dots, (x_n, y_n)$ in $\mathcal{X} \times \mathcal{Y}$, where \mathcal{X} is the input space (x_i are column vectors), and \mathcal{Y} denotes the space of outputs/labels (y_i are scalars), the goal is to learn a function f such that $(x_i, f(x_i))$ best fits the given data. In particular, we are interested in the nonparametric regression problem setting [5], where our search space is limited to a reproducing kernel Hilbert space (RKHS) \mathcal{F} . More formally, this learning problem can be formulated as a minimization problem:

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n l(y_i, f(x_i)) + \frac{\lambda}{2} \|f\|_{\mathcal{F}}^2, \quad (1)$$

where $\|f\|_{\mathcal{F}}$ is a regularization term, and $l : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ corresponds to a loss function. Different learning problems have different loss functions, e.g., mean squared loss function

is used in Kernel Ridge Regression, logistic loss is used in Kernel Logistic Regression, and hinge loss is used in the setting of Support Vector Machine.

It is known that any RKHS \mathcal{F} is associated with a kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ and a corresponding feature map $\phi(x) : \mathcal{X} \rightarrow \mathcal{F}$, such that $k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{F}}$ for every $x, x' \in \mathcal{X}$. Using this special structure of \mathcal{F} (more specifically, using the representer theorem of RKHS), we can transform problem (1) into an optimization problem over \mathbb{R}^n :

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n l(y_i, (K\alpha)_i) + \frac{\lambda}{2} \alpha^T K \alpha, \quad (2)$$

where K is a kernel matrix with $K_{ij} = k(x_i, x_j)$. Commonly used kernels include: 1) RBF kernel, where $k(x_i, x_j) = e^{-\|x_i - x_j\|^2 / (2\sigma^2)}$ with a free parameter σ ; and 2) polynomial kernel, where $k(x_i, x_j) = (x_i^T x_j + c)^d$ for two free parameters c and d . A kernel function evaluated at two data points can be regarded as a special similarity measure between them. However, unlike other similarity measures such as cosine similarity, a kernel matrix K must be positive semi-definite, since by definition $k(x_i, x_j)$ can be decomposed as the inner product between $\phi(x_i)$ and $\phi(x_j)$.

The formulation in (2) is quite general. To be more concrete, in this survey we focus on the Kernel Ridge Regression (KRR) problem, which corresponds to l being squared loss:

$$\text{Kernel Ridge Regression (KRR)} : \min_{\alpha \in \mathbb{R}^n} \frac{1}{2n} \sum_{i=1}^n (y_i - (K\alpha)_i)^2 + \frac{\lambda}{2} \alpha^T K \alpha. \quad (3)$$

One nice thing about the KRR problem is that it has a closed-form solution. The solution to (3) and the corresponding estimated output at point x is given by

$$\hat{f}_K(x) = \sum_i \alpha_i K(x, x_i), \text{ where } \alpha = (K + n\lambda I)^{-1} y. \quad (4)$$

Here we use subscript in \hat{f}_K to emphasize its dependence on the kernel matrix K . We assume that the underlying data model is $y_i = f^*(x_i) + \xi_i$, for $i = 1, \dots, n$, where ξ_i is independent

Gaussian random variable with zero-mean and variance σ^2 . Let $\mathcal{R}(\hat{f}_K)$ be the mean squared error of the estimator \hat{f}_K . According to [3], $\mathcal{R}(\hat{f}_K)$ can be decomposed into a bias term and a variance term:

$$\begin{aligned} \mathcal{R}(\hat{f}_K) &:= \frac{1}{n} E_{\xi} \|f^* - \hat{f}_K\|^2 = n\lambda^2 \|(K + n\lambda I)^{-1} f^*\|^2 + \frac{\sigma^2}{n} \text{Tr}(K^2(K + n\lambda I)^{-2}) \\ &= \text{bias}(K)^2 + \text{variance}(K). \end{aligned} \tag{5}$$

Clearly solving problem (3) requires computing K , and hence the total runtime is at least $O(n^2)$, i.e., quadratic in the number of observations n . This is usually not acceptable for large-scale learning. To achieve sub-quadratic running time, the common approach is to compute a smaller sketch matrix Z of size n -by- p such that $ZZ^T \approx K$, and use ZZ^T in place of K when solving (2). The sketching dimension p captures the tradeoff between complexity and accuracy, and is normally different for different sketching algorithms.

In this survey we consider four pass-efficient algorithms in the literature: random Fourier features [7], random Maclaurin features [6], vanilla Nyström method [9] [4] [3], and leverage score-based Nyström method [2] [1] [8]. In Section 2 we present the basic ideas of these algorithms and their theoretical bounds. In Section 3 we compare their empirical performance on synthetic data. This survey concludes in Section 4, where we provide suggestions on future work.

2 Kernel Sketching Algorithms

Recall that the goal is to avoid computing the entire kernel matrix K by replacing K with a small sketch matrix $Z \in \mathbb{R}^{n \times p}$ in solving the kernel learning problems. To see why an n -by- p sketch matrix helps achieve sub-quadratic time complexity, note that since $ZZ^T \approx K$, the sketch matrix Z can be regarded as features representing the original data points. As a consequence, the original kernel learning problem can be transformed to a linear learning

problem. More specifically, substituting Z into the KRR problem (3) gives

$$\min_{w \in \mathbb{R}^p} \frac{1}{2n} \sum_{i=1}^n (y_i - (Zw)_i)^2 + \frac{\lambda}{2} \|w\|^2. \quad (6)$$

Given Z , the above ridge regression problem can be solved in $O(np^2)$ time, which is linear in the number of observations. Therefore, if Z can be computed in sub-quadratic time, then the total runtime will be sub-quadratic in n . In this section we present the basic ideas of two types of methods for sketching a kernel matrix. A summary of these methods is given in Table 1.

Table 1: A summary of the sketching algorithms considered in this survey

Algorithms	No. of Passes	Major Pros and Cons
Random Fourier Features [7]	one	applicable in the streaming setting; limited to shift-invariant kernels
Random Maclaurin Features [6]	one	applicable in the streaming setting; limited to dot-product kernels
Vanilla Nyström [9] [4] [3]	two	no limitation on kernel types; inapplicable in the streaming setting
Leveraged Nyström [2] [1] [8]	at least four	tighter theoretical bound, no limitation on kernel types; inapplicable for streaming data

2.1 Random Features Approach

The random features approach is based on the key observation that a kernel function can be well approximated by sampling a subset of basis functions, using the probability measure induced by this kernel function. Since the probability distribution depends only on the kernel function, the algorithm can be performed in an online fashion, i.e., the random features can be generated as soon as a new data point comes.

2.1.1 Random Fourier Features

First proposed by Rahimi and Recht [7], the algorithm builds upon the Bochner’s theorem from harmonic analysis, which says that a positive definite shift-invariant kernel function has non-negative Fourier transform. Specifically, let us denote the shift-invariant kernel function $k(x, y)$ as $k(x - y)$, a function on \mathbb{R}^d , we can then express $k(x - y)$ as

$$k(x - y) = \int_{\mathbb{R}^d} p(\omega) e^{j\omega^T(x-y)} d\omega = \int_{\mathbb{R}^d} p(\omega) (\cos(\omega^T x)\cos(\omega^T y) + \sin(\omega^T x)\sin(\omega^T y)) d\omega, \quad (7)$$

where $p(\omega)$ is guaranteed to be non-negative. If $k(x - y)$ is properly scaled, $p(\omega)$ becomes a valid probability measure. Then the above equation provides a way of constructing new representations for each data point $x \in \mathbb{R}^d$. Let $z(x) \in \mathbb{R}^{2D}$ be the $2D$ -dimensional random Fourier features for data point x , then $z(x)$ can be constructed in the following way:

1. Draw i.i.d. samples $\omega_1, \dots, \omega_D$ from $p(\omega)$.
2. Compute $z(x) := \frac{1}{\sqrt{D}} [\cos(\omega_1^T x), \dots, \cos(\omega_D^T x), \sin(\omega_1^T x), \dots, \sin(\omega_D^T x)]$.

Let $Z \in \mathbb{R}^{n \times 2D}$, with $z(x_i)$, $i = 1, \dots, n$ as row vectors, then $ZZ^T \approx K$, i.e., Z is a sketch matrix of K . As proved by [7], uniform convergence can be achieved by random Fourier features based sketching method. Mathematically, $\sup_{x,y \in \mathcal{M}} |z(x)z(y)^T - k(x,y)| \leq \epsilon$ with constant probability if $D = \Omega(\frac{d}{\epsilon^2} \log \frac{\sigma_p \text{diam}(\mathcal{M})}{\epsilon})$, where σ_p^2 is the second moment of $p(\omega)$, \mathcal{M} is a compact set in \mathbb{R}^d with diameter $\text{diam}(\mathcal{M})$.

2.1.2 Random Maclaurin Features

Random Fourier features are limited to shift-invariant kernels. A lot of subsequent works develop similar algorithms for handling other families of kernels, among which one important work is given by Kar and Karnick [6]. They present random feature mapping for dot product kernels, i.e., $k(x, y) = f(\langle x, y \rangle)$, where f is a function $\mathbb{R} \rightarrow \mathbb{R}$. Their algorithm builds upon a classical result in harmonic analysis by Schoenberg, which guarantees that f has a Maclaurin

expansion with only non-negative coefficients, i.e.,

$$f(x) = \sum_{n=0}^{\infty} a_n x^n, \text{ where } a_n = \frac{f^{(n)}(0)}{n!} \geq 0. \quad (8)$$

Similar to the random Fourier features method, given a data point $x \in \mathbb{R}^d$, let $z(x) = \frac{1}{\sqrt{D}}[z_1(x), z_2(x), \dots, z_D(x)]$ be its D -dimensional random Maclaurin features. Then each of $z_i(x)$ ($i = 1, \dots, D$) can be constructed as follows:

1. Choose a non-negative integer N with probability $\mathbb{P}(N = n) = \frac{1}{2^{n+1}}$.
2. Generate N d -dimensional vectors $w_1, \dots, w_N \in \{-1, 1\}^d$, each w_i iid Bernoulli-(1/2).
3. Compute $z_i(x) = \sqrt{a_N 2^{N+1}} \prod_{j=1}^N w_j^T x$.

The random Maclaurin features generated by the above algorithm achieve a similar uniform convergence bound as that achieved by random Fourier features. As shown in [6], given a dot-product kernel function $k(x, y) = f(\langle x, y \rangle)$, $\sup_{x, y \in \mathcal{M}} |z(x)z(y)^T - k(x, y)| \leq \epsilon$ with probability at least $1 - \delta$ if $D = \Omega(\frac{d}{\epsilon^2} \log \frac{1}{\epsilon \delta})$.

2.2 Nyström Method

First proposed by Williams and Seeger [9] and further improved by Drineas and Mahoney [4], the Nyström method can be used to compute a low-rank approximation to any positive semidefinite (PSD) matrix. Since a kernel matrix is PSD, this method can be naturally applied to approximate K . The algorithm works as follows.

1. Pick p columns of K , concatenate these columns to form a matrix $C \in \mathbb{R}^{n \times p}$.
2. Let $W \in \mathbb{R}^{p \times p}$ be the overlap between C and C^T in K , then outputs $\hat{K} = CW^\dagger C^T$.

Here W^\dagger represents the Moore-Penrose pseudo inverse of W . The output \hat{K} is a low rank approximation of K . The corresponding sketching matrix Z can then be obtained by computing a matrix $Z \in \mathbb{R}^{n \times p}$ such that $ZZ^T = \hat{K}$. The only remaining problem is how to

subsample columns from K . Different sampling distribution lead to different performance guarantees. Here we present two sampling schemes.

2.2.1 Vanilla Nyström

This method uses uniform distribution for column subsampling. Suppose \hat{K} is used in solving the KRR problem (3), Bach [3] shows that if the number of sampled columns satisfies $p = O(n \|\text{diag}(K(K + n\lambda)^{-1})\|_{\infty} / \epsilon)$, then the expected empirical risk (see Eq. (5) for the definition of empirical risk) is within a factor of $1 + \epsilon$ of the optimal risk.

The vanilla Nyström can be executed in two passes over the data. The first pass is used to obtain a subset of data points that are uniformly sampled from the dataset. The second pass is used to compute the exact entries of the kernel matrix corresponds to the sampled columns. Since two passes are needed, the vanilla Nyström method cannot be applied when data are coming from live streams.

2.2.2 Leverage-score Based Nyström

This method is first proposed by Alaoui and Mahoney [2] [1]. They improve upon Bach’s result [3] by using a biased column sampling distribution. The intuition is that compared to uniform sampling, statistical leverage scores may capture more structural information of the kernel matrix. Here the leverage scores are specifically defined for the KRR problem (3). Given a kernel matrix K and regularization parameter λ , the λ -ridge leverage scores are defined as

$$l_i(\lambda) = (K(K + n\lambda I)^{-1})_{ii}, \quad i = 1, \dots, n, \quad (9)$$

where $(K(K + n\lambda I)^{-1})_{ii}$ represents the i -th diagonal entry of $K(K + n\lambda I)^{-1}$. Since exact computation of λ -ridge leverage scores requires the full kernel matrix, which we try to avoid, Alaoui and Mahoney [2] propose a randomized algorithm for approximating the leverage scores. According to [2], by sampling the columns of K proportionally to the approximated λ -ridge leverage scores, a similar $(1 + \epsilon)$ statistical guarantee can be achieved using only

$p = O(\text{Tr}(K(K + n\lambda)^{-1}))/\epsilon$ columns. Note that this bound is better than that achieved by the vanilla Nyström method since $\text{Tr}(M) \leq n\|M\|_\infty$ for any matrix M .

The leverage-score based Nyström algorithm proceeds in two stages. In the first stage, approximated λ -ridge leverage scores are computed. In the second stage, columns of the kernel matrix are subsampled with probabilities proportional to the approximated leverage scores. The sampled column vectors are then used to form a sketched matrix under the standard Nyström framework. Since each stage of the algorithm needs at least two passes over the dataset, in total this method requires at least four passes, and hence cannot be used in the streaming setting.

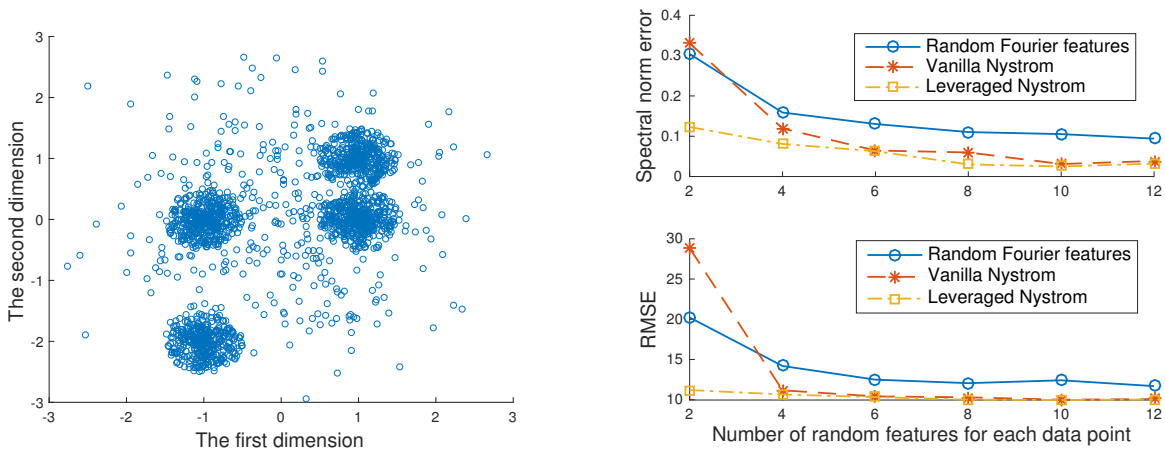
3 Simulations

We implement three sketching algorithms in Matlab and compare their empirical performance on synthetic data. The three sketching methods are random Fourier features, vanilla Nyström, and leverage-score based Nyström. We follow paper [10] to generate a synthetic dataset of $n = 2000$ data points in \mathbb{R}^{100} . The dataset consists of 1600 data points uniformly distributed in four balls of radius 0.5, and 400 points drawn randomly from a standard Gaussian distribution in the first two dimensions (see Figure 1a). The rest 98 coordinates of all data points are random numbers drawn from standard Gaussian distribution. We focus on the RBF kernel $k(x_i, x_j) = e^{-\|x_i - x_j\|^2 / (2\sigma^2)}$, with $\sigma = 6$. The observations are generated as $y_i = (K\alpha^*)_i + \xi_i$, for $i = 1, \dots, n$, where K is the RBF kernel matrix, α^* represents the true model coefficients (which is constructed by sampling from standard Gaussian distribution), and ξ_i is drawn randomly from Gaussian distribution with zero-mean and variance 0.01.

In Figure 1b we compare the three kernel sketching algorithms in terms of spectral norm error and estimation error. In the upper figure, we use y-axis to denote relative spectral norm error $\|K - \hat{K}\|/\|K\|$, where $\|K\|$ is the spectral norm of K . The x-axis represents the number of random features (or the number of randomly sampled columns) for each data

point. In the lower figure, we compare the estimation error achieved by using the sketched kernel matrices in place of the original kernel matrix when solving KRR problem (3) with $\lambda = 0.1$. The y-axis represents the root mean squared error $\sqrt{\|y - \hat{y}\|^2/n}$, where \hat{y} denotes the estimated response, and n is the total number of observations.

As shown in Figure 1b, the Nyström method outperforms the random Fourier features method as long as there are enough sampled columns. This result agrees with that in [10], and the main reason is that random features based methods are data independent while Nyström methods are data dependent, and hence may capture more structural information than the random features methods. Comparing the two variants of Nyström methods, we see that the leverage-score based Nyström method achieve better statistical performance than the vanilla Nyström method, especially when the number of sampled columns is small. This is because unlike uniform column subsampling, columns sampled according to leverage scores are better aligned to the non-uniformities structure of the kernel matrix. However, the major drawback of leverage-score based Nyström method is that it needs at least four passes over the dataset (two passes are used for estimating the λ -ridge leverage scores).



(a) Synthetic data

(b) Accuracy comparison

Figure 1: (a) A scatter plot of synthetic data in the first two dimensions. (b) Accuracy comparison between random Fourier features method, vanilla Nyström method, and the leverage-score based Nyström method: relative spectral norm error (upper), and root mean squared estimation error for the KRR problem (lower).

4 Conclusion

In this survey we compare two types of kernel matrix sketching methods that have been successfully used to speed up large scale kernel learning problems. The ideas behind the two approaches are quite different: random features based methods use the fact that some specific kernel functions can be written as an infinite sum of basis functions with non-negative coefficients, while Nyström methods use column subsampling to obtain low rank approximation of positive-semidefinite matrices. Both methods have strength and weakness. Random features based methods can be used for processing live data streams (since the features' sampling distribution depends only on the kernel function and is independent of the actual data), but are limited to certain types of kernel functions. In contrast, Nyström methods are generic but usually require two or more passes over the data, and hence cannot be used in the streaming setting.

Theoretically, both methods have the same $1/\epsilon^2$ dependence on the norm of error matrix. In practice, we see that under the same sampling complexity, the sketched kernel matrix obtained by Nyström methods has better statistical performance than those based on random features methods. This can be seen as a tradeoff between pass-efficiency and accuracy: random features methods are performed in one pass over the data, while Nyström methods use two or more passes, and hence have more chances exploring the data.

One future research direction is to develop new kernel sketching algorithm that can be used for streaming data but has better accuracy than the current random features based methods. As implied by the literature, random features methods generally have worse accuracy than Nyström methods, and one possible reason is that the random features are sampled from a distribution that is independent from the actual data. To alleviate this data independency and meanwhile maintain the desired one-pass efficiency, we propose to add a data dependent learning process for the random features method. This process will continuously examine the importance of randomly generated features (i.e., randomly sampled basis functions), and dynamically remove less important features as more data come.

References

- [1] A. Alaoui and M. W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*, pages 775–783, 2015.
- [2] A. E. Alaoui and M. W. Mahoney. Fast randomized kernel methods with statistical guarantees. *arXiv preprint arXiv:1411.0306*, 2014.
- [3] F. Bach. Sharp analysis of low-rank kernel matrix approximations. *arXiv preprint arXiv:1208.2015*, 2012.
- [4] P. Drineas and M. 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.
- [5] L. Györfi, M. Kohler, A. Krzyzak, and H. Walk. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.
- [6] P. Kar and H. Karnick. Random feature maps for dot product kernels. *arXiv preprint arXiv:1201.6530*, 2012.
- [7] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.
- [8] A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, pages 1648–1656, 2015.
- [9] C. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *Proceedings of the 14th Annual Conference on Neural Information Processing Systems*, 2001.

- [10] T. Yang, Y.-F. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. In *Advances in neural information processing systems*, pages 476–484, 2012.