7. Appendix

7.1. Proof of lower approximation error

Theorem 1. Using Algorithm 1 to generate \( m \) landmark points, we can guarantee that the approximation quality will become better than the traditional Nyström approximation with initial \( s \) landmark points:

\[
\|G - \bar{G}\| \leq \|G - \check{G}\|, \quad (15)
\]

where \( \bar{G} \) and \( \check{G} \) are the approximation of \( G \) from standard Nyström and Algorithm 1 respectively.

Proof. Let us first compare our method with standard Nyström. The generalization to other sampling strategies based Nyström is straightforward. Let \( G \) denote the kernel matrix form on the \( n \) data points, and suppose \( s \) landmark points \( x_1, \ldots, x_s \) are selected uniformly at random from the data. Let us define the sampling matrix \( S \in \mathbb{R}^{n \times s} \) to be a zero-one matrix where \( S_{ij} = 1 \) if \( i \)-th sample in the dataset is selected as landmark point. \( C \) is a \( n \times s \) matrix consisting of the corresponding \( s \) columns selected from \( G \) and \( W \) consists of the kernel matrix formed by these \( s \) landmark points. So by standard Nyström, \( \bar{G} = CW^+G^T \), \( C = GS \) and \( W = S^TGS \).

Using \( m_1, \ldots, m_s \) as initial landmark points in Algorithm 1, after fast transforms, we totally have \( m = sd \) landmark points \( v_1, \ldots, v_m \), of which the last \( s \) points are the original landmark points and the rest \( m - s \) are new landmark points. Assume the new kernel matrix \( G_H \) is the kernel matrix on the union of the original \( n \) data points and \( m - s \) new added landmark points. So \( G \) is a block in \( G_H \). Similarly we define \( S_H, C_H, \) and \( W_H \) as sampling matrix, \( m \) sampled columns in \( G_H \) and kernel matrix formed by \( m \) landmark points respectively. So \( C_H = G_HS_H \) and \( W_H = S_H^TG_HS_H \). Let the decomposition of \( G_H \) be \( G_H = L_H^TL_H \). So

\[
G_H = L_H^TL_H = [\bar{L}^T L^T][\bar{L} L] = [\bar{L}^T L L^T TL]. \quad (16)
\]

Since \( G \) is a block in \( G_H \), the decomposition of \( G \) is \( L^T L \).

Since \( C_H = G_HS_H = L_H^TL_HS_H \) and let the singular value decomposition of \( L_HS_H \) be \( U_HS_HV_H^T \), \( C_H = L_H^TU_HS_HV_H^T \). Also we have

\[
W_H = S_H^TG_HS_H = S_H^TL_H^TL_HS_H = V_H \Sigma_H^2 V_H^T. \quad (17)
\]

The Nyström approximation on \( G_H \) is written as

\[
G_H = C_HW_H^+C_H^T = L_H^TU_H \Sigma_H V_H^T \Sigma_H^2 V_H^T V_H \Sigma_H U_H^T L_H
\]

\[
= L_H^TU_H U_H^T L_H. \quad (18)
\]

So we have

\[
\]

The Nyström approximation error on the original \( n \) data points or \( G \) part is

\[
(G_H - C_HW_H^+C_H^T)_G = L^T L - L^T U_H U_H^T L = (L - U_H U_H^T L)^T (L - U_H U_H^T L).
\]

According to Lemma 1 in [Drineas & Mahoney 2005], we have the standard Nyström approximation on \( G \) as

\[
G - CW^+C^T = L^T L - L^T U_U^T L = (L - U_U^T L)^T (L - U_U^T L).
\]

where \( LS \)'s SVD is \( U \Sigma V^T \).

Since \( U \) is the basis for the range space of \( LS \) and \( U_U \) is the basis for the range space of \( L_H S_H \), so \( range(U) \subseteq range(U_H) \). According to the proposition 8.5 in [Halko et al. 2011], we have

\[
\|L - U_H U_H^T L\|_2 \leq \|L - U_U^T L\|_2, \quad (22)
\]

so

\[
\|(G_H - C_HW_H^+C_H^T)_G\| \leq \|G - CW^+C^T\|, \quad (23)
\]

or

\[
\|G - \bar{G}\| \leq \|G - \check{G}\|. \quad (24)
\]

\[ \square \]

7.2. Lemma 1

Lemma 1. If the kernel function can be written as (3), assume the maximum distance between the samples and the original point is a bounded number \( R \), and \( f, g \) are differentiable, then

\[
K(a, b)^2 - K(c, d)^2 \leq \eta (\|a - c\|^2 + \|b - d\|^2) \quad (25)
\]

for any \( a, b, c, d \in \mathbb{R}^d \), where

\[
\eta = 4M_f^2 L_f^2 R^2 + 4M_g^2 M_f^2 L_f^2,
\]

where \( M_f = \max_{\|x\| \leq R} |f(x)|, M_g = \max_{\|u\| \leq R} |g(u)|, L_f = \max_{\|x\| \leq R} |f'(x)|, L_g = \max_{\|u\| \leq R} |g'(u)|. \)
Proof. For any $a, b, c, d \in \mathbb{R}^d$, we have

$$(K(a, b) - K(c, d))^2$$

$$= \left( f(a)g(b)g(a^T b) - f(c)g(d)g(c^T d) \right)^2$$

$$= \left( f(a)g(b)g(a^T b) - f(c)g(d)g(a^T b) 
+ (f(c)g(d)g(a^T b) - f(c)g(d)g(c^T d)) \right)^2$$

$$\leq 2 \left( g(a^T b)(f(a)f(b) - f(c)f(d)) \right)^2$$

$$+ 2 \left( f(c)g(d)(g(a^T b) - g(c^T d)) \right)^2$$

$$\leq 2M_g^2 \left( f(a)f(b) - f(c)f(d) \right)^2$$

$$+ 2M_f^2 \left( g(a^T b) - g(c^T d) \right)^2.$$  

We can then bound each term by

$$\left( f(a)f(b) - f(c)f(d) \right)^2$$

$$\leq \left( f(a)f(b) - f(c)f(b) + f(c)f(b) - f(c)f(d) \right)^2$$

$$\leq 2(f(a) - f(c))^2 f(b)^2 + 2(f(b) - f(d))^2 f(c)^2$$

$$\leq 2M_f^2 \left( f(a) - f(c) \right)^2 + (f(b) - f(d))^2 \right)^2$$

$$= 2M_f^2 \left( f'(\xi)^2 \|a - c\|^2 + f'(\xi)^2 \|b - d\|^2 \right)$$

$$\leq 2M_f^2 L_f^2 \left( \|a - c\|^2 + \|b - d\|^2 \right)$$

Similarly, we have

$$\left( g(a^T b) - g(c^T d) \right)^2$$

$$= \left( g'(\xi)(a^T b - c^T d) \right)^2$$

$$\leq L_g^2 \left( a^T b - c^T b + c^T b - c^T d \right)^2$$

$$= L_g^2 \left( (a - c)^T b + (b - d)^T c \right)^2$$

$$\leq 2L_g^2 \left( \|a - c\|^2 + 2\|b - d\|^2 \right)$$

$$\leq 2L_g^2 \Gamma^2 \left( \|a - c\|^2 + \|b - d\|^2 \right)^2$$

This proves (25). \qed

7.3. Parameters for the experimental results

- All the experiments were conducted on a machine with an Intel Xeon X5440 2.83GHz CPU and 32G RAM. We tried to have the best implementation for each algorithm. Fast-Nys, DC-Pred++, Nys, KNys, RKS, Fastfood are all implemented in C sharing the same modules. LDKL is the highly optimized C++ implementation published along with the original paper [Jose et al., 2013].
- The degree for the polynomial kernel and homogeneous kernel is set to be 3.
- We do data normalization with mean to be 0 and variance to be 1 before running our algorithms.
- When working on fast prediction experiments, we first form the low-rank approximation for the kernel matrix and apply liblinear to perform the classification.
- For fast prediction parameters($\gamma$ is the width parameters for Gaussian kernel and $C$ is the regularization term in Liblinear SVM):
  - cifar: $\gamma = 2^{-10}$, $C = 64$;
  - mnist: $\gamma = 2^{-10}$, $C = 128$;
  - a9a: $C = 32$;
- For kernel approximation:
  - magic: $\gamma = 0.01$
  - ijenn: $\gamma = 0.01$
  - webspam: $\gamma = 1$
- When working on prediction, we use random samples as the initial landmarks for Fast-Nys. The number of initial landmarks ranges from 2 to 10.
- When using kmeans Nyström, we randomly sample 10000 data samples to perform clustering.
- For LDKL, for a fair comparison, we disable the SSD operation.
- We use an alternating minimization algorithm to find the seeds in our algorithm. The algorithm usually converges to a reasonably good solution in 10 iterations, so we fix the number of iterations to be 10 for all the experiments. For example, on MNIST dataset with $k=10$, the initial objective function value (using random samples) is 1750260, after 10 iterations it drops to 90041, and the converged solution has objective function value 89872.

7.4. Comparison with other kernel approximation methods

We show the comparison between fast-Nys with leverage score [Gittens & Mahoney, 2013] and entropy based landmark points [Brabanter et al., 2010] in Nyström approximation and random feature [Rahimi & Recht, 2007].
Figure 5. Low-rank kernel approximation results. $x$-axis is the time and $y$ axis shows the relative kernel approximation error. Methods with approximation error above the top of $y$-axis are not shown. (a) compares Fast-Nys with sampling landmark points based on leverage score (Gittens & Mahoney 2013a). Since this method needs to compute the entire kernel, it is much slower than our method. (b) compares Fast-Nys with entropy based landmark points based Nyström approximation (Brabanter et al. 2010) and random feature (Rahimi & Recht 2008). We can also observe Fast-Nys achieves much lower approximation error than these two methods.