Fast Sketching of Polynomial Kernels of Polynomial Degree

Zhao Song, David P. Woodruff, Zheng Yu and Lichen Zhang

Princeton University, Institute of Advanced Studies and Carnegie Mellon University

Introduction

Kernel method is an important prototype in solving many non-parametric machine learning problems, such as kernel ridge regression, support vector machine (SVM), principal component analysis (PCA) and many others. Suppose there are n data points $\{x_i\}_{i=1}^n$ each of dimension d, the goal is to compute an $n \times n$ matrix K where each entry $K_{i,j} := k(x_i, x_j)$ where k is some kernel function and K is a PSD matrix. Naively, computing K involves n^2 evaluations of function k, which takes $O(n^2 d)$ time. In this work, we consider the high-dimensional or over-parametrized setting, where $d = \mathrm{poly}(n)$, and data matrix $X \in \mathbb{R}^{d \times n}$ is dense. In this scenario, our goal is to spend nearly linear time on term nd to read the input matrix X and shave off the dependence on d in later computations. This is of particular interests in fields such as natural language processing, computational biology and training over-parametrized neural network.

Instead of computing kernel matrix directly, we consider designing algorithm to compute or approximate a *polynomial kernel*. The clear advantage is many popular kernels, such as Gaussian kernel, arc-cosine kernel [2] and neural tangent kernel [4] can be well-approximated via their Taylor expansions, which means an efficient algorithm for polynomial kernel leads to efficient algorithm to these kernels. In fact, we classify kernels based on the coefficients in their Taylor expansions and design algorithms for them.

Main Problem

We first define the polynomial kernel. Let $x, y \in \mathbb{R}^d$, then polynomial kernel function of degree q is defined as $p_q(x, y) = \langle x, y \rangle^q$, and polynomial kernel is $P \in \mathbb{R}^{n \times n}$ where $P_{i,j} = p(x_i, x_j)$. Observe that $P_q = (X^{\otimes q})^\top X^{\otimes q}$ where $X^{\otimes q} \in \mathbb{R}^{d^q \times n}$ is defined with a lifting function $\phi : \mathbb{R}^d \to \mathbb{R}^{d^q}$ with $\phi(x)_{i_1,i_2,\ldots,i_q} = x^{\otimes q}_{i_1,i_2,\ldots,i_q} = x_{i_1} x_{i_2} \ldots x_{i_q}$ for $i_1, i_2, \ldots, i_q \in [d]$.

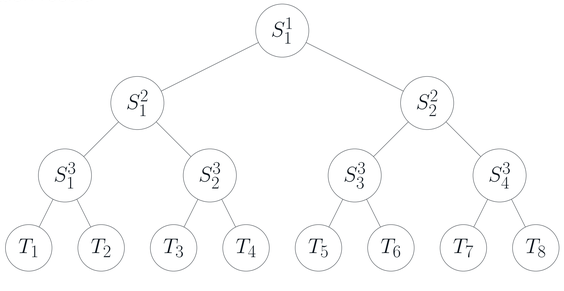
Goal: compute an ϵ-approximation of $X^{\otimes q}$ fast without explicitly forming $X^{\otimes q}$. By approximation, we mean compute a matrix $Z \in \mathbb{R}^{s \times n}$ where for any $y \in \mathbb{R}^n$, we have $y^\top Z^\top Z y \leq (1 \pm \epsilon) \cdot y^\top (X^{\otimes q})^\top X^{\otimes q} y$ which we will later denote it by $Z^\top Z \approx_\epsilon (X^{\otimes q})^\top X^{\otimes q}$.

Prior Works

There are many prior works addressing the problem of efficiently approximating the kernel matrix, such as using random Fourier features [6], Nystrom method [7, 5], oblivious sketching method [1] and adaptive leverage score sampling [8]. We remark that our method most resembles the work in [1], and we achieve superior runtime of computing such an approximation when $d = \mathrm{poly}(n)$ and matrix X is dense.

Our Result

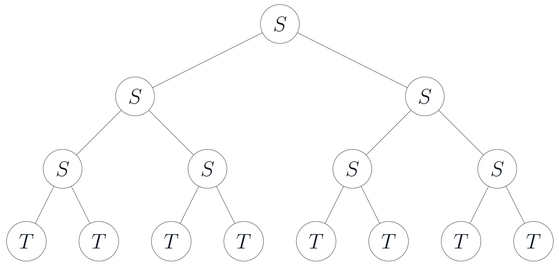
Before introducing our algorithm, we first study the algorithm in [1]. Given $x \in \mathbb{R}^d$, the goal is to compute an approximation to $\phi(x)$. Let $s \ll d^q$ be a smaller dimension, the algorithm proceeds in the form of a binary tree: it first samples q independent sketching matrices $T \in \mathbb{R}^{s \times d}$ and apply each of them to x. At each internal node, it picks an independent degree-2 tensor sketch $S_i : \mathbb{R}^s \times \mathbb{R}^s \to \mathbb{R}^s$ and then uses it to combine its two children. The root of the binary tree is the final computation result.



Our Result (cont.)

The advantage of above algorithm is it enjoys a great amount of independence, since it uses different sketches for each node of the tree. This gives the approximation a very strong guarantee in the expense of greater computation cost, since it needs a total of q applies of T's and q applies of S's.

In contrary, our algorithm only uses one copy of T and one copy of S. It computes Tx first then repeatedly applying the same S. In the language of binary tree, all leaves are the same and all internal nodes use the same sketch as illustrated in the diagram.



By doing so, we only need to perform $O(\log q)$ instead of $O(q)$ computations. Especially, we only need to compute one leaf node, shaving off a q factor on term nd and hence achieve a nearly linear dependence on nd. However, we only use constant amount of randomness, and therefore get a weaker guarantee (only spectral approximation of $X^{\otimes q}$). Nevertheless, it suffices for our applications of approximating various kernels and solving kernel ridge regression by composing with a smaller sketching matrix.

Theorem 1. *Let $q \in \mathbb{N}_+$ and $\epsilon, \delta \in (0, 1)$. For every $X \in \mathbb{R}^{d \times n}$, there exists a distribution over oblivious linear sketches $\Pi : \mathbb{R}^{d^q} \times \mathbb{R}^s$ such that if $s = \widetilde{\Theta}(\epsilon^{-2} n q^2)$, we have*

$$(\Pi X^{\otimes q})^\top (\Pi X^{\otimes q}) \approx_\epsilon (X^{\otimes q})^\top X^{\otimes q}.$$

Moreover, $\Pi X^{\otimes q}$ can be computed in time $\widetilde{O}(nd + \epsilon^{-2} n^2 p^2)$.

Applications

We highlight four major applications of our algorithm.

Approximate Gaussian kernels. As one of the most well-known and popular kernels, Gaussian kernel, defined as $G_{i,j} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{2}\right)$, enjoys the benefit that the coefficients in its Taylor expansion decay very fast $(\frac{1}{n!})$. This means in order to achieve a good spectral approximation to G, it suffices to approximate the first logarithmic number of terms in its Taylor expansion. We hence get the following theorem:

Theorem 2. *Let $r \in \mathbb{R}_+$ and $X \in \mathbb{R}^{d \times n}$ be the data matrix such that for any $i \in [n]$, $\|x_i\|_2 \leq r$ where x_i is the i-th column of X. Suppose $G \in \mathbb{R}^{n \times n}$ is the Gaussian kernel of X, then there exists an algorithm that computes a matrix $W_g(X) \in \mathbb{R}^{s \times n}$ in time $\widetilde{O}(\epsilon^{-2} n^2 q^3 + nd)$ such that for any $\epsilon > 0$,*

$$\Pr[W_g(X)^\top W_g(X) \approx_\epsilon G] \geq 1 - 1/\mathrm{poly}(n),$$

where $s = \widetilde{\Theta}(\epsilon^{-2} n q^3)$ and $q = \Theta(r^2 + \log(n/\epsilon))$.

Classify p-convergent kernels. In addition to Gaussian kernels, we classify a family of kernels based on the coefficients of their Taylor expansion:

Definition 3. *We say a kernel matrix K is p-convergent if its corresponding Taylor expansion is $\sum_{l=0}^\infty C_l \cdot (X^{\otimes l})^\top X^{\otimes l}$ where $C_l = (l+1)^{-\Theta(p)}$.*

As a direct extension to our result to Gaussian kernel, we get the following theorem:

Applications (cont.)

Theorem 4. *Let $r \in \mathbb{R}_+$ and $p > 1$ be an integer and $X \in \mathbb{R}^{d \times n}$ be the data matrix such that $\|x_i\|_2 \leq r$ for all $i \in [n]$. Suppose K is a p-convergent kernel matrix of X. Let $s = \widetilde{\Theta}(\epsilon^{-2} n q^3)$ and $q = \Theta(r^2 + (n/\epsilon)^{1/p})$. There exists an algorithm that computes a matrix $W_k(X) \in \mathbb{R}^{s \times n}$ in time $\widetilde{O}(\epsilon^{-2} n^2 q^3 + nd)$ such that with probability at least $1 - 1/\mathrm{poly}(n)$, $W_k(X)^\top W_k(X) \approx_\epsilon K$.*

We note that our classification also includes NTK, which is approximate a 1.5-convergent kernel. To get a fast algorithm for approximating NTK, we use a novel sampling scheme, where we exactly compute the first m terms in Taylor expansion and then sample m terms from the remaining terms. By carefully choosing the parameter m, we achieve a fast algorithm for NTK:

Theorem 5. *Let $X \in \mathbb{R}^{d \times n}$ be the data matrix with $\|x_i\|_2 \leq 1$ for any $i \in [n]$. Suppose $\mathrm{K} \in \mathbb{R}^{n \times n}$ is the NTK matrix of X. Then there exists an algorithm to compute a matrix $W_k(X)$ in time $\widetilde{O}(\epsilon^{-3} n^{11/3} + nd)$ such that with probability at least $1 - 1/\mathrm{poly}(n)$, $W_k(X)^\top W_k(X) \approx_\epsilon \mathrm{K}$.*

Kernel Linear System. Solving PSD system involving kernel matrix via preconditioning is another interesting line of application [3].

Theorem 6. *Let $G \in \mathbb{R}^{n \times n}$ be the Gaussian kernel matrix for $X \in \mathbb{R}^{d \times n}$ with $\|x_i\|_2 \leq 1$ for all $i \in [n]$. Let $G = Z^\top Z$ and κ denote the condition number of Z, then there exists an algorithm that with probability at least $1 - 1/\mathrm{poly}(n)$, computes an approximate solution $\widehat{x}$ such that $\|G\widehat{x} - y\|_2 \leq \epsilon \|y\|_2$ in $\widetilde{O}(\epsilon^{-2} n^2 \log(\kappa/\epsilon) + n^\omega + nd)$ time, where ω is the exponent of matrix multiplication.*

Kernel Ridge Regression. Solving kernel ridge regression (KRR) is one of the core tasks of applying kernel method. As we have noted before, our algorithm does not provide a very strong guarantee that directly supports solving KRR, however, by composing it with a sketching matrix with a much smaller size, we can solve KRR fast and with provable guarantee.

Theorem 7. *Let $\epsilon \in (0, 1)$, $p > 1$, $\lambda > 0$ and $X \in \mathbb{R}^{d \times n}$. If K is its degree-p polynomial kernel with statistical dimension $s_\lambda := \mathrm{tr}[K(K + \lambda I_n)^{-1}]$ where $\lambda < \epsilon^{-2} \lambda_{\max}(K)$, then we can compute Z such that $Z^\top Z \approx_\epsilon K$ in $\widetilde{O}(\epsilon^{-2} p^2 n^2 + nd)$ time. Moreover, there exists a matrix S with $m = \widetilde{O}(\epsilon^{-1} s_\lambda)$ rows such that the optimal solution x^* to $\|S(Z^\top Z x - y)\|_2^2 + \lambda \|Zx\|_2^2$ gives rise to a $(1 \pm \epsilon)$ approximation to the optimum of $\|Kx - y\|_2^2 + \lambda \|X^{\otimes p} x\|_2^2$. The KRR can be solved in time $\widetilde{O}(\epsilon^{-2} p^2 n(n + m^2) + n^\omega)$.*

Acknowledgments: D. Woodruff would like to thank partial support from NSF grant No. CCF-1815840, Office of Naval Research grant N00014-18-1-256, and a Simons Investigator Award.

References

[1] Thomas D Ahle et al. "Oblivious sketching of high-degree polynomial kernels". In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2020, pp. 141–160.

[2] Youngmin Cho and Lawrence K Saul. "Kernel methods for deep learning". In: *Advances in neural information processing systems (NIPS)*. 2009, pp. 342–350.

[3] Kurt Cutajar et al. *Preconditioning Kernel Matrices*. 2016. arXiv: 1602.06693 [stat.ML].

[4] Arthur Jacot, Franck Gabriel, and Clément Hongler. "Neural tangent kernel: Convergence and generalization in neural networks". In: *Advances in neural information processing systems (NeurIPS)*. 2018, pp. 8571–8580.

[5] Cameron Musco and Christopher Musco. "Recursive sampling for the nystrom method". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017, pp. 3833–3845.

[6] Ali Rahimi and Benjamin Recht. "Random Features for Large-Scale Kernel Machines." In: *NIPS*. Vol. 3. 4. Citeseer. 2007, p. 5.

[7] Christopher K. I. Williams and Matthias Seeger. "Using the Nyström Method to Speed Up Kernel Machines". English. In: *Advances in Neural Information Processing Systems 13 (NIPS 2000)*. Ed. by T.K. Leen, T.G. Dietterich, and V. Tresp. MIT Press, 2001, pp. 682–688.

[8] David P Woodruff and Amir Zandieh. "Near Input Sparsity Time Kernel Embeddings via Adaptive Sampling". In: *ICML*. arXiv preprint arXiv:2007.03927, 2020.