CS 839 Probability and Learning in High Dimension Lecture 5 - 02/09/2022

Lecture 5: Kernel Methods and Random Features

Lecturer: Yudong Chen *Scribe: Hao Yan*

In this lecture,[1] we will introduce kernel methods and random features. Kernel method is one of the most important ideas in the history of machine learning. Even these days kernel method still performs very well on certain tasks. On the theory side, a lot of efforts have been devoted to understand the connection between deep learning and kernel method (e.g. neural tangent kernel). However, it is well known that original kernel method suffers from scalability issue. It requires a lot of computation and storage. Random feature is a very clever idea to get around this scalability issue. We will also apply matrix Bernstein's inequality to provide error bounds.

1 Notation

A quick summary of the notation.

For a vector $u \in \mathbb{R}^d$, we use $\|u\|_2$ to denote its ℓ_2 norm. For a matrix $A \in \mathbb{R}^{d_1 \times d_2}$, we use $\|A\|_{\mathrm{F}}$ to denote its Frobenius norm and $\|A\|_{\mathrm{op}}$ its operator/spectral norm (i.e., the largest singular value of A).

If A is a $d \times d$ symmetric matrix and its eigenvalues are sorting as $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_d$, then we denote $\lambda_i(A)$ be the i-th largest eigenvalue of A, namely $\lambda_i(A) = \lambda_i$.

For two $d \times d$ symmetric matrices A, B, the positive-semidefinite ordering $A \succeq B$ means that $A - B$ is positive-semidefinite matrix, i.e., $\lambda_i(A - B) \geq 0, \forall i$. Note that $A \succeq B$ implies that $\lambda_i(A) \geq \lambda_i(B)$ for $i = 1, \ldots, d$.

2 Motivations: Ridge Regression and Kernelization

A simple example of kernel method is given by ridge regression. Consider the linear regression setting. For a feature matrix $X \in \mathbb{R}^{N \times d}$, we denote the ith row as x_i, which is the feature vector for the ith data point. Denote $y \in \mathbb{R}^N$ as the response vector. To fit a linear model between X and y, ridge regression solves the optimization problem

$$\begin{aligned} \hat{\beta} &= \arg\min_{\beta \in \mathbb{R}^d} \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2, \\ &= (X^\top X + \lambda I_{d \times d})^{-1} X^\top y, \\ &= X^\top (X X^\top + \lambda I_{N \times N})^{-1} y. \end{aligned} \tag{1}$$

The last identity is called the "dual form" of ridge regression solution, and can be proved by SVD or Woodbury inversion lemma. Given a new data point $x_0 \in \mathbb{R}^d$, our prediction for y_0 is

$$\begin{aligned} \hat{y}_0 = x_0^\top \hat{\beta} = x_0^\top X^\top & \underbrace{(X X^\top + \lambda I_{N \times N})^{-1} y}_{:= \hat{\alpha} \in \mathbb{R}^N}, \\ = \sum_{i=1}^N & \hat{\alpha}_i \langle x_0, x_i \rangle. \end{aligned} \tag{2}$$

[1] *Reading:*

- Section 6.5 in Tropp, *An Introduction to Matrix Concentration Inequalities*, `https://arxiv.org/abs/1501.01571`.
- Also relevant: Ali Rahimi, Benjamin Recht, *Random Features for Large-Scale Kernel Machines*, NeurIPS 2007, `https://people.eecs.berkeley.edu/~brecht/papers/07.rah.rec.nips.pdf`;
- Also relevant: Fanghui Liu, Xiaolin Huang, Yudong Chen, Johan Suykens, *Random Features for Kernel Approximation: A Survey on Algorithms, Theory, and Beyond*, T-PAMI 2022, `https://arxiv.org/abs/2004.11154`.

Remark Because $(XX^\top)_{ij} = \langle x_i, x_j \rangle$, an important observation here is that the prediction $\hat{y}_0$ only depends on inner products between data points.

The idea of kernel method is to replace every inner product $\langle x_i, x_j \rangle$ by some nonlinear function $\Phi(x_i, x_j)$, where $\Phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a user-specified kernel. This is called the "kernel trick". The function Φ can also be interpreted as some nonlinear similarity measure between x_i and x_j. In this way, we can extend ridge regression to the nonlinear setting also known as "kernel ridge regression". The same idea also applies to Support Vector Machine (SVM), PCA and beyond.

3 Kernels: Properties and Examples

In the following, we impose some assumptions on our kernel function Φ:

1. $\Phi(x, x) = 1$ for all $x \in \mathbb{R}^d$,
2. $\Phi(x, y) \in [-1, 1]$ for all $x, y \in \mathbb{R}^d$,
3. $\Phi(x, y) = \Phi(y, x)$.

Example 1 (Angular Kernel)**.**

$$\Phi(x, y) = \frac{2}{\pi} \arcsin \frac{\langle x, y \rangle}{\|x\|_2 \|y\|_2} = 1 - \frac{2\angle(x, y)}{\pi}. \tag{3}$$

This is also known as the "arc-cosine kernel". It is one example of the rotation-invariant kernels, since it only depends on the angle between two data points.

Example 2 (Gaussian/Radial Basis Function(RBF) Kernel)**.**

$$\Phi(x, y) = e^{-\alpha \|x-y\|_2^2/2}. \tag{4}$$

Gaussian kernel is one example of the shift-invariant kernel. A kernel is shift-invariant if it only depends on the difference of the two points, that is $\Phi(x, y) = \varphi(x - y)$ for some function φ.

Definition 1 (Kernel Matrix)**.** *Given N data points $x_1, \ldots, x_N \in \mathbb{R}^d$, the kernel matrix $G \in \mathbb{R}^{N \times N}$ is $G_{ij} = \Phi(x_i, x_j)$ for $i, j = 1, \ldots, N$.*

A kernel Φ is called *positive semidefinite* (p.s.d.) if G is p.s.d. for any finite dataset $\{x_i\}_{i=1}^N$. We can verify that both Gaussian kernel and Angular kernel are p.s.d.

Given a kernel Φ, we can "kernelize" ridge regression by replacing every inner product $\langle x_i, x_j \rangle$ in the prediction rule (2) by $\Phi(x_i, x_j)$. Doing so leads to the kernel ridge regression prediction rule:

$$\hat{y}_0 = \sum_{i=1}^{N} \hat{\alpha}_i \Phi(x_0, x_i), \qquad \text{where } \hat{\alpha} = (G + \lambda I_{N \times N})^{-1} y \in \mathbb{R}^N.$$

To use the kernel method, we need to construct the $N \times N$ kernel matrix G. It takes N^2 storage and requires $O(dN^2)$ operations to compute, which is very challenging when N is large. To get around with this, we want to find approximation of G. This is where the random features idea comes in.

4 Random Features

The idea was proposed in a paper in 2007 by Ali Rahimi and Benjamin Recht, both of them were at UW-Madison at that time. The paper is one of the most influential papers of NeurIPS and won the Test-of-Time Award by NeuIPS in 2017.

Assume there exists a scalar random variable $w \in \mathcal{W}$ (with distribution μ) and a feature map function $\psi : \mathbb{R} \times \mathcal{W} \to \mathbb{R}$ such that

$$\Phi(x, y) = \mathbb{E}_{w \sim \mu}\left[\psi(x; w)\psi(y; w)\right]. \tag{5}$$

This assumption is also called the "reproducing property".

Example 3 (Angular)**.** For Angular kernel, we have

$$\Phi(x, y) = \underbrace{1 - \frac{2\angle(x, y)}{\pi} = \mathbb{E}_w\left[\overbrace{\operatorname{sgn}(\langle x, w\rangle)}^{\psi(x;w)}\operatorname{sgn}(\langle y, w\rangle)\right]}_{\substack{\text{“Grothendieck identity”}\\ \text{Proof by elementary geometry}}}, \tag{6}$$

where $w \sim \mu =$ uniform over unit sphere in $\mathbb{R}^d$.

For shift-invariant kernel, we have the following classical theorem:

Theorem 4 (Bochner's Theorem)**.** *A continuous shift-invariant kernel $\Phi(x, y) = \varphi(x - y)$ on $\mathbb{R}^d$ is positive definite if and only if $\varphi(\cdot)$ is the Fourier transform of a positive finite measure.*

Example 5 (Gaussian Kernel)**.** For Gaussian kernel, we have

$$\psi(x; w_1, w_2) = \sqrt{2}\cos(\langle x, w_1\rangle + w_2), \tag{7}$$

where $w_1 \sim \mathcal{N}(0, \alpha I_{d\times d})$, $w_2 \sim \mathrm{Unif}(0, 2\pi)$.

Define the random feature vector Z to be

$$Z = \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix} = \begin{bmatrix} \psi\left(x_1; \omega\right) \\ \vdots \\ \psi\left(x_N; \omega\right) \end{bmatrix} \in \mathbb{R}^N.$$

Then $R := ZZ^\top$ is a rank-1 unbiased estimate of the kernel matrix G, since $G = \mathbb{E}_w\left[ZZ^\top\right]$. To approximate the kernel matrix G by random features, we generate n independent copies of R:

$$R_1, \ldots, R_n \in \mathbb{R}^{N\times N}.$$

Then our estimator of G is given by

$$\hat{G} := \frac{1}{n}\sum_{\ell=1}^{n} R_\ell. \tag{8}$$

$\hat{G}$ is an rank-n approximation of $G \in \mathbb{R}^{N\times N}$. To generate a copy of Z, we need $O(Nd)$ operations. So in total it needs $O(nNd)$ operations to generate all n copies. A question of interest is how large n, the number of random features, needs to be to guarantee a reasonable approximation.

5 Approximation Guarantees

For the following, we focus on Angular Kernel with

$$\psi(x;w) = \operatorname{sgn}(\langle x, w\rangle).$$

We would like to bound

$$\left\|\sum_{l=1}^{n}\left[\left(\frac{1}{n}R_\ell\right) - \mathbb{E}\left(\frac{1}{n}R_\ell\right)\right]\right\|_{\mathrm{op}}$$

by matrix Bernstein inequality.

Note that

1. $\left\|\frac{1}{n}R_\ell\right\|_{\mathrm{op}} = \frac{1}{n}\|Z_\ell\|_2^2 = \frac{N}{n}$, since every entry of Z_ℓ is either 1 or -1. We have

$$\begin{aligned}\left\|\frac{1}{n}R_\ell - \mathbb{E}\left(\frac{1}{n}R_\ell\right)\right\|_{\mathrm{op}} &\le \left\|\frac{1}{n}R_\ell\right\| + \left\|\mathbb{E}\left(\frac{1}{n}R_\ell\right)\right\|,\\ &\le \left\|\frac{1}{n}R_\ell\right\| + \mathbb{E}\left\|\frac{1}{n}R_\ell\right\| \qquad \text{(Jensen)}\ ,\\ &= \frac{2N}{n} =: b.\end{aligned}$$

2. $\mathbb{E}\left[R_\ell^2\right] = \mathbb{E}\left[Z_\ell Z_\ell^\top Z_\ell Z_\ell^\top\right] = N\mathbb{E}\left[Z_\ell Z_\ell^\top\right] = NG$. It yields that

$$\begin{aligned}\left\|\frac{1}{n^2}\sum_{\ell=1}^{n}\mathbb{E}\left[(R_\ell - \mathbb{E}R_\ell)^2\right]\right\|_{\mathrm{op}} &\le \left\|\frac{1}{n^2}\sum_{\ell=1}^{n}\mathbb{E}\left[R_\ell^2\right]\right\|_{\mathrm{op}}\\ &= \frac{N}{n}\|G\|_{\mathrm{op}} =: \sigma^2.\end{aligned}$$

To prove the first inequality above, observe that

$$\begin{aligned}0 &\overset{\text{(i)}}{\preceq} \sum_{\ell=1}^{n}\mathbb{E}\left[(R_\ell - \mathbb{E}R_\ell)^2\right],\\ &= \sum_{\ell=1}^{n}\left(\mathbb{E}\left[R_\ell^2\right] - \left(\mathbb{E}\left[R_\ell\right]\right)^2\right),\\ &\overset{\text{(ii)}}{\preceq} \sum_{\ell=1}^{n}\mathbb{E}\left[R_\ell^2\right],\end{aligned}$$

where step (i) holds because the sum of the expectations of p.s.d. matrices is p.s.d, and step (ii) holds because the matrix $\left(\mathbb{E}\left[R_\ell\right]\right)^2$ is p.s.d. It follows that

$$\begin{aligned}&\left\|\frac{1}{n^2}\sum_{\ell=1}^{n}\mathbb{E}\left[(R_\ell - \mathbb{E}R_\ell)^2\right]\right\|_{\mathrm{op}},\\ =&\frac{1}{n^2}\lambda_1\left(\sum_{\ell=1}^{n}\mathbb{E}\left[(R_\ell - \mathbb{E}R_\ell)^2\right]\right),\\ \le&\frac{1}{n^2}\lambda_1\left(\sum_{\ell=1}^{n}\mathbb{E}\left[R_\ell^2\right]\right)\\ =&\left\|\frac{1}{n^2}\sum_{\ell=1}^{n}\mathbb{E}\left[R_\ell^2\right]\right\|_{\mathrm{op}}.\end{aligned}$$

Applying (the user-friendly form of) matrix Bernstein inequality, we have that w.h.p.

$$\begin{aligned}\|\hat{G}-G\|_{\mathrm{op}} &\lesssim \sqrt{\sigma^2 \log N} + b\log N,\\ &\lesssim \sqrt{\frac{N}{n}\|G\|_{\mathrm{op}}\log N} + \frac{N}{n}\log N. \qquad (*)\end{aligned}$$

To better understand the sample complexity, we introduce the definition of matrix intrinsic dimension as follows.

Definition 2 (Intrinsic dimension)**.** *Denote the intrinsic dimension of a p.s.d. matrix G as* $\operatorname{intdim}(G)$*, defined as*

$$\operatorname{intdim}(G) \coloneqq \frac{\operatorname{tr} G}{\|G\|_{\mathrm{op}}} = \frac{\sum_i \lambda_i(G)}{\lambda_1(G)} = \text{stable-rank}(G^{1/2}).$$

This can be thought of as a robust version of matrix rank.

For a kernel matrix $G \in \mathbb{R}^{N\times N}$, the diagonal entries are always 1, so

$$\operatorname{intdim}(G) = \frac{\sum_i G_{ii}}{\|G\|_{\mathrm{op}}} = \frac{N}{\|G\|_{\mathrm{op}}}.$$

For each $\varepsilon \in (0,1)$, the bound (*) implies that if $n \gtrsim \frac{\operatorname{intdim}(G)\log N}{\varepsilon^2}$, then $\frac{\|\hat{G}-G\|_{\mathrm{op}}}{\|G\|_{\mathrm{op}}} \leqslant \varepsilon + \varepsilon^2 \leqslant 2\varepsilon$. Thus, random features method gives good approximation using small n when G is (approximately) low-rank, i.e., $\operatorname{intdim}(G) \ll N$.

Remark The bound (*) controls the approximation error of the kernel. In practice, we often care about the prediction error for a new test data point. An important problem of studying random features is: what is the relation between the approximation error and the prediction error? Some experiments show that they are not necessarily proportional to each other.