Random Features for Large-Scale Kernel Machines[1]

Kailong Wang[1]

[1]Rutgers University

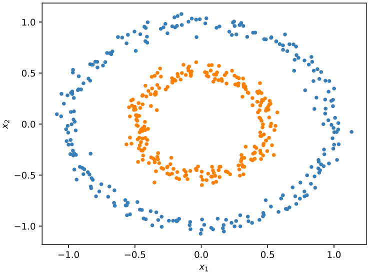
August 6, 2023

[1]Ali Rahimi and Benjamin Recht. "Random Features for Large-Scale Kernel Machines". In: 20 (2007). Ed. by J. Platt et al. URL: https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

Table of Contents

Linear Non-separable Problem

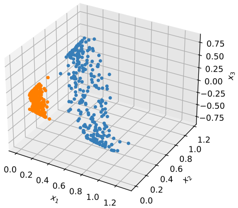
Consider a binary classification problem with non-linear samples.



e.g. For the above dataset $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}]$ where column vector $\mathbf{x_i} \in \mathbb{R}^N$, a linear decision boundary does not exist.

Lifting

One idea is **LIFTING** the samples into a high dimensional space in which the samples are linearly separable.



In this case, the function $\phi(\mathbf{X}) = \left[\mathbf{x_1} \circ \mathbf{x_1}, \mathbf{x_2} \circ \mathbf{x_2}, \sqrt{2}\mathbf{x_1} \circ \mathbf{x_2}\right]$, where $\circ$ is the Hadamard product, lifts the samples into $\mathbb{R}^3$ and the samples are linearly separable.

SVM[2]

The idea of lifting has been implemented in many classification algorithms such as *support vector machine* (SVM).

Dual Problem of SVM

$$\max_{\alpha} \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \langle \mathbf{x_n}, \mathbf{x_m} \rangle$$

Dual Problem with Lifting

$$\max_{\alpha} \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \langle \phi(\mathbf{x_n}), \phi(\mathbf{x_m}) \rangle$$

This is the hard-margin SVM. The soft-margin SVM is similar.

[2] Stephen Boyd and Lieven Vandenberghe. "Convex optimization". In: (2004).

Curse of Dimensionality–Type I

$$
\begin{aligned}
\langle \phi(\mathbf{x_n}), \phi(\mathbf{x_m}) \rangle &= \left[x_{n,1}^2, x_{n,2}^2, \sqrt{2}x_{n,1}x_{n,2}\right]\left[x_{m,1}^2, x_{m,2}^2, \sqrt{2}x_{m,1}x_{m,2}\right]^\mathsf{T} \\
&= x_{n,1}^2 x_{m,1}^2 + x_{n,2}^2 x_{m,2}^2 + 2x_{n,1}x_{n,2}x_{m,1}x_{m,2}
\end{aligned}
$$

For the given example, it does three multiplication to get the result (a constant value). For a function lifting the original vector space to a much higher dimension, such a calculation can be computationally thirsty. Alternatively, this can be done as follow, whose computational complexity only depends on the dimension of the original vector space.

$$
\begin{aligned}
(\langle \mathbf{x_n}, \mathbf{x_m} \rangle)^2 &= \left([x_{n,1}\ x_{n,2}][x_{m,1}\ x_{m,2}]^\mathsf{T}\right)^2 \\
&= (x_{n,1}x_{m,1} + x_{n,2}x_{m,2})^2 \\
&= x_{n,1}^2 x_{m,1}^2 + x_{n,2}^2 x_{m,2}^2 + 2x_{n,1}x_{n,2}x_{m,1}x_{m,2} \\
&= \langle \phi(\mathbf{x_n}), \phi(\mathbf{x_m}) \rangle
\end{aligned}
$$

Kernel Trick

The type of function, such as $(\langle\cdot,\cdot\rangle)^2$, that provides a computationally efficient way to compute the inner product in the high dimensional space is called a **Kernel Function**.

$$K(\cdot,\cdot) = \langle\phi(\cdot),\phi(\cdot)\rangle$$

The matrix that is formed by stacking the kernel function for all samples is called the **Kernel Matrix** or **Gram Matrix** $\mathbf{K}$,

$$\mathbf{K}_{nm} \equiv K(\mathbf{x_n},\mathbf{x_m}).$$

Some kernel functions can lift the original vector space to an infinite dimensional space. The algorithms involve kernel trick is called **Kernel Machines**.

Curse of Dimensionality–Type II

Another famous kernel machine is kernel ridge regression (KRR). With $\mathbf{y} \in R^N$, $\mathbf{X} \in \mathbb{R}^{N \times d}$, and $\phi_{d \to k}(\cdot) : \mathbb{R}^d \to \mathbb{R}^k$, the loss function is

$$\mathcal{L}(\mathbf{w}) = \underset{\mathbf{w}}{\text{argmin}} (\mathbf{y} - \phi(\mathbf{X})\mathbf{w})^{\mathsf{T}} (\mathbf{y} - \phi(\mathbf{X})\mathbf{w}) + \lambda \mathbf{w}^{\mathsf{T}} \mathbf{w}.$$

The normal equation of KRR is

$$\begin{aligned} \mathbf{w} &= (\phi(\mathbf{X})^{\mathsf{T}} \phi(\mathbf{X}) + \lambda \mathbf{I}_k)^{-1} \phi(\mathbf{X})^{\mathsf{T}} \mathbf{y} \\ &= (\mathbf{K} + \lambda \mathbf{I}_k)^{-1} \phi(\mathbf{X})^{\mathsf{T}} \mathbf{y}. \end{aligned}$$

Solving this problem requires $\Theta(k^3)$ time and $\Theta(k^2)$ memory.

Motivation

Can we find a way to construct the **Kernel Matrix**, which is equivalent to lift $\mathbf{X}$ to $\mathbb{R}^s$ with $d < s \ll k$, while not sacrifices model performance?

Some Prerequisites

Definition: Shift Invariant Kernel (Radial Basis Function (RBF))

A kernel function $K(\mathbf{x_n}, \mathbf{x_m})$ is called **shift invariant** if it can be written as $K(\mathbf{x_n}, \mathbf{x_m}) = k(\mathbf{x_n} - \mathbf{x_m})$ for some function $g(\cdot)$ (*e.g.* $K_{Gaussian}(\mathbf{x_n}, \mathbf{x_m}) = \exp(-\gamma\|\mathbf{x_n} - \mathbf{x_m}\|_2^2)$).

Mercer's Theorem

A continuous function $K(\mathbf{x_n}, \mathbf{x_m})$ is a valid kernel function if and only if the kernel matrix $\mathbf{K}$ is **positive semi-definite**.

Bochner's Theorem

A continuous function $k(\cdot)$ is **positive semi-definite** if and only if it is the Fourier transform of a non-negative measure.

Random Fourier Features

Conclusion

A continuous **shift invariant** kernel $K(\mathbf{x_n}, \mathbf{x_m})$, which is **positive semi-definite** (Mercer's Theorem), is the Fourier transform of a non-negative measure $p(\cdot)$.

$$\phi(\mathbf{x_n})^\mathsf{T}\phi(\mathbf{x_m}) = K(\mathbf{x_n}, \mathbf{x_m}) = k(\mathbf{x_n} - \mathbf{x_m}) \tag{1}$$

$$= \int_{\mathbb{R}^d} p(\omega) \exp(i\omega^\mathsf{T}(\mathbf{x_n} - \mathbf{x_m}))\, \mathrm{d}(\mathbf{x_n} - \mathbf{x_m}) \tag{2}$$

$$= \mathbb{E}_\omega \left[\xi_\omega(\mathbf{x_n})^\mathsf{H} \xi_\omega(\mathbf{x_m})\right] \tag{3}$$

Here $\xi_\omega(\mathbf{x_n} - \mathbf{x_m}) = \exp(i\omega^\mathsf{T}(\mathbf{x_n} - \mathbf{x_m}))$.

Random Fourier Features

Since both the $p(\cdot)$ and $k(\triangle)$ are real-valued, we can replace $\exp(i\omega^\intercal(\mathbf{x_n} - \mathbf{x_m}))$ with $\cos(\omega^\intercal(\mathbf{x_n} - \mathbf{x_m}))$. Let $z_\omega(\mathbf{x}) = \begin{bmatrix}\cos(\mathbf{x})\\ \sin(\mathbf{x})\end{bmatrix} = \sqrt{2}\cos(\omega^\intercal \mathbf{x} + b)$ where ω is drawn from $p(\omega)$ and b is uniformly drawn from $[0, 2\pi]$. Then eq. (3) becomes $\mathbb{E}_\omega[z_\omega(\mathbf{x_n})^\intercal z_\omega(\mathbf{x_m})]$.

To further reduce the variance of the estimator, we can randomly draw s samples of ω and normalize each corresponding $z_\omega(\mathbf{x})$ by $\sqrt{s}$. Then the inner product $z(\mathbf{x_n})^\intercal z(\mathbf{x_m}) = \frac{1}{s}\sum_{j=1}^{s} z_{\omega j}(\mathbf{x_n})^\intercal z_{\omega j}(\mathbf{x_m})$

Algorithm

Algorithm Random Fourier Features

Require: A shift invariant kernel $K(\mathbf{x_n}, \mathbf{x_m}) = k(\mathbf{x_n} - \mathbf{x_m})$.

Ensure: A randomized feature map $z(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^s$ so that $z(\mathbf{x_n})^\mathsf{T} z(\mathbf{x_m}) \approx K(\mathbf{x_n}, \mathbf{x_m})$.

Compute the Fourier transform $p(\cdot)$ of the kernel K : $p(\omega) = \frac{1}{2\pi} \int \exp(-i\omega^\mathsf{T} \triangle) k(\triangle) \, \mathrm{d}\triangle$

Draw s i.i.d. samples $\omega_1, \omega_2, \ldots, \omega_s \in \mathbb{R}^d$ from $p(\cdot)$ and s i.i.d. samples $b_1, b_2, \ldots, b_s \in [0, 2\pi]$.

Let $z(\mathbf{x}) \equiv \sqrt{\frac{2}{s}}[\cos(\omega_1{}^\mathsf{T}\mathbf{x} + b_1), \cos(\omega_2{}^\mathsf{T}\mathbf{x} + b_2), \ldots, \cos(\omega_s{}^\mathsf{T}\mathbf{x} + b_s)]^\mathsf{T}$

Common RFF

Kernel	$K(\triangle)$	$p(\omega)$
Gaussian	$\exp(-\gamma\|\triangle\|_2^2)$	$(2\pi)^{-\frac{s}{2}}\exp -\gamma\|\omega\|_2^2$
Laplacian	$\exp(-\|\triangle\|_1)$	$\prod_d(\pi(1+\omega_d^2))^{-1}$
Cauchy	$\prod_d 2(1+\triangle_d^2)^{-1}$	$\exp(-\|\omega\|_1)$

Convergence with Hoeffding's Inequality[3]

Hoeffding's Inequality

Let $X_1, X_2, \cdots, X_N$ be independent random variables. Assume that $X_i \in [m_i, M_i]$ for every i. Then, for any $\epsilon > 0$, we have

$$\mathbb{P}\left(\left|\sum_{i=i}^{N}(X_i - \mathbb{E}[X_i])\right| \geq \epsilon\right) \leq 2\exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^{N}(M_i - m_i)^2}\right).$$

Bound for *any* pair of samples $\mathbf{x_n}$ and $\mathbf{x_m}$

Given z_ω is bounded random variable between $[-\sqrt{2/s}, \sqrt{2/s}]$, with Hoeffding's Inequality, we have

$$\mathbb{P}(|z(\mathbf{x_n})^\mathsf{T} z(\mathbf{x_m}) - K(\mathbf{x_n}, \mathbf{x_m})| \geq \epsilon) \leq 2\exp\left(-\frac{s\epsilon^2}{4}\right).$$

[3] Roman Vershynin. "High-Dimensional Probability: An Introduction with Applications in Data Science". In: (2018).

Convergence

Bound for *all* pair of samples $\mathbf{x_n}$ and $\mathbf{x_m}$

Let $\mathcal{M}$ be a compact subset of $\mathbb{R}^d$ with diameter diam$(\mathcal{M})$. Then, for the mapping z defined in Algorithm 1, we have

$$\mathbb{P}\left(\sup_{x,y\in\mathcal{M}} |z(\mathbf{x_n})^\mathsf{T} z(\mathbf{x_m}) - K(\mathbf{x_n},\mathbf{x_m})| \geq \epsilon\right)$$
$$\leq 2^8\left(\frac{\sigma_{p(\cdot)}\text{diam}(\mathcal{M})}{\epsilon}\right)^2 \exp\left(-\frac{s\epsilon^2}{4(d+2)}\right).$$

The $\sigma^2_{p(\cdot)} = \mathbb{E}_{p(\cdot)}\left[\omega^\mathsf{T}\omega\right]$ is the second moment of the Fourier transform of the $K(\cdot,\cdot)$.

The proof of this bound uses the knowledge of ϵ-net and ϵ-covering number.