Random Features For Large-Scale Kernel Machines

Blanca Cano Camarero

Universidad Autónoma de Madrid

March 22, 2023

Overview

Random Features for Large-Scale Kernel Machines

- Ali Rahimi and Recht
- Year 2007
- Cited by 3784.

Rahimi and Recht (2007)

Abstract To accelerate the training of kernel machines, we propose to **map the input data to a randomized low-dimensional feature space and then apply existing fast linear methods**. The features are designed so that the inner products of the transformed data are approximately equal to those in the feature space of a user specified shift-invariant kernel. We explore two sets of random features, provide convergence bounds on their ability to approximate various radial basis kernels, and show that in large-scale classification and regression tasks linear machine learning algorithms applied to these features outperform state-of-the-art large-scale kernel machines.

Introduction

Kernel support vector machine are universal approximators.

- This result was first proved by Vladimir Vapnik and Alexey Chervonenkis in their paper "On the uniform convergence of relative frequencies of events to their probabilities" published in 1971. Vapnik and Chervonenkis (1971).
- The universality of kernel SVMs was later proved by Bernhard Schölkopf and Alexander J. Smola in their influential book "Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond" published in 2002. In chapter 8 of the book, they prove that under certain conditions, a kernel **SVM can approximate any continuous function to arbitrary accuracy**, making it a universal approximator. Schölkopf and Smola (2002)

On the Equivalence between Neural Network and Support Vector Machine

Chen et al. (2021) We prove the equivalence of infinitely wide neural network with support vector machine and other kinds of ℓ_2 regularized kernel machines.

Abstract: Recent research shows that the dynamics of an infinitely wide neural network (NN) trained by gradient descent can be characterized by Neural Tangent Kernel (NTK). Under the squared loss, the infinite-width NN trained by gradient descent with an infinitely small learning rate is equivalent to kernel regression with NTK. However, the equivalence is only known for ridge regression currently , while the equivalence between NN and other kernel machines (KMs), e.g. support vector machine (SVM), remains unknown. Therefore, in this work, we propose to establish the equivalence between NN and SVM, and specifically, the infinitely wide NN trained by soft margin loss and the standard soft margin SVM with NTK trained by subgradient descent. Our main theoretical results include establishing the equivalence between NN and a broad family of ℓ_2 regularized KMs with finite-width bounds, which cannot be handled by prior work, and showing that every finite-width NN trained by such regularized loss functions is approximately a KM. Furthermore, we demonstrate our theory can enable three practical applications, including (i) *non-vacuous* generalization bound of NN via the corresponding KM; (ii) *nontrivial* robustness certificate for the infinite-width NN (while existing robustness verification methods would provide vacuous bounds); (iii) intrinsically more robust infinite-width NNs than those from previous kernel regression.

Kernel matrix scale poorly

If we have a set of training samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ and a kernel function $k(\cdot, \cdot)$, then the Gram matrix $\mathbf{K}$ is defined as:

$$K_{i,j} = k(x_i, x_j), \tag{1}$$

where the (i, j)-th element of K a $n \times n$ symmetric positive semi-definite matrix. Given any positive definite function $k(\mathbf{x}, \mathbf{y})$ with $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, there exists an inner product and a lifting ϕ such that the inner product between lifted data points can be quickly computed as $\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = k(\mathbf{x}, \mathbf{y})$.

Fundamental theorem of Galois theory

Wikipedia (2023) The theorem that guarantees the existence of a lifting in the kernel trick is the Fundamental theorem of Galois theory. This theorem states that given a finite group G, a field K, and a Galois extension L of K with Galois group $Gal(L/K) \cong G$, for every subgroup $H \subseteq G$, there exists a Galois extension M of K such that $Gal(M/K) \cong H$, and furthermore, M is an extension of L.
In the context of the kernel trick, this means that if we want to solve a system of equations using the kernel trick, and the system of equations can be expressed as the kernel of a group homomorphism $\phi : G \to H$, then we can find a solution to the system of equations in the subgroup $Ker(\phi)$ of G if we find a Galois extension L of K such that $Gal(L/K) \cong G$ and L contains all the roots of the polynomial that defines H over K. The Galois lifting theorem guarantees that such a Galois extension L exists, which means we can solve the system of equations using the kernel trick.

Kernel Trick

The kernel trick is a technique used in machine learning and kernel methods to implicitly map the input data into a higher-dimensional feature space without actually computing the mapping explicitly.
Instead of computing the mapping explicitly, we can define a kernel function $k(\cdot, \cdot)$ that computes the inner product between the feature vectors in the higher-dimensional space.
Specifically, given two input points $\mathbf{x}_i$ and $\mathbf{x}_j$, we can define the kernel function as:

$$k(x_i, x_j) = <\phi(x_i), \phi(x_j)> \tag{2}$$

Article improvement

- Given a set of input data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ in a low-dimensional space, a lifting $\phi : \mathbb{R}^d \to \mathbb{R}^K$ maps each input point $\mathbf{x}_i$ to a higher-dimensional feature vector $\phi(\mathbf{x}_i) \in \mathbb{R}^K$. The lifted feature vectors can then be used as input to a learning algorithm that operates in the higher-dimensional space.
- they propose explicitly mapping the data to a low-dimensional Euclidean inner product space using a randomized feature map $z : \mathbb{R}^d \to \mathbb{R}^D$ so that the inner product between a pair of transformed points approximates their kernel evaluation:

$$k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle \approx z(\mathbf{x})' z(\mathbf{y}). \tag{3}$$

$$D << K \tag{4}$$

In what follows, they show how to construct feature spaces that uniformly approximate popular shift-invariant kernels $k(\mathbf{x} - \mathbf{y})$ to within ϵ with only $D = O(d\epsilon^{-2} \log^{1/\epsilon})$.

Optimized

With the kernel trick, evaluating the machine at a test point $\mathbf{x}$ requires computing

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i k(\mathbf{x}_i, \mathbf{x}) \tag{5}$$

, which requires $O(Nd)$ operations to compute and requires retaining much of the dataset unless the machine is very sparse.

This is often unacceptable for large datasets. On the other hand, after learning a hyperplane $\mathbf{w}$, a linear machine can be evaluated by simply computing

$$f(\mathbf{x}) = \mathbf{w}'\mathbf{z}(\mathbf{x}), \tag{6}$$

which, with the randomized feature maps presented here, requires only $O(D + d)$ operations and storage.

Random Fourier Features introduction

Let be $x \in \mathbb{R}^d$ (a column vector), the first set of random features consists of random Fourier bases

$$\cos(w^T x + b) \tag{7}$$

where $w \in \mathbb{R}^d$ and b are random variables.
See that $T(x) = w^t x + b$ is affine transformation $T : \mathbb{R}^d \longrightarrow \mathbb{R}$ and then cos function maps $\cos : \mathbb{R} \longrightarrow S^1$.

Algorithm: Random Fourier Features

Input: K a positive definite shift-invariant kernel $k(x, y) = k(x - y)$.
Output: A randomized feature map $z(x) : \mathbb{R}^d \longrightarrow \mathbb{R}^D$ so that $z(x)^T z(y) \approx k(x - y)$
Compute the Fourier transform p of the kernel k:

$$p(w) = \frac{1}{2\pi} \int e^{-jw^T \delta} k(\delta) d\Delta. \tag{8}$$

Draw D iid samples $\{w_1, \ldots, w_D\} \subset \mathbb{R}^d$ from p and D iid samples $b_1, \ldots, b_D \in \mathbb{R}$ from the uniform distribution on $[0, 2\pi]$.
Let

$$z(x) \equiv \sqrt{\frac{2}{D}} \left[\cos(w_1^T x + b_1) \ldots \cos(w_D^T x + b_D)\right]^T. \tag{9}$$

Proof I

Our objective is to proof that $z(x)^T z(y)$ is close to $k(x-y)$. Mathematically, we want to garante the uniform convergence of the Fourier Features.

Theorem (Bochner's theorem)

A continuos kernel $k(x,y) = k(x-y)$ on $\mathbb{R}^d$ is positive if and only if $k(\delta)$ is the Fourier transforme of a non-negative measure.

Proof II

If a shift-invariant kernel $k(\delta)$ is properly scaled, Bochner's theorem guarantees that its Fourier transform $p(w)$ is a proper probability distribution.
Defining $\zeta_w(x) = e^{jw^t x}$, we have

$$k(x-y) = \int_{\mathbb{R}^d} p(w) e^{jw^t(x-y)} dw = E_w\left[\zeta_w(x)\zeta_w(y)^*\right], \tag{10}$$

where $\zeta_w(y)^* = e^{-jw^t x}$ is the conjugate. We have proof in (10) that $\zeta_w(x)\zeta_w(y)^*$ is a unbiased estimate of $k(x, y)$ when w is drawn from p.

Proof III

Defining

$$z_w(x) = \sqrt{2}\cos\left(w^T x + b\right), \tag{11}$$

where w is drawn from a $p(w)$ and b is drawn uniformly from $[0, 2\pi]$ we obtain that
Now we are going to proof that:

$$E\left[z_w(x)z_w(y)\right] = k(x, y). \tag{12}$$

Proof

Secondly, as a consequence of the sum of angles:

$$
\begin{aligned}
z_w(x)z_w(y) &= 2\cos\left(w^T x + b\right)\cos\left(w^T y + b\right) \\
&= \left(\cos\left(w^T x + b\right)\cos\left(w^T y + b\right) + \sin\left(w^T x + b\right)\sin\left(w^T y + b\right)\right) \\
&\quad + \left(\cos\left(w^T x + b\right)\cos\left(w^T y + b\right) - \sin\left(w^T x + b\right)\sin\left(w^T y + b\right)\right) \\
&= \cos\left(w^T(x - y)\right) + \cos\left(w^T(x + y) + 2b\right). \qquad (13)
\end{aligned}
$$

$$E\left[\cos\left(w^T(x-y)\right)\right] = \frac{1}{2}\left(E\left[e^{jw^T(x-y)}\right] + E\left[e^{jw^T(y-x)}\right]\right) = k(x,y), \tag{14}$$

since k is symmetric and shift invariant and (10).

$$E\left[\cos\left(w^T(x-y)\right)\right] = \frac{1}{2}\left(E\left[e^{jw^T(x-y)}\right] + E\left[e^{jw^T(y-x)}\right]\right) = k(x,y), \tag{14}$$

since k is symmetric and shift invariant and (10).
Finally, as a result of Euler formula and (16)

$$E\left[\cos\left(w^T(x+y)+2b\right)\right] = 0, \tag{15}$$

Proof

For $s \in \{1, -1\}$ notice that using chain rule and $p(w)$ is a probability function and therefore $\int_{\mathbb{R}^d} p(w)dw = 1$.

$$\begin{aligned} E\left[e^{sjw^T(x+y)+s2b}\right] &= \int_{\mathbb{R}^d} e^{sjw^T(x+y)+s2b} p(w)dw \\ &= e^{s2b} \int_{\mathbb{R}^d} e^{sjw^T(x+y)} p(w)dw \\ &= e^{s2b} \left\{ e^{sjw^T(x+y)} - \int j(x+y)e^{sjw(x+y)}dw \right\}_{\mathbb{R}^d} \\ &= e^{s2b} \left\{ e^{sjw^T(x+y)} - e^{sjw(x+y)} \right\} = 0. \end{aligned} \tag{16}$$

First bound

We can lower the variance of the estimate of the kernel by concatenating D randomly chosen z_w into one $D-\textit{dimensional}$ vector and normalizing each component by $\sqrt{2}$. The inner product

$$z(x)^T z(y) = \frac{1}{D} \sum_{j=1}^{D} z_{w_j}(x) z_{w_j}(y) \tag{17}$$

is a sample average of z_w and is therefore a lower variance approximation to the expectation.

Theorem

Hoeffding's inequality Let $X_1, \ldots, X_n$ be independent random variables such that $a_i \leq X_i \leq b_i$ almost surely. Consider $S_n = X_1 + \ldots + X_n$.
The Hoeffding's theorem states that, for all $t > 0$,

$$P\left(|S_n - E[S_n]| \geq t\right) \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right) \tag{18}$$

(See the proof at Hoeffding (1994)).
Using Hoeffding's inequality

$$P\left(|z(x)^T z(y) - k(x,y)| > \varepsilon\right) \leq 2\exp\left[-\frac{2\varepsilon^2}{(4/\sqrt{D})^2}\right] \leq 2\exp\left[-\frac{D\varepsilon^2}{8}\right] \tag{19}$$

Theorem

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

$$P\left[\sup_{x,y\in M} |z(x)^T z(y) - k(y,x)| \geq \varepsilon\right] \leq 2^8 \left(\frac{\sigma_p \operatorname{diam}(M)}{\varepsilon}\right) \exp\left(-\frac{D\varepsilon^2}{4(d+2)},\right) \quad (20)$$

where $\sigma_p^2 \equiv \mathbb{E}_{p(\omega)}[\omega'\omega]$ is the second moment of the Fourier transform of k. Further,

$$\sup_{x,y\in M} |z(x)^T z(y) - k(y,x)| \geq \varepsilon$$

with any constant probability when $D = \Omega\left(\frac{d}{\epsilon^2} \log \frac{\sigma_p diam M}{\epsilon}\right)$.

Define $s(x,y) \equiv z(x)^T z(y)$, and $f(x,y) \equiv s(x,y) - k(y,x)$, and for a bigger enough D in the first inequality (19) and by construction we would have $|f(x,y)| \leq 2$ and $E[f(x,y)] = 0$.
Let define

$$M_\Delta = \{x - y : x, y \in M\}. \tag{21}$$

Since M is compact, M_Δ is also compact. Moreover, by the triangle inequality, M_Δ has diameter at most twice $\text{diam}(M)$. Since M_Δ is compact, we can construct an ϵ-net that covers M_Δ using at most $T = (4\,\text{diam}(M)/r)^d$ balls of radius r.
Let $\{\Delta_i\}i = 1^T$ denote the centers of these balls, and let L_f be the Lipschitz constant of f. If we ensure that $|f(\Delta_i)| < \epsilon/2$ for all i and $L_f < \epsilon$, then we can guarantee that $|f(\Delta_i)| < \epsilon$ for all $\Delta \in M_\Delta$ by using triangle inequality, Lipschitz definition and all the hypothesis:

$$
\begin{aligned}
|f(\Delta)| &= |f(\Delta) \pm f(\Delta_i)| && (22)\\
&\leq L_f\,|\Delta - \Delta_i| && (23)\\
&\leq L_f r + \frac{\varepsilon}{2} = \varepsilon. && (24)
\end{aligned}
$$

Since f differentiable, $L_f = \|\nabla f(\Delta^*)\|$, where $\Delta^* = \arg\max_{\Delta \in M_\Delta} \|\nabla f(\Delta)\|$.
By variance expansion in expectations and s gradient,

$$E[\nabla s(\Delta)] = \nabla k(\Delta), \tag{25}$$

so

$$\begin{aligned} E[L_f^2] &= E\left[\|\nabla s(\Delta*) - \nabla k(\Delta^*)\|^2\right] = && (26)\\ &= E\left[\|\nabla s(\Delta*)\|^2\right] - E\left[\|\nabla k(\Delta*)\|\right]^2 && (27)\\ &\leq E\left[\|\nabla s(\Delta*)\|^2\right] && (28)\\ &= E\left[w^2 \sin(2\Delta)\right] && (29)\\ &\leq E\left[\|w\|^2\right] = \sigma_p^2. && (30) \end{aligned}$$

By Markov's inequality,

$$P\left[L_f^2 \geq t\right] \leq \frac{E[L_f^2]}{t}, \tag{31}$$

so

$$P\left[L_f \geq \frac{\epsilon}{2r}\right] \leq \left(\frac{2r\sigma_p}{\epsilon}\right)^2. \tag{32}$$

The onion bound followed by Hoeffding's inequality applied to the anchors in the $\epsilon-$net gives

$$P\left[\cup_{i=1}^{T}\|f(\Delta_i)\| \geq \epsilon/2\right] \leq 2T\exp\left(-D^2/8\right). \tag{33}$$

Combining previous inequalities in term of the free variable r:

$$
\begin{aligned}
P\left[\sup_{\Delta\in M_\Delta}|f(\Delta)|\le\epsilon\right] &= P\left[\cup_{i=1}^{T}\|f(\Delta_i)\|\le\epsilon/2\wedge L_f\le\frac{\epsilon}{2r}\right] && (34)\\
&= 1-P\left[\cup_{i=1}^{T}\|f(\Delta_i)\|\ge\epsilon/2\vee L_f\ge\frac{\epsilon}{2r}\right] && (35)\\
&= 1-P\left[\cup_{i=1}^{T}\|f(\Delta_i)\|\ge\epsilon/2\right]-P\left[L_f\ge\frac{\epsilon}{2r}\right] && (36)\\
&\ge 1-2\left(\frac{4diam(M)}{r}\right)^{d}\exp(-D\epsilon^2/8)-\left(\frac{2r\sigma_p}{\epsilon}\right)^{2}. && (37)
\end{aligned}
$$

This has the form $1 - \kappa_1 r^{-d} - \kappa_2 r^2$. Setting $r = \left(\frac{k_1}{k_2}\right) \frac{1}{d+2}$ turns this to

$$1 - 2k_2^{\frac{d}{d+2}} k_1^{\frac{d}{d+2}}, \tag{38}$$

and assuming that $\frac{\sigma_p \operatorname{diam}(M)}{\epsilon} \geq 1$, proves the first part of the claim. To prove the second part of the claim, pick any probability for the right hand side and solve for D.

Random Binning features

Objective:

$$k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle \approx z(\mathbf{x})' z(\mathbf{y}). \tag{39}$$

Algorithm description

- Partition the input space using randomly shifted grids at randomly chosen resolutions.
- Assigns to an input point a binary bit string that corresponds to the bins in which it falls.
- This mapping is well-suited for kernels that depend only on the L_1 distance between pairs of points.
- The probability of two points of being in the same bin is proportional to $k(x, y)$.
- Finally $z(x)$ is a binary encoding of the bin where x falls.

Random Binning features: Graphical explanation

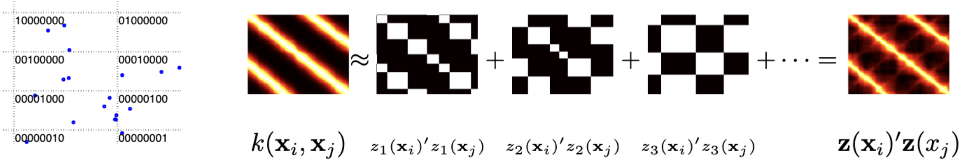

Figure 2: Random Binning Features. (left) The algorithm repeatedly partitions the input space using a randomly shifted grid at a randomly chosen resolution and assigns to each point $\mathbf{x}$ the bit string $z(\mathbf{x})$ associated with the bin to which it is assigned. (right) The binary adjacency matrix that describes this partitioning has $z(\mathbf{x}_i)' z(\mathbf{x}_j)$ in its ijth entry and is an unbiased estimate of kernel matrix.

$$z(x) = \sqrt{\frac{1}{P}} \left[z_1(x), \cdots, z_p(x) \right]^T \tag{40}$$

Sklearnt aproximation: KBinsDiscretizer

```
class sklearn.preprocessing.KBinsDiscretizer(
    n_bins=5, *,
    encode='onehot',
    strategy='quantile',
    dtype=None,
    subsample='warn',
    random_state=None)
```

Parameters:[1]

- **n_bins**: The number of bins to produce.
- **encode**: {'onehot', 'onehot-dense', 'ordinal'}.
- **strategy** {'uniform', 'quantile', 'kmeans'}, default='quantile' Strategy used to define the widths of the bins.

[1] Source: `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.KBinsDiscretizer.html`

How to use Random Binning Features

```
>>> from sklearn.preprocessing import KBinsDiscretizer
>>> X = [[0,100],[0,1],[1,0],[1,1],
[2,2],[-1,1],[-1,1]]
>>> est = KBinsDiscretizer(n_bins=3, encode='ordinal',
    strategy='uniform')
>>> est.fit(X)
KBinsDiscretizer(...)
>>> Xt = est.transform(X)
>>> Xt
array([[1., 2.],
[1., 0.],
[2., 0.],
[2., 0.],
[2., 0.],
[0., 0.],
[0., 0.]])
```

Sklearnt's KBinsDiscretizer is not the same that Random Binning Features

- Deterministic.
- Feature by feature (do no transform the data).

Table 1: Input parameters for KBinsDiscretizer and the Random Binning Features algorithm

KBinsDiscretizer	**Random Binning Features algorithm**
n_bins	The output size P
encode	A kernel function
strategy	

Related works

Nimit Kalra (2018) https://github.com/qw3rtman/random-feature-maps

- Fast Random Kernelized Features: Support Vector Machine Classification for High-Dimensional IDC Dataset (2018),
- First K-means,
- then random Features
- finally linear SVM classification.

Random Binning Features algorithm

Input

- A kernel function $k(x, y) = k(x - y) = \prod_{m=1}^{d} k_m(|x^m - y^m|)$, so that $p_m(h) \equiv h k_m''(h)$ is a probability distribution on $h \geq 0$.

Algorithm For $p \in \{1, \ldots, P\}$

1. Draw grid parameters $h, u \in \mathbb{R}^d$ with the pitch $h^m \sim p_m$, and shift u^m from the uniform distribution on $[0, h^m]$.
2. Let z return the coordinate of the bin containing x as a binary indicator vector

$$z_p(x) \equiv \text{hash}\left(\left\lfloor \frac{x^1 - u^1}{h^1} \right\rfloor, \ldots, \left\lfloor \frac{x^d - u^d}{h^d} \right\rfloor \right).$$

Return: A randomized feature map $z(x)$ so that $z(x)^T z(y) \approx k(x - y)$.

kernel restrictions and how to compute p

Lemma

Suppose a function $k(h) : \mathbb{R} \to \mathbb{R}$ is twice differentiable and has the form

$$k(x) = \int_{\mathbb{R}} p(h) \max\left(0, 1 - \frac{x}{h}\right) dh. \tag{41}$$

Then $p(h) = hk''(h)$.

kernel restrictions and how to compute p

Lemma

Suppose a function $k(h) : \mathbb{R} \to \mathbb{R}$ is twice differentiable and has the form

$$k(x) = \int_{\mathbb{R}} p(h) \max\left(0, 1 - \frac{x}{h}\right) dh. \tag{41}$$

Then $p(h) = hk''(h)$.

Proof

$$\begin{aligned} k(x) &= \int_{\mathbb{R}} p(x) \max\left(0, 1 - \frac{x}{h}\right) dh && (42)\\ &= \int_0^x p(x) 0 dx + \int_x^\infty p(x)\left(1 - \frac{x}{h}\right) dx && (43)\\ &= \int_x^\infty p(h) dh - \int_x^\infty \frac{p(h)x}{h} dh. && (44) \end{aligned}$$

Proof

The Leibniz rule for derivatives formula:

$$\frac{d}{dx}\int_{a(x)}^{b(x)} f(x,t)dt = f(x,b(x))\frac{d}{dx}b(x) - f(x,a(x))\frac{d}{dx}a(x) + \int_{a(x)}^{b(x)} \frac{\partial}{\partial x} f(x,t)dt \quad (45)$$

Hence

$$k'(x) = -p(x) - \left[\int_x^{\infty} \frac{p(x)}{x}dx - x\frac{p(x)}{x}\right] = -\int_x^{\infty} \frac{p(x)}{x}dx. \quad (46)$$

Applying a the Fundamental theorem of calculus:

$$k''(x) = \frac{P(x)}{x}. \quad (47)$$

K restriction

- Twice differentiable,
- convex.

Math formulation

$$k_{hat}(x, y; h) = \max\left(0, 1 - \frac{|x - y|}{h}\right) \tag{48}$$
$$= P\left[z(x)^T z(y) = 1|h\right] \tag{49}$$
$$= E\left[z(x)^T z(y) = 1|h\right] \tag{50}$$

By uniform $u \in U([0, h])$

Claim

Theorem

Let M be a compact subset of $\mathbb{R}^d$ with diameter $diam(M)$. Let $\alpha = \mathbb{E}[1/delta]$ and let L_k denote the Lipschitz constant of k with respect to the L_1 norm. With z as above, we have:

$$P\left[\sup_{x,y\in M} |z(x)^T z(y) - k(y,x)| \geq \varepsilon\right] \geq 1 - 36dP\alpha\ diam(M) \exp\left(\frac{-\left(\frac{P\epsilon^2}{8} + \ln\frac{\epsilon}{L_k}\right)}{d+1}\right) \tag{51}$$

Next week

1. "Nystroem Method vs Random Fourier Features: A Theoretical and Empirical Comparison", Advances in Neural Information Processing Systems 2012
2. Random features for kernel approximation: A survey on algorithms, theory, and beyond
3. Williams, C.K.I. and Seeger, M. "Using the Nystroem method to speed up kernel machines", Advances in neural information processing systems 2001 T. Yang, Y. Li, M. Mahdavi, R. Jin and Z. Zhou
4. `https://proceedings.neurips.cc/paper_files/paper/2008/file/0efe32849d230d7f53049ddc4a4b0c60-Paper.pdf`
5. Randomness in neural networks: an overview
6. Fast and scalable polynomial kernels via explicit feature maps
7. On the error of random Fourier features
8. A survey on large-scale machine learning
9. Sharp analysis of low-rank kernel matrix approximations

¡title¿

Para guardar código: https://scikit-learn.org/stable/model_persistence.html
Para sacar los modelos:
https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

References I

Yilan Chen, Wei Huang, Lam M. Nguyen, and Tsui-Wei Weng. On the equivalence between neural network and support vector machine. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=npUxA--_nyX.

Wassily Hoeffding. *Probability Inequalities for sums of Bounded Random Variables*, pages 409–426. Springer New York, New York, NY, 1994. ISBN 978-1-4612-0865-5. doi: 10.1007/978-1-4612-0865-5_26. URL https://doi.org/10.1007/978-1-4612-0865-5_26.

Tianshu Huang Nimit Kalra. Fast random kernelized features: Support vector machine classification for high-dimensional idc dataset. 2018.

References II

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Vladimir Vapnik and Alexey Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

Wikipedia. Fundamental theorem of galois theory — Wikipedia, the free encyclopedia, 2023. URL https://en.wikipedia.org/wiki/Fundamental_theorem_of_Galois_theory. [Online; accessed 22-March-2023].