Title: Random feature

URL: https://en.wikipedia.org/wiki/Random_feature

PageID: 77992820

Categories: Category:Kernel methods for machine learning, Category:Machine learning, Category:Monte Carlo methods

---

Random features (RF) are a technique used in machine learning to approximate kernel methods , introduced by Ali Rahimi and Ben Recht in their 2007 paper " Random Features for Large-Scale Kernel Machines ", [1] and extended by. [2] [3] RF uses a Monte Carlo approximation to kernel functions by randomly sampled feature maps . It is used for datasets that are too large for traditional kernel methods like support vector machine , kernel ridge regression , and gaussian process .

Mathematics

Kernel method

Given a feature map φ : R d → V {\textstyle \phi :\mathbb {R} ^{d}\to V} , where V {\textstyle V} is a Hilbert space (more specifically, a reproducing kernel Hilbert space), the kernel trick replaces inner products in feature space ■ φ (x i) , φ (x j) ■ V {\displaystyle \langle \phi (x_{i}),\phi (x_{j})\rangle _{V}} by a kernel function k (x i , x j) : R d × R d → R {\displaystyle k(x_{i},x_{j}):\mathbb {R} ^{d}\times \mathbb {R} ^{d}\to \mathbb {R} } Kernel methods replaces linear operations in high-dimensional space by operations on the kernel matrix: K X := [k (x i , x j)] i , j ∈ 1 : N {\displaystyle K_{X}:=[k(x_{i},x_{j})]_{i,j\in 1:N}} where N {\textstyle N} is the number of data points.

Random kernel method

The problem with kernel methods is that the kernel matrix K X {\textstyle K_{X}} has size N × N {\textstyle N\times N} . This becomes computationally infeasible when N {\textstyle N} reaches the order of a million. The random kernel method replaces the kernel function k {\textstyle k} by an inner product in low-dimensional feature space R D {\textstyle \mathbb {R} ^{D}} : k (x , y) ≈ ■ z (x) , z (y) ■ {\displaystyle k(x,y)\approx \langle z(x),z(y)\rangle } where z {\textstyle z} is a randomly sampled feature map z : R d → R D {\textstyle z:\mathbb {R} ^{d}\to \mathbb {R} ^{D}} .

This converts kernel linear regression into linear regression in feature space, kernel SVM into SVM in feature space, etc. Since we have K X ≈ Z X T Z X {\displaystyle K_{X}\approx Z_{X}^{T}Z_{X}} where Z X = [z (x 1) , … , z (x N)] {\displaystyle Z_{X}=[z(x_{1}),\dots ,z(x_{N})]} , these methods no longer involve matrices of size O (N 2) {\textstyle O(N^{2})} , but only random feature matrices of size O (D N) {\textstyle O(DN)} .

Random Fourier feature

Radial basis function kernel

The radial basis function (RBF) kernel on two samples x i , x j ∈ R d {\displaystyle x_{i},x_{j}\in \mathbb {R} ^{d}} is defined as [4]

where ■ x i – x j ■ 2 {\displaystyle \|x_{i}-x_{j}\|^{2}} is the squared Euclidean distance and σ {\displaystyle \sigma } is a free parameter defining the shape of the kernel. It can be approximated by a random Fourier feature map z : R d → R 2 D {\displaystyle z:\mathbb {R} ^{d}\to \mathbb {R} ^{2D}} : z (x) := 1 D [cos ■ ■ ω 1 , x ■ , sin ■ ■ ω 1 , x ■ , … , cos ■ ■ ω D , x ■ , sin ■ ■ ω D , x ■] T {\displaystyle z(x):={\frac {1}{\sqrt {D}}}[\cos \langle \omega _{1},x\rangle ,\sin \langle \omega _{1},x\rangle ,\ldots ,\cos \langle \omega _{D},x\rangle ,\sin \langle \omega _{D},x\rangle]^{T}} where ω 1 , . . . , ω D {\displaystyle \omega _{1},...,\omega _{D}} are IID samples from the multidimensional normal distribution N (0 , σ – 2 I) {\displaystyle N(0,\sigma ^{-2}I)} .

Theorem — -

(Unbiased estimation) E ■ [■ z (x) , z (y) ■] = e ■ x – y ■ 2 / (2 σ 2) . {\displaystyle \operatorname {E} [\langle z(x),z(y)\rangle]=e^{\|x-y\|^{2}/(2\sigma ^{2})}.}

(Variance bound) Var ■ [■ z (x) , z (y) ■] = O (D – 1) {\displaystyle \operatorname {Var} [\langle z(x),z(y)\rangle]=O(D^{-1})}

(Convergence) As D → ∞ {\displaystyle D\to \infty } , the approximation converges in probability to the true kernel.

(Unbiased estimation) By independence of ω 1 , . . . , ω D {\displaystyle \omega _{1},...,\omega _{D}} , it suffices to prove the case of D = 1 {\displaystyle D=1} . By the trigonometric identity cos ■ (a – b) = cos ■ (a) cos ■ (b) + sin ■ (a) sin ■ (b) {\displaystyle \cos(a-b)=\cos(a)\cos(b)+\sin(a)\sin(b)} , ■ z (x) , z (y) ■ = 1 D ∑ i = 1 D cos ■ ■ ω i , x – y ■ {\displaystyle \langle z(x),z(y)\rangle ={\frac {1}{D}}\sum _{i=1}^{D}\cos \langle \omega _{i},x-y\rangle } Apply the spherical symmetry of normal distribution, then evaluate the integral: ∫ – ∞ ∞ cos ■ (k x) e – x 2 / 2 2 π d x = e – k 2 / 2 . {\displaystyle \int _{-\infty }^{\infty }{\frac {\cos(kx)e^{-x^{2}/2}}{\sqrt {2\pi }}}dx=e^{-k^{2}/2}.}

(Variance bound) Since ω 1 , . . . , ω D {\displaystyle \omega _{1},...,\omega _{D}} are IID, it suffices to prove that the variance of cos ■ ■ ω 1 , x – y ■ {\displaystyle \cos \langle \omega _{1},x-y\rangle } is finite, which is true since it is bounded within [– 1 , + 1] {\displaystyle [-1,+1]} .

(Convergence) By Chebyshev's inequality .

Since cos , sin {\displaystyle \cos ,\sin } are bounded, there is a stronger convergence guarantee by Hoeffding's inequality . [1] : Claim 1

Random Fourier features

By Bochner's theorem , the above construction can be generalized to arbitrary positive definite shift-invariant kernel k (x , y) = k (x – y) {\displaystyle k(x,y)=k(x-y)} .

Define its Fourier transform p (ω) = 1 2 π ∫ R d e – j ■ ω , Δ ■ k (Δ) d Δ {\displaystyle p(\omega)={\frac {1}{2\pi }}\int _{\mathbb {R} ^{d}}e^{-j\langle \omega ,\Delta \rangle }k(\Delta)d\Delta } then ω 1 , . . . , ω D {\displaystyle \omega _{1},...,\omega _{D}} are sampled IID from the probability distribution with probability density p {\displaystyle p} . This applies for other kernels like the Laplace kernel and the Cauchy kernel.

Neural network interpretation

Given a random Fourier feature map z {\displaystyle z} , training the feature on a dataset by featurized linear regression is equivalent to fitting complex parameters θ 1 , … , θ D ∈ C {\displaystyle \theta _{1},\dots ,\theta _{D}\in \mathbb {C} } such that f θ (x) = R e (∑ k θ k e i ■ ω k , x ■) {\displaystyle f_{\theta }(x)=\mathrm {Re} \left(\sum _{k}\theta _{k}e^{i\langle \omega _{k},x\rangle }\right)} which is a neural network with a single hidden layer, with activation function t ■ e i t {\displaystyle t\mapsto e^{it}} , zero bias, and the parameters in the first layer frozen.

In the overparameterized case, when 2 D ≥ N {\displaystyle 2D\geq N} , the network linearly interpolates the dataset { (x i , y i) } i ∈ 1 : N {\displaystyle \{(x_{i},y_{i})\}_{i\in 1:N}} , and the network parameters is the least-norm solution: θ ^ = arg ■ min θ ∈ C D , f θ (x k) = y k ∀ k ∈ 1 : N ■ θ ■ {\displaystyle {\hat {\theta }}=\arg \min _{\theta \in \mathbb {C} ^{D},f_{\theta }(x_{k})=y_{k}\forall k\in 1:N}\|\theta \|} At the limit of D → ∞ {\displaystyle D\to \infty } , the L2 norm ■ θ ^ ■ → ■ f K ■ H {\displaystyle \|{\hat {\theta }}\|\to \|f_{K}\|_{H}} where f K {\displaystyle f_{K}} is the interpolating function obtained by the kernel regression with the original kernel, and ■ · ■ H {\displaystyle \|\cdot \|_{H}} is the norm in the reproducing kernel Hilbert space for the kernel. [5]

Other examples

Random binning features

A random binning features map partitions the input space using randomly shifted grids at randomly chosen resolutions and assigns to an input point a binary bit string that corresponds to the bins in which it falls. The grids are constructed so that the probability that two points x i , x j ∈ R d {\displaystyle x_{i},x_{j}\in \mathbb {R} ^{d}} are assigned to the same bin is proportional to K (x i , x

j) {\displaystyle K(x_{i},x_{j})} . The inner product between a pair of transformed points is proportional to the number of times the two points are binned together, and is therefore an unbiased estimate of K (x i , x j) {\displaystyle K(x_{i},x_{j})} . Since this mapping is not smooth and uses the proximity between input points, Random Binning Features works well for approximating kernels that depend only on the L 1 {\displaystyle L_{1}} distance between datapoints.

Orthogonal random features

Orthogonal random features [6] uses a random orthogonal matrix instead of a random Fourier matrix.

Historical context

In NIPS 2006, deep learning had just become competitive with linear models like PCA and linear SVMs for large datasets, and people speculated about whether it could compete with kernel SVMs. However, there was no way to train kernel SVM on large datasets. The two authors developed the random feature method to train those.

It was then found that the O (1 / D) {\displaystyle O(1/D)} variance bound did not match practice: the variance bound predicts that approximation to within 0.01 {\displaystyle 0.01} requires D ~ 10 4 {\displaystyle D\sim 10^{4}} , but in practice required only ~ 10 2 {\displaystyle \sim 10^{2}} . Attempting to discover what caused this led to the subsequent two papers. [2] [3] [7]

See also

Kernel method

Support vector machine

Fourier transform

Monte Carlo method

References

External links

Random Walks - Random Fourier features