Random Features for Kernel Approximation: A Survey

Blanca Cano Camarero

Universidad Autónoma de Madrid

April 25, 2023

Overview

On the Error of Random Fourier Features

- Authors: Danica J. Sutherland and Jeff Schneider
- Year 2015.
- Cited by 185.
- See Sutherland and Schneider (2015)

Why is interesting

- Which of two embedding is interesting.
- Error considerations.

Two embdedding based on Fourier tansform

One of the form:

$$\tilde{z}(x) = \sqrt{\frac{2}{D}}[\sin(w_1^T x)\cos(w_1^T x)\ldots\sin(w_{D/2}^T x)\cos(w_{D/2}^T x)]^T, w_i \sim P \quad (1)$$

and another of the form

$$\check{z}(x) = \sqrt{\frac{2}{D}}[\cos(w_1^T x + b_1)\ldots\cos(w_D^T x + b_D)]^T, w_i \sim P, b_i \sim Unif([0, 2\pi]) \quad (2)$$

Kernel approximation

$$\tilde{s}(x,y) = \frac{1}{D/2}\sum_{i=1}^{D/2}\cos({w_i}^T(x-y)) \tag{3}$$

$$\check{s}(x,y) = \frac{1}{D}\sum_{i=1}^{D}\cos({w_i}^T(x-y)) + \cos({w_i}^T(x+b)+2b_i) \tag{4}$$

Models comparatives

- of the first 100 papers cit- ing Rahimi and Recht (2007) in a Google Scholar search, used either $\tilde{z}$ or the equivalent formulation.
- The article show that $\tilde{z}$ is superior for Gaussian kernel. Providing a complementary view of the quality of these embeddings:
  - Variance of each embedding $\tilde{s}$ for RBF uniformly lower variance.
  - Uniform convergence bounds, tightening constants in the original $\tilde{z}$ (one worse).
  - L_2 convergence of each approximation $\tilde{z}$ superior again.
  - Finally empirically.

Random Features for Kernel Approximation: A Survey on Algorithms, Theory, and Beyond

- Fanghui Liu and Xiaolin Huang and Yudong Chen and Johan A. K. Suykens
- Year 2021
- Cited by 94!!!

Liu et al. (2021)

Theoretical Foundation of Random Features

Theorem

Bochner's Theorem A continuous and shift-invariant function $k : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$ *is positive definite if and only if it can be represented as*

$$k(x - x') = \int_{\mathbb{R}^d} \exp iw^T (x - x') \mu_k(d_w) \tag{5}$$

where μ_k *is a is a positive finite measure on the frequencies* w*.*

According to Bochner's theorem, the spectral distribution μ_k of a stationary kernel k is the finite measure induced by a Fourier transform.

By setting $k(0) = 1$, we may normalize μ_k to a probability density p (the Fourier transform associated with k), hence

$$k(x - x') = \int_{\mathbb{R}^d} \exp iw^T(x - x')\mu(d_w) \tag{6}$$

$$= E_{w\sim p}\left[exp(iw^T x)exp(iw^T x')^*,\right] \tag{7}$$

The kernels used in practice are typically real-valued and thus the imaginary part can be discarded.

$$\varphi_p(x) = \left[exp(-iw_1^T x), \ldots, exp(-iw_s^T x)\right]^T. \quad (8)$$

Two-layer network correspondence to random features approximation

Given a two-layer network

$$f(x;\theta) = \sqrt{\frac{2}{s}} \sum_{j=1}^{s} a_j \sigma(w_j^T x)$$

for some activation function σ and $x \in \mathbb{R}^d$ when $w \sim \mathcal{N}(0, I_d)$ and only the second layer are optimized this correspond to random features approximation

$$k(x, x') = E_{w \sim \mathcal{N}(0, I_d)} \left[\sigma(w^T x) \sigma(w^T x'), \right] \tag{9}$$

depends on the kernel type such that

$$\varphi(x) := \sigma(Wx). \tag{10}$$

Some examples:

- Correspond to a standard RFF model for a Gaussian kernel:

$$\sigma(x) = [\cos(x)\sin(x)] \tag{11}$$

- First order arc-cosine kernel termed as

$$k(x, x') = k(u) = \frac{1}{\pi}(u(\pi - \arccos(u)) + \sqrt{1 - u^2}) \tag{12}$$

where

$$u = \frac{\langle x, x' \rangle}{\|x\|\|x'\|} \tag{13}$$

if $\sigma(x) = \max\{0, x\}$.

For fully connected deep neural network

Pregunta: Una red neuronal de dos capas representa un kernel. Ese kernel se puede aproximar a una red neuronales de una capa oculta. LUEGO SE ESTARÍA TRANSFORMANDO LAS REDES NEURALES DE UNA CAPA A OTRA.
Referencias que ver Daniely et al. (2017) and Lee et al. (2018) and from the origin article references (12, 63,64,65,66,67 and 68).

Commonly used kernels in Random Features

Actually not interesting

Taxonomy of random features based algorithms

$$\varphi(x) = \left[a_1 \exp\left(-iw_1^T x\right), \ldots, a_s exp(-iw_s^T x),\right]^T \tag{14}$$

They can be grouped into two categories, data-independent algorithms and data-dependent algorithms, based on whether or not the selection of w_i and a_i is independent of the training data

See figure 2 of the article.

And Rahimi and Recht (2008) is improved by Le et al. (2014)

Error approaches

- **Approximation**: how many random features are needed to ensure a high quality estimator in kernel approximation?
- **Generalization** how many random features are needed to incur no loss of empirical risk and expected risk in a learning estimator?
  - See convergences on table 3 of Liu et al. (2021).
  - Rates and required random features in table 4.
  - Comparison of learning rates and required random features for expected risk with a Lipschitz continuous loss function. Table 5.

Other comparation and interesting fields

- Space and time complexities see table 2 page 8 of Liu et al. (2021).
- Trends: High-dimensional Random Features in Over parameterized settings (page 21)
  - Double descent in the test error curve (as we saw in DNN)

Future

RFF in DNNs!!! See references Figure 1 Liu et al. (2021).

Next week: Neural Network?

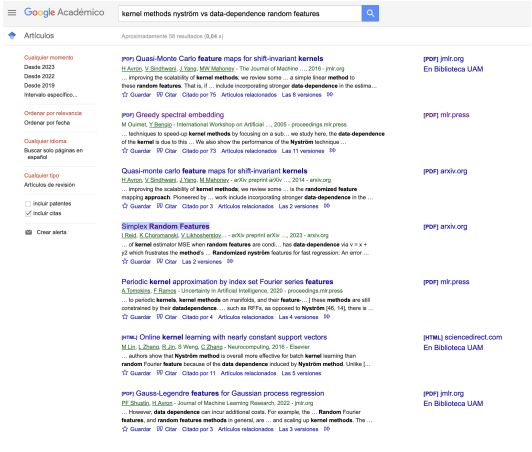
1. **Nystroem Method vs Random Fourier Features: A Theoretical and Empirical Comparison, Advances in Neural Information Processing Systems 2012**
2. **Random features for kernel approximation: A survey on algorithms, theory, and beyond**
3. **Williams, C.K.I. and Seeger, M. "Using the Nystroem method to speed up kernel machines", Advances in neural information processing systems 2001 T. Yang, Y. Li, M. Mahdavi, R. Jin and Z. Zhou**
4. Weighted Sums of Random Kitchen Sinks: Replacing minimization with randomization in learning.
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

Previous meets

1. Nyström vs Random Fourier Features.
   - When eigenvalues are sparse Nyström is better.
   - Article hypothesis: training data dependence.
2. Random Features for kernel approximation survey.
   - Random features general approach.
   - Survey of plenty techniques: **data-independent and data-dependent**.

Some questions

If the article hypothesis are correct, Does data independent solve the problem?



Questions

- Why are there no comparisions articles?
- The hypothesis was wrong?
- Why there are no comparative between algorithm and spectral sparsity.
- How relevant is spectral sparsity?
- If it is important why not a lineal combination models and adjust the coefficient by some spectral approximation or heuristic?

This survey would indirectly answer some of the questions

Taxonomy

- data-independent
  - i) Monte Carlo sampling
    - acceleration
      - structural: Fastfood [36], $\mathcal{P}$-model [41], SORF [24]
      - circulant: SCRF [40]
    - variance reduction
      - ℓ_2 normalization: NRFF [79]
      - orthogonal constraint: ORF [24], ROM [80]
  - ii) Quasi-Monte Carlo sampling
    - QMC [37]
    - structural spherical feature: SSF [43]
    - moment matching: MM [42]
  - iii) Quadrature rules
    - deterministic quadrature rules: GQ, SGQ [26]
    - stochastic spherical-radial rule: SSR [27]
- data-dependent
  - leverage score sampling: LSS-RFF [31], fast leverage score approximation [47], [48], [81]
  - re-weighted random features
    - weighted random features: [32], [82] for RFF, [25] for QMC, [26] for GQ
    - kernel alignment: KA-RFF [83] and KP-RFF [44]
    - compressed low-rank approximation: CLR-RFF [46]
  - kernel learning by random features
    - one-stage: [84] via generative models
    - two-stage
      - joint optimization: [85], [86]
      - spectral learning in mixture models: [87], [88], [89], [90]
  - others: quantization [45]; doubly stochastic [38]

Space and time complexities: No clear difference

Table 2
Comparison of different kernel approximation methods on space and time complexities to obtain $\boldsymbol{W}\boldsymbol{x}$.

Method	Kernels (in theory)	Extra Memory	Time	Lower variance than RFF
Random Fourier Features (RFF) [9]	shift-invariant kernels	$\mathcal{O}(sd)$	$\mathcal{O}(sd)$	-
Quasi-Monte Carlo (QMC) [37]	shift-invariant kernels	$\mathcal{O}(sd)$	$\mathcal{O}(sd)$	Yes
Normalized RFF (NRFF) [79]	Gaussian kernel	$\mathcal{O}(sd)$	$\mathcal{O}(sd)$	Yes
Moment matching (MM) [42]	shift-invariant kernels	$\mathcal{O}(sd)$	$\mathcal{O}(sd)$	Yes
Orthogonal Random Feature (ORF) [24]	Gaussian kernel	$\mathcal{O}(sd)$	$\mathcal{O}(sd)$	Yes
Fastfood [36]	Gaussian kernel	$\mathcal{O}(s)$	$\mathcal{O}(s \log d)$	No
Spherical Structured Features (SSF) [43]	shift and rotation-invariant kernels	$\mathcal{O}(s)$	$\mathcal{O}(s \log d)$	Yes
Structured ORF (SORF) [24], [91]	shift and rotation-invariant kernels	$\mathcal{O}(s)$	$\mathcal{O}(s \log d)$	Unknown
Signed Circulant (SCRF) [40]	shift-invariant kernels	$\mathcal{O}(s)$	$\mathcal{O}(s \log d)$	The same
$\mathcal{P}$-model [41]	shift and rotation-invariant kernels	$\mathcal{O}(s)$	$\mathcal{O}(s \log d)$	No
Random Orthogonal Embeddings (ROM) [80]	rotation-invariant kernels	$\mathcal{O}(d)$	$\mathcal{O}(d \log d)$	Yes
Gaussian Quadrature (GQ), Sparse Grids Quadrature (SGQ) [26]	shift invariant kernels	$\mathcal{O}(d)$	$\mathcal{O}(d \log d)$	Yes
Stochastic Spherical-Radial rules (SSR) [27]	shift and rotation-invariant kernels	$\mathcal{O}(d)$	$\mathcal{O}(d \log d)$	Yes

Models:

Here we test various random features based algorithms:

Data-independient	Data-dependient
RFF	LS-RFF
ORF	
SORF	
ROM	
Fastfood	
QMC	
SSF	
GQ	

Biased tests!

Datesets:

Table 6
Dataset statistics.

datasets	d	#traing	#test	random split	scaling
ijcnn1	22	49,990	91,701	no	-
EEG	14	7,490	7,490	yes	mapstd
cod-RNA	8	59,535	157,413	no	mapstd
covtype	54	290,506	290,506	yes	minmax
magic04	10	9,510	9,510	yes	minmax
letter	16	12,000	6,000	no	minmax
skin	3	122,529	122,529	yes	minmax
a8a	123	22,696	9,865	no	-
MNIST	784	60,000	10,000	no	minmax
CIFAR-10	3072	50,000	10,000	no	-
MNIST-8M	784	8,100,000	10,000	no	-

Results:

datasets	approximation		lr		liblinear	
	small s	large s	small s	large s	small s	large s
ijcnn1	SSF	SORF, QMC, ORF	-	-	Fastfood	-
EEG	SSF	ORF	-	-	-	-
cod-RNA	SSF	-	-	-	-	-
covtype	ORF	-	-	-	-	-
magic04	SSF	SSF, ORF, QMC, ROM	-	-	-	-
letter	SSF	SSF, ORF	-	-	-	-
skin	SSF, ROM	QMC	-	-	-	-
a8a	-	-	-	-	SSF	-

- Approximation: small (SSF: 8/10), large (ORF 4/10, QMC: 3/10, SSF: 2/10).
- Lineal regression: Adjusting the model kernel error are alleviated.

Having in mind taxonomy

- SSF: data independent, Quasi-Monte Carlo sampling.
- ORF: data independent, Monte Carlo sampling.
- QMC: data independent, Quasi-Monte Carlo sampling.

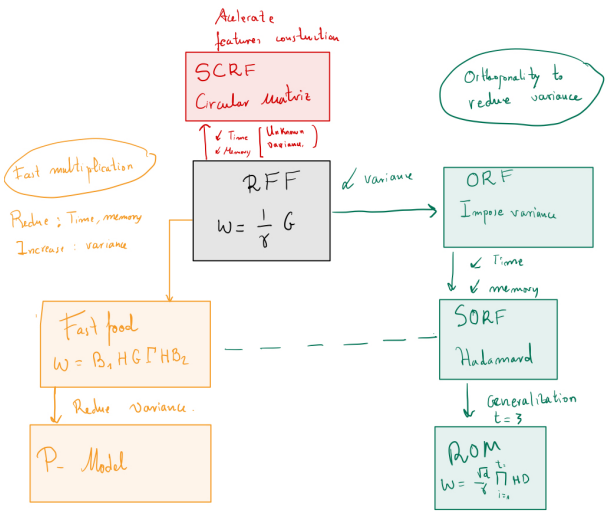
Where is LS-RFF? (See appendix B)

- Highest (except ijcnn1) approximation error.
- Most expensive time cost.

Why they did not test data indepence techniques?

- Not worthy? (A bit weird)

Monte Carlo sampling based approaches



Hadamard matrix (wikipedia)

The Hadamard matrix can be constructed using the Sylvester construction, which involves recursively combining smaller Hadamard matrices to form larger ones.
Start with the 1×1 Hadamard matrix $H_1 = [1]$. To construct the $2n \times 2n$ Hadamard matrix H_{2n}, form the matrix

$$H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}.$$

Using this construction, we can generate Hadamard matrices of any power of 2 size. For example, we can construct the 4×4 Hadamard matrix as follows:

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

Walsh-Hadamard matrix

Rearrange the rows of the matrix according to the number of sign change of each row.

$$W_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

Note that the Walsh matrix of size 4 is equivalent to the second-order Hadamard matrix that I gave earlier.

Quasi-Monte Carlo Sampling

Hypothesis: p factorizes with respect to the dimension

$$p(x) = \prod_{j=1}^{d} p_j(x_j)$$

QMC transform the integral to

$$k(x - x') = \int_{[0,1]^d} \exp(i(x - x')^T \phi^{-1}(t))dt,$$

where $\phi^{-1}(t) = (\phi_1^{-1}(t_1), \ldots, \phi_d^{-1}(t_d))$ where $\{t\}$ is a low discrepancy sequence.

$$W = [\phi_1^{-1}(t_1), \ldots, \phi_d^{-1}(t_s)]$$

SSF: Spherical structured feature maps for kernel approximation

It improves the space and time complexities of QMC for approximating shift- and rotation-invariant kernels
SSF generates points

$$\{v_1, v_2, \ldots, v_s\}$$

asymptotically uniformly distributed on the sphere $S_{d-1}$and constructs the transformation matrix as

$$W_{\text{SSF}} = \left[\Phi^{-1}(t)v_1, \Phi^{-1}(t)v_2, \ldots, \Phi^{-1}(t)v_s\right]^T \in \mathbb{R}^{s\times d},$$

where $\Phi^{-1}(t)$ is the inverse of the cumulative distribution function of the standard normal distribution evaluated at t.

$$V = \frac{1}{\sqrt{d/s}} \begin{bmatrix} \Re F_\Lambda & -\Im F_\Lambda \\ \Im F_\Lambda & \Re F_\Lambda \end{bmatrix}$$

where F_Λ consiste of the row of the Fourier discrete matrix.
The discrete Fourier transform (DFT) matrix is defined as:

$$F_N = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_N & w_N^2 & \cdots & w_N^{N-1} \\ 1 & w_N^2 & w_N^4 & \cdots & w_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_N^{N-1} & w_N^{2(N-1)} & \cdots & w_N^{(N-1)^2} \end{pmatrix},$$

where $w_N = e^{-2\pi i/N}$ is a primitive N-th root of unity.

The matrix F_N is a unitary matrix and its inverse is given by the Hermitian transpose, i.e., $F_N^{-1} = F_N^H$.

$$F = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

The selection of d^2 rows from F is done by minimizing the discrete Riesz 0-energy such that the points spread as evenly as possible on the sphere.

Leverage score based sampling: main idea

Generate samples w from a distribution q.
Feature mapping:

$$\varphi_q(x) = \frac{1}{\sqrt{s}} \left(\sqrt{\frac{p(w_1)}{q(w_1)}} e^{-iw_1^T x}, \dots \sqrt{\frac{p(w_s)}{q(w_s)}} e^{-iw_s^T x} \right)^T \tag{15}$$

How to design q

Let define ridge leverage function:

$$l_\lambda(w_i) = p(w_i) z_{p,w_i}^T(X)(K + n\lambda I)^{-1} z_{p,w_i}(X), \tag{16}$$

where λ s the KRR regularization parameter.
Define

$$d_K^\lambda = \int_{\mathbb{R}^d} l_\lambda(w) dw = tr[K(K + n\lambda I)^{-1}]. \tag{17}$$

Number of effective degrees of freedom (independent parameters in a learning problem)
$d^\lambda << n$
q is designed as

$$q(w) = \frac{l_\lambda(w)}{d_K^\lambda}. \tag{18}$$

Improvements

- Cost K approximation and inverse $(O(n^3))$
- LS-RFF uses a subset of data to approximate K. $(O(ns^2 + s^3))$.
- SLS-RFF (Surrogate Leverage Score-RFF) to avoid inverting an $s \times s$ matrix.$O(ns^2)$ same complexity as RFF.

Re-weighted random features

- KA-RFF (Kernel Alignment-RFF) : It pre-computes a large number of random features that are generated by RFF, and then select a subset of them by solving a simple optimization problem based on kernel alignment. In particular, the optimization problem is

$$\max_{a \in P_J} \sum_{n}^{i,j=1} y_i y_j \sum_{t=1}^{J} a_t z_p(x_i, w_t) z_p(x_j, w_t). \tag{19}$$

- KP-RFF (Kernel Polarization-RFF) and CLR-RFF (Compression Low Rank-RFF): It first generates a large number of random features by RFF and then selects a subset from them using an energy-based scheme or solving and optimization problem.

Hilbert coreset construction problem

The problem of building a small, weighted subset of the data that approximates the full dataset, is known as the **Hilbert coreset construction problem**. Ans some solution:

- Greedy iterative geodesic ascent,
- Frank-Wolfe based methods,
- Johnson-Lindenstrauss random projection.

Kernel learning by random features

Construct random features using sophisticated learning techniques, e.g., by learning the spectral distribution of kernel from the data.

- Representative approaches in this class often involve a one- stage or two-stage process.
  1. Learns the random features,
  2. and then incorporates them into kernel methods for prediction.
- Example: leverage sampling and random features selection.
- One-stage algorithms aim to simultaneously learn the spectral distribution of a kernel and the prediction model by solving a single joint optimization problem or using a spectral inference scheme.

computational cost

Random Fourier Features Li (2021): reducing the computational complexity (roughly from $O(n^3)$ in time and $O(n^2)$ in space to $O(n^2)$ and $O(n\sqrt{n})$ respectively) without having to trade-off the statistical prediction accuracy.

Step	Task	Theory	Cost	Memory
1	Sampling Fourier components $u_1, \dots u_m$ from $p(u)$.	Inverse transform sampling, Accept or reject, Montecarlo.	O(1)	O(m)
2	Compute $z_f(x) = (\sin(u_1^T x), \cos(u_1^T x), \dots, \sin(u_m^T x), \cos(u_m^T x))$		O(m)	O(2m)

Nyström

Step	Task	Theory	Cost	Memory
1	Sampling		O(m)	O(m)
2	constructing a low rank matrix by $\hat{K}_r = K_b \hat{K}^{\dagger} K_b^T$.	SVD and matrix multiplication	$O(n^3)$	$O(m^2)$

Training model	Cost	Memory
Ridge regression	$O(m^2(N+m))$	$O(mn+n)$
KRR	$O(n^3)$	$O(n^2)$
SVM	$O(n^2)$ or $O(n^3)$	$O(n^2)$
Backpropagation one layer	$O(nod)$ (o output, d: mini batch size)	$O(no)$

References I

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity, 2017.

Quoc Viet Le, Tamas Sarlos, and Alexander Johannes Smola. Fastfood: Approximate kernel expansions in loglinear time, 2014.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes, 2018.

Zhu Li. Sharp analysis of random fourier features in classification, 2021.

Fanghui Liu, Xiaolin Huang, Yudong Chen, and Johan A. K. Suykens. Random features for kernel approximation: A survey on algorithms, theory, and beyond, 2021.

References II

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL `https://proceedings.neurips.cc/paper_files/paper/2008/file/0efe32849d230d7f53049ddc4a4b0c60-Paper.pdf`.

Danica J. Sutherland and Jeff Schneider. On the error of random fourier features, 2015.