RANDOM FOURIER FEATURES FOR KERNEL RIDGE REGRESSION: APPROXIMATION BOUNDS AND STATISTICAL GUARANTEES

Haim Avron, Michael Kapralov, **Cameron Musco**, Christopher Musco, Ameya Velingker, and Amir Zandieh

Tel Aviv University, EPFL, and MIT.
ICML 2017.

Our Contributions:

Our Contributions:

- Analyze the random Fourier features method (Rahimi Recht '07) for kernel approximation using leverage score-based techniques.

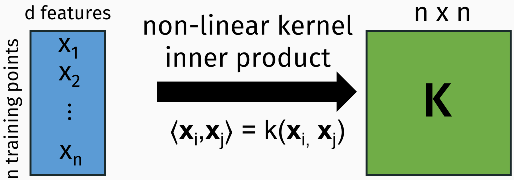
Our Contributions:

- Analyze the random Fourier features method (Rahimi Recht '07) for kernel approximation using leverage score-based techniques.
- **Concrete:** Introduce new sampling distribution that gives statistical guarantees for kernel ridge regression when used to approximate the Gaussian kernel.

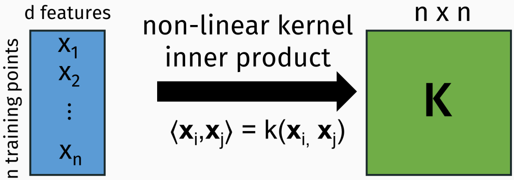
Our Contributions:

- Analyze the random Fourier features method (Rahimi Recht '07) for kernel approximation using leverage score-based techniques.
- **Concrete:** Introduce new sampling distribution that gives statistical guarantees for kernel ridge regression when used to approximate the Gaussian kernel.
- **High Level:** Hope that Fourier leverage scores will have further applications in kernel approximation, function approximation, and sparse Fourier transform methods.

Kernel methods are expensive.

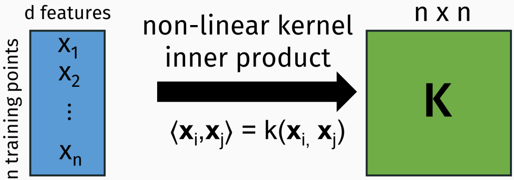


Kernel methods are expensive.



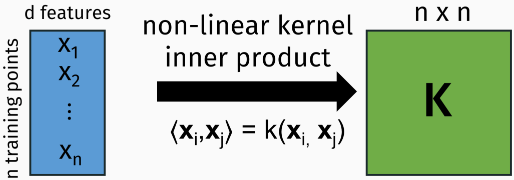
- Even writing down $\mathbf{K}$ requires $\Omega(n^2)$ time.

Kernel methods are expensive.



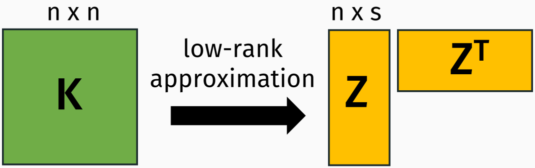
- Even writing down **K** requires $\Omega(n^2)$ time.
- Other operations require even more. A single iteration of a linear system solver takes $\Omega(n^2)$ time.

Kernel methods are expensive.



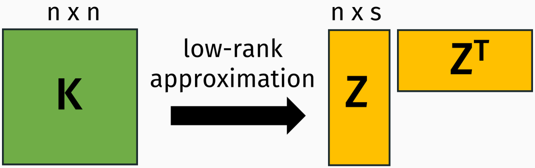
- Even writing down $\mathbf{K}$ requires $\Omega(n^2)$ time.
- Other operations require even more. A single iteration of a linear system solver takes $\Omega(n^2)$ time.
- For $n = 100,000$, $\mathbf{K}$ has 10 billion entries. Takes 80 GB of storage if each is a `double`.

Employ classic solution: low-rank approximation



SOLUTION: LOW-RANK APPROXIMATION.

Employ classic solution: low-rank approximation



- Storing $\mathbf{Z}$ uses $O(ns)$ space and computing $\mathbf{Z}\mathbf{Z}^T\mathbf{x}$ takes $O(ns)$ time. Orthogonalization, eigendecomposition, and pseudo-inversion of $\mathbf{Z}\mathbf{Z}^T$ all take just $O(ns^2)$ time.

Low-rank approximation is itself an expensive task.

EFFICIENT LOW-RANK APPROXIMATION?

Low-rank approximation is itself an expensive task.

- Optimal low-rank approximation via a direct eigendecomposition, or even approximation via Krylov subspace methods are out of the question since they at least require fully forming $\mathbf{K}$.

Low-rank approximation is itself an expensive task.

- Optimal low-rank approximation via a direct eigendecomposition, or even approximation via Krylov subspace methods are out of the question since they at least require fully forming $\mathbf{K}$.
- Many faster methods have been studied: incomplete Cholesky factorization (Fine & Scheinberg '02, Bach & Jordan '02), entrywise sampling (Achlioptas, McSherry, & Schölkopf '01), Nyström approximation (Williams & Seeger '01), random Fourier features (Rahimi & Recht '07)

EFFICIENT LOW-RANK APPROXIMATION?

Low-rank approximation is itself an expensive task.

- Optimal low-rank approximation via a direct eigendecomposition, or even approximation via Krylov subspace methods are out of the question since they at least require fully forming $\mathbf{K}$.
- Many faster methods have been studied: incomplete Cholesky factorization (Fine & Scheinberg '02, Bach & Jordan '02), entrywise sampling (Achlioptas, McSherry, & Schölkopf '01), Nyström approximation (Williams & Seeger '01), random Fourier features (Rahimi & Recht '07)

Rahimi & Recht NIPS ‘07:

Rahimi & Recht NIPS '07:

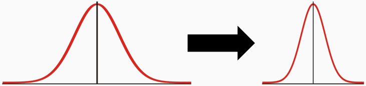
- For any shift-invariant $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i - \mathbf{x}_j)$ let $p(\cdot)$ be the Fourier transform of $k(\cdot)$. By Bochner's theorem, $p(\boldsymbol{\eta}) \geq 0$ for all $\boldsymbol{\eta}$.

Rahimi & Recht NIPS '07:

- For any shift-invariant $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i - \mathbf{x}_j)$ let $p(\cdot)$ be the Fourier transform of $k(\cdot)$. By Bochner's theorem, $p(\boldsymbol{\eta}) \geq 0$ for all $\boldsymbol{\eta}$.
- Sample $\boldsymbol{\eta}_1, ..., \boldsymbol{\eta}_s \in \mathbb{R}^d$ with probabilities proportional to $p(\boldsymbol{\eta})$.

Rahimi & Recht NIPS '07:

- For any shift-invariant $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i - \mathbf{x}_j)$ let $p(\cdot)$ be the Fourier transform of $k(\cdot)$. By Bochner's theorem, $p(\boldsymbol{\eta}) \geq 0$ for all $\boldsymbol{\eta}$.
- Sample $\boldsymbol{\eta}_1, ..., \boldsymbol{\eta}_s \in \mathbb{R}^d$ with probabilities proportional to $p(\boldsymbol{\eta})$.

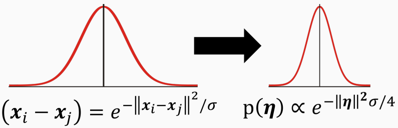


$k(\boldsymbol{x}_i - \boldsymbol{x}_j) = e^{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/\sigma}$ $\quad$ $\mathrm{p}(\boldsymbol{\eta}) \propto e^{-\|\boldsymbol{\eta}\|^2\sigma/4}$

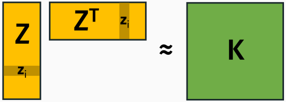
RANDOM FOURIER FEATURES

Rahimi & Recht NIPS '07:

- For any shift-invariant $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i - \mathbf{x}_j)$ let $p(\cdot)$ be the Fourier transform of $k(\cdot)$. By Bochner's theorem, $p(\boldsymbol{\eta}) \geq 0$ for all $\boldsymbol{\eta}$.
- Sample $\boldsymbol{\eta}_1, ..., \boldsymbol{\eta}_s \in \mathbb{R}^d$ with probabilities proportional to $p(\boldsymbol{\eta})$.

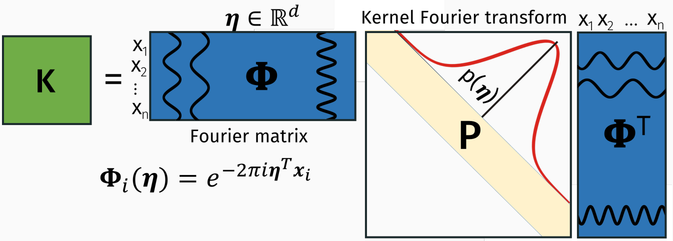


- Set $\mathbf{z}_i = \frac{1}{\sqrt{s}}\left[e^{-2\pi i \boldsymbol{\eta}_1^T \mathbf{x}_i}, ..., e^{-2\pi i \boldsymbol{\eta}_s^T \mathbf{x}_i}\right]$.

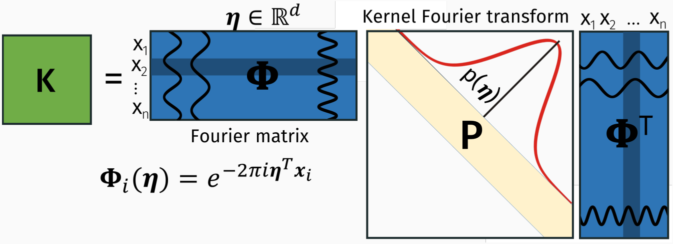


Fourier transform $k(\mathbf{z}) = \int_{\boldsymbol{\eta} \in \mathbb{R}^d} p(\boldsymbol{\eta}) e^{-2\pi i \boldsymbol{\eta}^T \mathbf{z}} d\boldsymbol{\eta}$ gives:

Fourier transform $k(\mathbf{z}) = \int_{\boldsymbol{\eta}\in\mathbb{R}^d} p(\boldsymbol{\eta}) e^{-2\pi i \boldsymbol{\eta}^T \mathbf{z}} d\boldsymbol{\eta}$ gives:



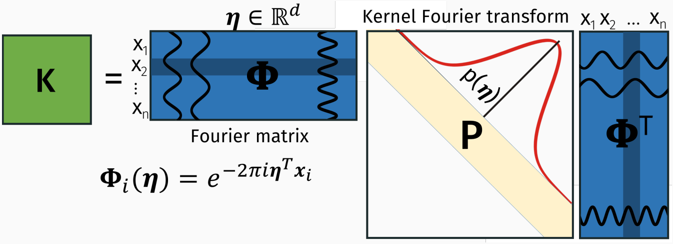
Fourier transform $k(\mathbf{z}) = \int_{\boldsymbol{\eta}\in\mathbb{R}^d} p(\boldsymbol{\eta}) e^{-2\pi i \boldsymbol{\eta}^T \mathbf{z}} d\boldsymbol{\eta}$ gives:



- $\int_{\boldsymbol{\eta}} \boldsymbol{\Phi}_i(\eta) p(\eta) \boldsymbol{\Phi}_j(\eta)^* d\eta$

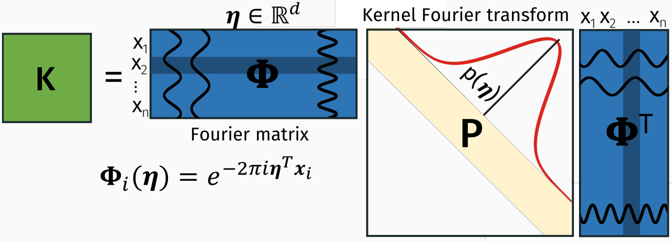
VIEW AS MATRIX SAMPLING METHOD

Fourier transform $k(\mathbf{z}) = \int_{\boldsymbol{\eta}\in\mathbb{R}^d} p(\boldsymbol{\eta}) e^{-2\pi i \boldsymbol{\eta}^T \mathbf{z}} d\boldsymbol{\eta}$ gives:



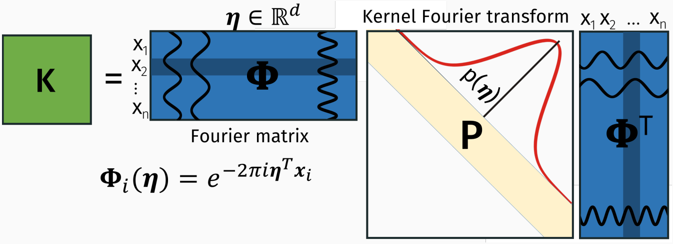
- $\int_{\boldsymbol{\eta}} \boldsymbol{\Phi}_i(\eta) p(\eta) \boldsymbol{\Phi}_j(\eta)^* d\eta = \int_{\boldsymbol{\eta}} e^{-2\pi i \boldsymbol{\eta}^T (\mathbf{x}_i - \mathbf{x}_j)} p(\eta) d\eta$

Fourier transform $k(\mathbf{z}) = \int_{\boldsymbol{\eta}\in\mathbb{R}^d} p(\boldsymbol{\eta}) e^{-2\pi i \boldsymbol{\eta}^T \mathbf{z}} d\boldsymbol{\eta}$ gives:



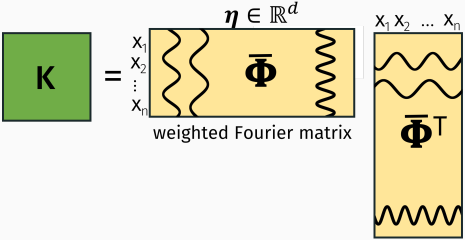
- $\int_{\boldsymbol{\eta}} \boldsymbol{\Phi}_i(\eta) p(\eta) \boldsymbol{\Phi}_j(\eta)^* d\eta = \int_{\boldsymbol{\eta}} e^{-2\pi i \boldsymbol{\eta}^T(\mathbf{x}_i - \mathbf{x}_j)} p(\boldsymbol{\eta}) d\boldsymbol{\eta} = k(\mathbf{x}_i - \mathbf{x}_j)$

Fourier transform $k(\mathbf{z}) = \int_{\boldsymbol{\eta}\in\mathbb{R}^d} p(\boldsymbol{\eta}) e^{-2\pi i \boldsymbol{\eta}^T \mathbf{z}} d\boldsymbol{\eta}$ gives:



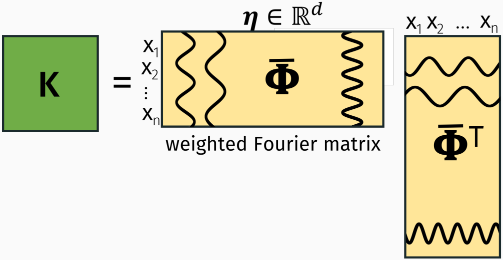
- $\int_{\boldsymbol{\eta}} \boldsymbol{\Phi}_i(\eta) p(\eta) \boldsymbol{\Phi}_j(\eta)^* d\eta = \int_{\boldsymbol{\eta}} e^{-2\pi i \boldsymbol{\eta}^T (\mathbf{x}_i - \mathbf{x}_j)} p(\eta) d\eta = k(\mathbf{x}_i - \mathbf{x}_j) = \mathbf{K}_{i,j}$.

Fourier transform $k(\mathbf{z}) = \int_{\boldsymbol{\eta}\in\mathbb{R}^d} p(\boldsymbol{\eta}) e^{-2\pi i \boldsymbol{\eta}^T \mathbf{z}} d\boldsymbol{\eta}$ gives:

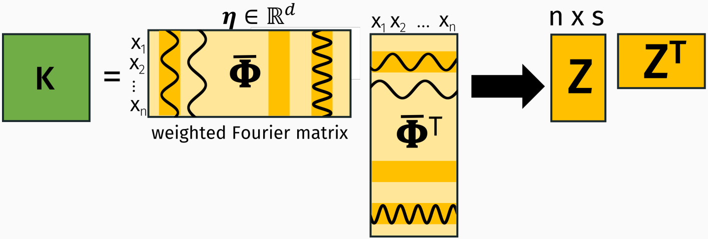


- $\int_{\boldsymbol{\eta}} \mathbf{\Phi}_i(\boldsymbol{\eta}) p(\boldsymbol{\eta}) \mathbf{\Phi}_j(\boldsymbol{\eta})^* d\boldsymbol{\eta} = \int_{\boldsymbol{\eta}} e^{-2\pi i \boldsymbol{\eta}^T(\mathbf{x}_i - \mathbf{x}_j)} p(\boldsymbol{\eta}) d\boldsymbol{\eta} = k(\mathbf{x}_i - \mathbf{x}_j) = \mathbf{K}_{i,j}$.
- Set $\bar{\mathbf{\Phi}} = \mathbf{\Phi}\mathbf{P}^{1/2}$. So $\mathbf{K} = \bar{\mathbf{\Phi}}\bar{\mathbf{\Phi}}^T$.

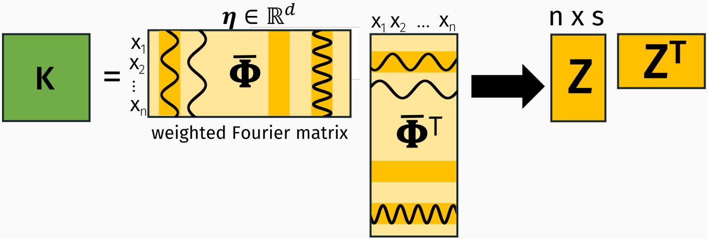
VIEW AS MATRIX SAMPLING METHOD



VIEW AS MATRIX SAMPLING METHOD

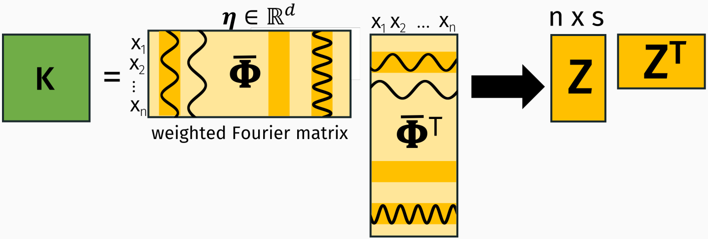


- $\mathbf{Z}(j) = \frac{1}{\sqrt{sp(\boldsymbol{\eta})}} \bar{\mathbf{\Phi}}(\boldsymbol{\eta})$ with probability $p(\boldsymbol{\eta})$. So $\mathbb{E}[\mathbf{Z}\mathbf{Z}^T] = \mathbf{K}$.

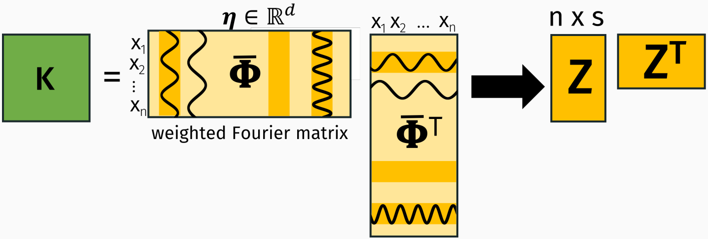


- $\mathbf{Z}(j) = \frac{1}{\sqrt{sp(\boldsymbol{\eta})}} \bar{\boldsymbol{\Phi}}(\boldsymbol{\eta})$ with probability $p(\boldsymbol{\eta})$. So $\mathbb{E}[\mathbf{Z}\mathbf{Z}^T] = \mathbf{K}$.
- $\mathbf{z}_i = \frac{1}{\sqrt{s}} \left[e^{-2\pi i \boldsymbol{\eta}_1^T \mathbf{x}_i}, \ldots, e^{-2\pi i \boldsymbol{\eta}_s^T \mathbf{x}_i} \right]$ for $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_s$ sampled according to $p(\boldsymbol{\eta})$.

VIEW AS MATRIX SAMPLING METHOD

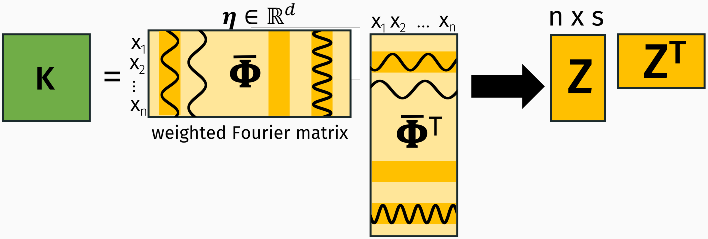


VIEW AS MATRIX SAMPLING METHOD



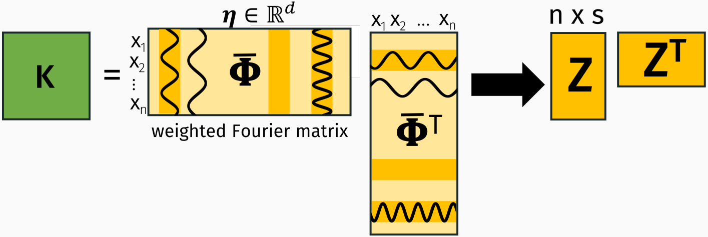
- $\mathbf{Z}$ is a sample of $\bar{\mathbf{\Phi}} = \mathbf{\Phi}\mathbf{P}^{1/2}$. Columns are sampled with probability $\propto p(\boldsymbol{\eta})$, i.e., their squared column norms.

VIEW AS MATRIX SAMPLING METHOD



- $\mathbf{Z}$ is a sample of $\bar{\boldsymbol{\Phi}} = \boldsymbol{\Phi}\mathbf{P}^{1/2}$. Columns are sampled with probability $\propto p(\boldsymbol{\eta})$, i.e., their squared column norms.
- It is well known from work on randomized methods in linear algebra that there are better sampling probabilities (in both theory and practice): the column leverage scores.

VIEW AS MATRIX SAMPLING METHOD



- $\mathbf{Z}$ is a sample of $\bar{\mathbf{\Phi}} = \mathbf{\Phi}\mathbf{P}^{1/2}$. Columns are sampled with probability $\propto p(\boldsymbol{\eta})$, i.e., their squared column norms.
- It is well known from work on randomized methods in linear algebra that there are better sampling probabilities (in both theory and practice): the column leverage scores.
- Also noted by Bach '17, implicit in Rudi et al. '16.

Column Norm Sampling: $s = \tilde{O}(d/\epsilon^2)$ samples ensure that $\left(\mathbf{Z}\mathbf{Z}^T\right)_{i,j} = \mathbf{K}_{i,j} \pm \epsilon$ for all i, j with high probability [RR07].

Column Norm Sampling: $s = \tilde{O}(d/\epsilon^2)$ samples ensure that $\left(\mathbf{ZZ}^T\right)_{i,j} = \mathbf{K}_{i,j} \pm \epsilon$ for all i, j with high probability [RR07].

Ridge Leverage Score Sampling: $s = \tilde{O}(s_\lambda/\epsilon^2)$ samples gives spectral approximation:

$$(1-\epsilon)(\mathbf{ZZ}^T + \lambda \mathbf{I}) \preceq \mathbf{K} + \lambda \mathbf{I} \preceq (1+\epsilon)(\mathbf{ZZ}^T + \lambda \mathbf{I}).$$

where $s_\lambda = tr(\mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1})$ is the statistical dimension.

Column Norm Sampling: $s = \tilde{O}(d/\epsilon^2)$ samples ensure that $\left(\mathbf{ZZ}^T\right)_{i,j} = \mathbf{K}_{i,j} \pm \epsilon$ for all i, j with high probability [RR07].

Ridge Leverage Score Sampling: $s = \tilde{O}(s_\lambda/\epsilon^2)$ samples gives spectral approximation:

$$(1-\epsilon)(\mathbf{ZZ}^T + \lambda\mathbf{I}) \preceq \mathbf{K} + \lambda\mathbf{I} \preceq (1+\epsilon)(\mathbf{ZZ}^T + \lambda\mathbf{I}).$$

where $s_\lambda = tr(\mathbf{K}(\mathbf{K} + \lambda\mathbf{I})^{-1})$ is the statistical dimension.

- Spectral approximation gives statistical guarantees for kernel ridge regression (this work), and approximation bounds for kernel PCA and k-means clustering (Cohen, Musco, Musco ‘16,‘17)

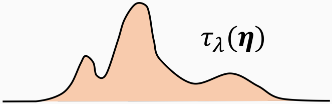
The ridge leverage score for frequency $\boldsymbol{\eta}$ is given by:

$$\tau_\lambda(\boldsymbol{\eta}) = \bar{\boldsymbol{\Phi}}(\boldsymbol{\eta})^T(\mathbf{K} + \lambda\mathbf{I})^{-1}\bar{\boldsymbol{\Phi}}(\boldsymbol{\eta}).$$

The ridge leverage score for frequency $\boldsymbol{\eta}$ is given by:

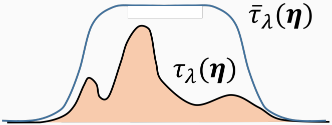
$$\tau_\lambda(\boldsymbol{\eta}) = \bar{\boldsymbol{\Phi}}(\boldsymbol{\eta})^T(\mathbf{K} + \lambda\mathbf{I})^{-1}\bar{\boldsymbol{\Phi}}(\boldsymbol{\eta}).$$

- Expensive to invert $\mathbf{K} + \lambda\mathbf{I}$. But even if you could do that efficiently, it is not at all clear you could efficiently sample from the leverage score distribution.

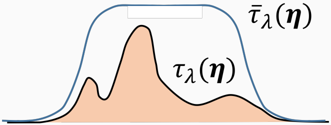


Main goal: Get a handle on the Fourier ridge leverage scores for common kernels by upper bounding them with simple distributions.

Main goal: Get a handle on the Fourier ridge leverage scores for common kernels by upper bounding them with simple distributions.



Main goal: Get a handle on the Fourier ridge leverage scores for common kernels by upper bounding them with simple distributions.



1. Improve random Fourier features.
2. Bound statistical dimension by the sum of leverage scores.
3. Connections with sparse Fourier transforms, Fourier interpolation, and other problems.

ALTERNATIVE LEVERAGE SCORE CHARACTERIZATION

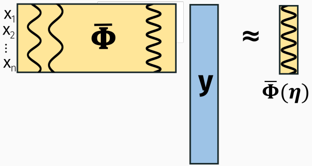
Ridge leverage score $\tau_\lambda(\boldsymbol{\eta}) = \bar{\mathbf{\Phi}}(\boldsymbol{\eta})^T(\mathbf{K}+\lambda\mathbf{I})^{-1}\bar{\mathbf{\Phi}}(\boldsymbol{\eta})$ also equals:

$$\tau_\lambda(\boldsymbol{\eta}) = \min_{\mathbf{y}} \lambda^{-1}\|\bar{\mathbf{\Phi}}\mathbf{y} - \bar{\mathbf{\Phi}}(\boldsymbol{\eta})\|_2^2 + \|\mathbf{y}\|_2^2.$$

Ridge leverage score $\tau_\lambda(\boldsymbol{\eta}) = \bar{\boldsymbol{\Phi}}(\boldsymbol{\eta})^T(\mathbf{K} + \lambda\mathbf{I})^{-1}\bar{\boldsymbol{\Phi}}(\boldsymbol{\eta})$ also equals:

$$\tau_\lambda(\boldsymbol{\eta}) = \min_{\mathbf{y}} \lambda^{-1}\|\bar{\boldsymbol{\Phi}}\mathbf{y} - \bar{\boldsymbol{\Phi}}(\boldsymbol{\eta})\|_2^2 + \|\mathbf{y}\|_2^2.$$

Intuition: $\tau_\lambda(\boldsymbol{\eta})$ is small iff there exists a function $\mathbf{y} : \mathbb{R}^d \to \mathbb{C}$ with low energy ($\|\mathbf{y}\|_2^2$ small) whose ($\sqrt{p(\boldsymbol{\eta})}$ weighted) Fourier transform is close to the frequency $e^{-2\pi i \mathbf{x}_j^T \boldsymbol{\eta}}$ at each data point.

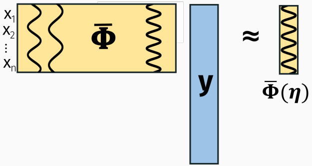


ALTERNATIVE LEVERAGE SCORE CHARACTERIZATION

Ridge leverage score $\tau_\lambda(\boldsymbol{\eta}) = \bar{\boldsymbol{\Phi}}(\boldsymbol{\eta})^T(\mathbf{K} + \lambda\mathbf{I})^{-1}\bar{\boldsymbol{\Phi}}(\boldsymbol{\eta})$ also equals:

$$\tau_\lambda(\boldsymbol{\eta}) = \min_{\mathbf{y}} \lambda^{-1}\|\bar{\boldsymbol{\Phi}}\mathbf{y} - \bar{\boldsymbol{\Phi}}(\boldsymbol{\eta})\|_2^2 + \|\mathbf{y}\|_2^2.$$

Intuition: $\tau_\lambda(\boldsymbol{\eta})$ is small iff there exists a function $\mathbf{y} : \mathbb{R}^d \to \mathbb{C}$ with low energy ($\|\mathbf{y}\|_2^2$ small) whose ($\sqrt{p(\boldsymbol{\eta})}$ weighted) Fourier transform is close to the frequency $e^{-2\pi i \mathbf{x}_j^T \boldsymbol{\eta}}$ at each data point.



- $\mathbf{y}$ reconstructs frequency $\boldsymbol{\eta}$ from other frequencies. The easier it is to reconstruct, the less important it is to sample.

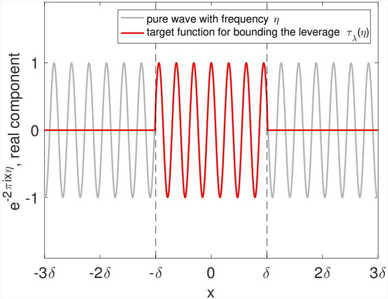
Assume data points are 1-dimensional and bounded: $x_1, ..., x_n \in [-\delta, \delta]$. One possibility is to choose $\mathbf{y}$ with ($\sqrt{p(\eta)}$ weighted) Fourier transform equal to $e^{-2\pi i x \eta}$ for all $x \in [-\delta, \delta]$.

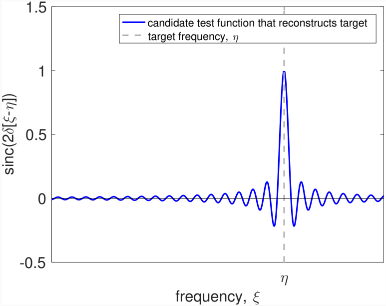
Assume data points are 1-dimensional and bounded: $x_1, ..., x_n \in [-\delta, \delta]$. One possibility is to choose $\mathbf{y}$ with ($\sqrt{p(\eta)}$ weighted) Fourier transform equal to $e^{-2\pi i x \eta}$ for all $x \in [-\delta, \delta]$.

- Achieved by the shifted sinc function weighted by $1/\sqrt{p(\eta)}$.

Assume data points are 1-dimensional and bounded: $x_1, \ldots, x_n \in [-\delta, \delta]$. One possibility is to choose $\mathbf{y}$ with ($\sqrt{p(\eta)}$ weighted) Fourier transform equal to $e^{-2\pi i x \eta}$ for all $x \in [-\delta, \delta]$.

- Achieved by the shifted sinc function weighted by $1/\sqrt{p(\eta)}$.





Unfortunately the sinc function falls off too slowly.

IMPROVED TEST FUNCTION

Unfortunately the sinc function falls off too slowly.

- For the Gaussian kernel, the $\frac{1}{\sqrt{p(\eta)}} \approx e^{\eta^2/4}$ weighting, will grow faster than $sinc(2\delta\eta) = \frac{\sin(2\delta\eta)}{\eta}$ falls off. So $\|\mathbf{y}\|_2$ is unbounded.

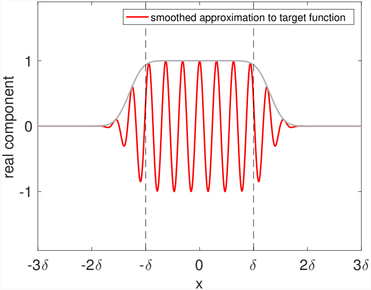
Unfortunately the sinc function falls off too slowly.

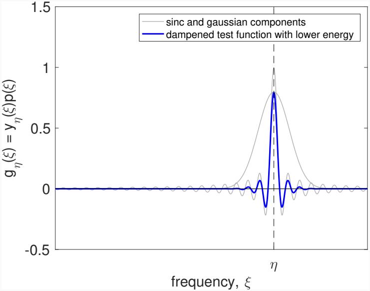
- For the Gaussian kernel, the $\frac{1}{\sqrt{p(\eta)}} \approx e^{\eta^2/4}$ weighting, will grow faster than $sinc(2\delta\eta) = \frac{\sin(2\delta\eta)}{\eta}$ falls off. So $\|\mathbf{y}\|_2$ is unbounded.
- **Solution:** Dampen the sinc by multiplying with a Gaussian, keeping Fourier transform nearly identical.

IMPROVED TEST FUNCTION

Unfortunately the sinc function falls off too slowly.

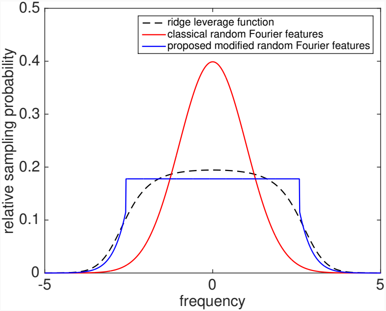
- For the Gaussian kernel, the $\frac{1}{\sqrt{p(\eta)}} \approx e^{\eta^2/4}$ weighting, will grow faster than $sinc(2\delta\eta) = \frac{\sin(2\delta\eta)}{\eta}$ falls off. So $\|\mathbf{y}\|_2$ is unbounded.
- **Solution:** Dampen the sinc by multiplying with a Gaussian, keeping Fourier transform nearly identical.



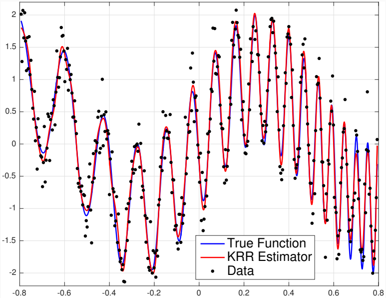


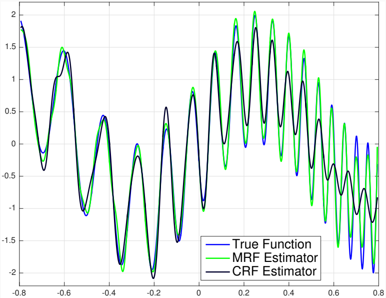
Upshot: easy to sample from approximate leverage distribution for the Gaussian kernel with $x_1, ..., x_n \in [-\delta, \delta]^d$:

$$\bar{\tau}_\lambda(\boldsymbol{\eta}) \begin{cases} \tilde{O}(\delta^d) \text{ when } \|\boldsymbol{\eta}\|_\infty \leq \sqrt{\log n/\lambda} \\ p(\boldsymbol{\eta}) = e^{-\|\boldsymbol{\eta}\|_2^2/2} \text{ otherwise.} \end{cases}$$



Example of approximating a synthetic 'wiggly function':





CRF = classic random Fourier features 'column norm' sampling,
MRF = our modified sampling distribution.

Questions?