IEOR E4403 - Data Mining

Columbia University - IEOR

Data Mining
Problems Spring 2025

Spring 2025

Author: Edward Lucyszyn
Professor: Krzysztof Choromanski

Contents

1 Problem 2: Performer model for the custom attention kernel

Let d be a strictly positive integer. We describe two Monte Carlo methods to approximate a polynomial attention kernel

$$\mathrm{K} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R},\ (\boldsymbol{x}, \boldsymbol{y}) \mapsto (\boldsymbol{x}^\top \boldsymbol{y})^n \tag{1}$$

in a lower-dimensional feature space, with $n \in \mathbb{N}$ such that $n \geq 1$. Both methods yield an unbiased estimator of K, but with different variance and computational trade-offs. These will be useful to approach any polynomial later in Problem 3.

1.1 Sum-decomposing approximation approach

1.1.1 Set up

Firstly, the following holds:

$$\mathrm{K}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\top \boldsymbol{y})^n \ = \ \Big(\sum_{i=1}^{d} x_i y_i\Big)^n = \sum_{1 \leq i_1, i_2, \ldots, i_n \leq d} \left(\prod_{t=1}^{n} x_{i_t}\, y_{i_t}\right). \tag{2}$$

The main idea of this approach relies on selecting some tuples $(i_t)_{1 \leq t \leq n} \in [\![1, d]\!]^n$ randomly, and then doing the summing the product of the queries and the keys only under these tuples. Choosing uniformly the tuples and applying this methods allows to approximate in expectation the desired kernel.

Let m be a strictly positive integer, and $\omega_1, \omega_2, \ldots, \omega_m \overset{\text{iid}}{\sim} \mathcal{U}([\![1, d]\!]^n)$ (uniform distribution). We define the following random feature map:

$$\begin{aligned} \phi_m^S : \mathbb{R}^d \to \mathbb{R}^m,\ \boldsymbol{x} \mapsto \phi_m^S(\boldsymbol{x}) &= \frac{1}{\sqrt{m}} d^{n/2} \left[\prod_{\ell_1 \in \omega_1} x_{\ell_1}, \prod_{\ell_2 \in \omega_2} x_{\ell_2}, \ldots, \prod_{\ell_m \in \omega_m} x_{\ell_m} \right]^\top \\ &= \frac{1}{\sqrt{m}} d^{n/2} \left(\prod_{\ell \in \omega_j} x_\ell \right)_{1 \leq j \leq m}. \end{aligned} \tag{3}$$

We also define the estimator:

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, \quad \widehat{\mathrm{K}_m^S}(\boldsymbol{x}, \boldsymbol{y}) = \phi_m^S(\boldsymbol{x})^\top \phi_m^S(\boldsymbol{y}). \tag{4}$$

Result 1 (Unbiasedness)**.** *For every integer $m \geq 1$, the estimator $\widehat{\mathrm{K}_m^S}$ is an unbiased estimator of* K*.*

Proof. By construction, and using the fact that the $(\omega_j)_{1 \leq j \leq m}$ are independent and identically distributed, we have:

$$\mathbb{E}\big[\phi_m^S(\boldsymbol{x})^\top \phi_m^S(\boldsymbol{y})\big] = \frac{d^n}{m} \sum_{j=1}^{m} \mathbb{E}\Big[\prod_{\ell_j \in \omega_j} x_{\ell_j} y_{\ell_j}\Big] = d^n \mathbb{E}_{\omega_1 \sim \mathcal{U}([\![1,d]\!]^n)}\Big[\prod_{\ell_1 \in \omega_1}^{n} x_{\ell_1} y_{\ell_1}\Big], \tag{5}$$

Because each n-tuple is drawn uniformly, every term in the full expansion of $(\boldsymbol{x}^\top \boldsymbol{y})^n$ described in (2) is equally likely. Thus the expectation yields

$$\mathbb{E}_{\omega_1 \sim \mathcal{U}([\![1,d]\!]^n)}\Big[\prod_{\ell_1 \in \omega_1}^{n} x_{\ell_1} y_{\ell_1}\Big] = \frac{1}{d^n} \sum_{i=(i_1, i_2, \ldots i_n) \in [\![1,d]\!]^n} \left(\prod_{1 \leq t \leq n}^{n} x_{i_t} y_{i_t}\right) = \frac{1}{d^n} (\boldsymbol{x}^\top \boldsymbol{y})^n, \tag{6}$$

and

$$\mathbb{E}\big[\widehat{\mathrm{K}_m^S}(\boldsymbol{x}, \boldsymbol{y})\big] = \mathbb{E}\big[\phi_m^S(\boldsymbol{x})^\top \phi_m^S(\boldsymbol{y})\big] = d^n \cdot \frac{1}{d^n} (\boldsymbol{x}^\top \boldsymbol{y})^n = \mathrm{K}(\boldsymbol{x}, \boldsymbol{y}).$$

□

Although $\widehat{\mathrm{K}_m^S}$ is unbiased, it incurs variance due to the sampling process.

1.1.2 Variance and Analysis

Result 2 (Variance of the estimator)**.** *For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, the mean square error (MSE) of the estimator $\widehat{\mathrm{K}_m^S}(\boldsymbol{x}, \boldsymbol{y})$ is:*

$$\mathrm{MSE}\Big[\widehat{\mathrm{K}_m^S}(\boldsymbol{x}, \boldsymbol{y})\Big] = \frac{1}{m}\Big(\mathrm{K}(\alpha(\boldsymbol{x}), \alpha(\boldsymbol{y}))d^n - \mathrm{K}(\boldsymbol{x}, \boldsymbol{y})\Big), \tag{7}$$

with $\alpha : \mathbb{R}^d \to \mathbb{R}^d$, $\boldsymbol{x} = (x_i)_{1\leq i \leq d} \mapsto \alpha(\boldsymbol{x}) = (x_i^2)_{1 \leq i \leq d}$.

Proof. By bias variance decomposition and using the fact the $(\omega_j)_{1\leq j \leq m}$ are iid, we have:

$$\begin{aligned} \mathrm{MSE}\left[\widehat{\mathrm{K}_m^S}(\boldsymbol{x}, \boldsymbol{y})\right] &= \mathrm{Var}\left[\widehat{\mathrm{K}_m^S}(\boldsymbol{x}, \boldsymbol{y})\right] \\ &= \mathrm{Var}\left[\frac{d^n}{m}\sum_{j=1}^{m}\prod_{\ell_j \in \omega_j}^{n} x_{\ell_j} y_{\ell_j}\right] \\ &= \frac{d^{2n}}{m}\,\mathrm{Var}\Big[\prod_{\ell_1 \in \omega_1}^{n} x_{\ell_1} y_{\ell_1}\Big] \\ &= \frac{d^{2n}}{m}\Big(\mathbb{E}\big[\prod_{\ell_1 \in \omega_1}^{n} x_{\ell_1}^2 y_{\ell_1}^2\big] - \mathbb{E}\big[\prod_{\ell_1 \in \omega_1}^{n} x_{\ell_1} y_{\ell_1}\big]^2\Big). \end{aligned} \tag{8}$$

Using (6), we obtain:

$$\mathbb{E}\big[\prod_{\ell_1 \in \omega_1}^{n} x_{\ell_1}^2 y_{\ell_1}^2\big] = \frac{1}{d^n}\mathrm{K}(\alpha(\boldsymbol{x}), \alpha(\boldsymbol{y})). \tag{9}$$

Thus,

$$\begin{aligned} \mathrm{MSE}\left[\widehat{\mathrm{K}_m^S}(\boldsymbol{x}, \boldsymbol{y})\right] &= \frac{d^{2n}}{m}\Big(\frac{1}{d^n}\mathrm{K}(\alpha(\boldsymbol{x}), \alpha(\boldsymbol{y})) - \frac{1}{d^{2n}}\mathrm{K}(\boldsymbol{x}, \boldsymbol{y})\Big) \\ &= \frac{1}{m}\Big(\mathrm{K}(\alpha(\boldsymbol{x}), \alpha(\boldsymbol{y}))d^n - \mathrm{K}(\boldsymbol{x}, \boldsymbol{y})\Big). \end{aligned}$$

□

The term d^n growths exponentially with n, which can lead a very bad approximation for high value of n. In the Problem 3, we will want to approximate sequences of polynomial functions of increasing degrees. Therefore, we want to lower the value of the variance of each polynomial approximation the best as possible. This is the reason why later we will have the second approach that allows us to remove this exponential dependency on n. Also, notice that the $\mathrm{K}(\alpha(\boldsymbol{x}), \alpha(\boldsymbol{y}))$ also depends on n. If we have data with components in norms above 1, we will often have $\mathrm{K}(\alpha(\boldsymbol{x}), \alpha(\boldsymbol{y})) >> \mathrm{K}(\boldsymbol{x}, \boldsymbol{y})$.

1.1.3 Absolute and Relative Errors

For purpose of comparisons, we compute the absolute error and the relative error of $\widehat{\mathrm{K}_m^S}(\boldsymbol{x}, \boldsymbol{y})$.

Result 3 (Expectation of relative and absolute errors)**.** *For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, the expected absolute error of the estimator is bounded as*

$$\mathbb{E}\Big[\Big|\widehat{\mathrm{K}_m^S}(\boldsymbol{x}, \boldsymbol{y}) - \mathrm{K}(\boldsymbol{x}, \boldsymbol{y})\Big|\Big] \;\leq\; \sqrt{\frac{1}{m}\Big(\mathrm{K}(\alpha(\boldsymbol{x}), \alpha(\boldsymbol{y}))\,d^n - \mathrm{K}(\boldsymbol{x}, \boldsymbol{y})\Big)}. \tag{10}$$

In particular, provided that $\mathrm{K}(\boldsymbol{x}, \boldsymbol{y}) \neq 0$, the relative error is bounded by

$$\frac{\mathbb{E}\Big[\Big|\widehat{\mathrm{K}_m^S}(\boldsymbol{x}, \boldsymbol{y}) - \mathrm{K}(\boldsymbol{x}, \boldsymbol{y})\Big|\Big]}{|\,\mathrm{K}(\boldsymbol{x}, \boldsymbol{y})|} \;\leq\; \frac{1}{|\,\mathrm{K}(\boldsymbol{x}, \boldsymbol{y})|}\sqrt{\frac{1}{m}\Big(\mathrm{K}(\alpha(\boldsymbol{x}), \alpha(\boldsymbol{y}))\,d^n - \mathrm{K}(\boldsymbol{x}, \boldsymbol{y})\Big)}. \tag{11}$$

Proof. Since $\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y})$ is an unbiased estimator of $\mathrm{K}(\boldsymbol{x},\boldsymbol{y})$, by definition of the mean squared error (MSE) we have

$$\mathrm{MSE}\Big[\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y})\Big] \;=\; \mathbb{E}\Big[\Big(\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y}) - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\Big)^2\Big] \;=\; \frac{1}{m}\Big(\mathrm{K}(\alpha(\boldsymbol{x}),\alpha(\boldsymbol{y}))\,d^n - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\Big).$$

By an application of Jensen's inequality, we know that

$$\mathbb{E}\Big[\Big|\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y}) - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\Big|\Big] \;\le\; \sqrt{\mathbb{E}\Big[\Big(\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y}) - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\Big)^2\Big]} \;=\; \sqrt{\mathrm{MSE}\Big[\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y})\Big]}. \tag{12}$$

Substituting the expression for the MSE, we obtain the stated bound for the expected absolute error:

$$\mathbb{E}\Big[\Big|\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y}) - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\Big|\Big] \;\le\; \sqrt{\frac{1}{m}\Big(\mathrm{K}(\alpha(\boldsymbol{x}),\alpha(\boldsymbol{y}))\,d^n - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\Big)}.$$

Dividing both sides by $\big|\,\mathrm{K}(\boldsymbol{x},\boldsymbol{y})\big|$ (when $\mathrm{K}(\boldsymbol{x},\boldsymbol{y}) \neq 0$) leads to the bound on the relative error:

$$\frac{\mathbb{E}\Big[\Big|\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y}) - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\Big|\Big]}{\big|\,\mathrm{K}(\boldsymbol{x},\boldsymbol{y})\big|} \;\le\; \frac{1}{\big|\,\mathrm{K}(\boldsymbol{x},\boldsymbol{y})\big|}\sqrt{\frac{1}{m}\Big(\mathrm{K}(\alpha(\boldsymbol{x}),\alpha(\boldsymbol{y}))\,d^n - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\Big)}.$$

This completes the proof. □

Result 4 (High-Probability Error Bounds)**.** *Let $\epsilon > 0$ be a tolerance. Then for any $\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^d$, the absolute error bound of the $\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y})$ satisfies:*

$$\mathbb{P}\Big(|\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y}) - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})| < \epsilon\Big) \;\ge\; 1 \;-\; \frac{1}{\epsilon^2 m}\Big(\mathrm{K}(\alpha(\boldsymbol{x}),\alpha(\boldsymbol{y}))\,d^n - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\Big). \tag{13}$$

Also, provided that $\mathrm{K}(\boldsymbol{x},\boldsymbol{y}) \neq 0$, the relative error bound satisfies:

$$\mathbb{P}\Big(|\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y}) - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})| < \epsilon\,|\,\mathrm{K}(\boldsymbol{x},\boldsymbol{y})|\Big) \;\ge\; 1 \;-\; \frac{1}{\epsilon^2\,|\,\mathrm{K}(\boldsymbol{x},\boldsymbol{y})|^2\,m}\Big(\mathrm{K}(\alpha(\boldsymbol{x}),\alpha(\boldsymbol{y}))\,d^n - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\Big). \tag{14}$$

Proof. Recall that from our previous result, the estimator $\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y})$ is unbiased and its mean squared error (MSE) is given by

$$\mathrm{MSE}\Big[\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y})\Big] = \mathrm{Var}\Big[\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y})\Big] = \frac{1}{m}\Big(\mathrm{K}(\alpha(\boldsymbol{x}),\alpha(\boldsymbol{y}))\,d^n - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\Big).$$

Using Chebyshev's inequality, for any $\epsilon > 0$, we have

$$\mathbb{P}\Big(|\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y}) - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})| \ge \epsilon\Big) \;\le\; \frac{\mathrm{Var}\Big[\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y})\Big]}{\epsilon^2} \;=\; \frac{1}{\epsilon^2 m}\Big(\mathrm{K}(\alpha(\boldsymbol{x}),\alpha(\boldsymbol{y}))\,d^n - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\Big). \tag{15}$$

Equivalently, this implies

$$\mathbb{P}\Big(|\widehat{\mathrm{K}^S_m}(\boldsymbol{x},\boldsymbol{y}) - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})| < \epsilon\Big) \;\ge\; 1 \;-\; \frac{1}{\epsilon^2 m}\Big(\mathrm{K}(\alpha(\boldsymbol{x}),\alpha(\boldsymbol{y}))\,d^n - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\Big).$$

For the relative error bound, assuming $\mathrm{K}(\boldsymbol{x},\boldsymbol{y}) \neq 0$, we apply Chebyshev's inequality again with $\epsilon|\,\mathrm{K}(\boldsymbol{x},\boldsymbol{y})|$ in place of ϵ and we obtain the desired result. □

1.2 Tensor and Hashing Functions Approach

1.2.1 Explanation

This approach comes from the work of Pham and Pagh in [1]. The general procedure is the following. Let

$$h : [\![1, d]\!] \to [\![1, m]\!] \quad \text{and} \quad \xi : [\![1, d]\!] \to \{\pm 1\}$$

be two hashing functions chosen from 2-wise independent families. We define the random feature map

$$\phi^{(h,\xi)} : \mathbb{R}^d \to \mathbb{R}^m, \quad (\phi^{(h,\xi)}(\boldsymbol{x}))_{1 \le j \le m} = \left(\sum_{i:\, h(i)=j} \xi(i)\, x_i \right)_{1 \le j \le m}. \tag{16}$$

Introducing the tensor notations, we have:

$$\mathrm{K}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\top \boldsymbol{y})^n = \langle \boldsymbol{x}^{\otimes n},\, \boldsymbol{y}^{\otimes n} \rangle. \tag{17}$$

To avoid computing $\boldsymbol{x}^{\otimes n}$ explicitly, which is of size d^n, we compute n independent Count Sketches:

$$\forall j \in [\![1, n]\!], \quad s^{(l)}(\boldsymbol{x}) = \phi^{(h_l, \xi_l)}(\boldsymbol{x}), \tag{18}$$

and define the Tensor Sketch feature map by

$$\phi_m^T(\boldsymbol{x}) = \mathrm{IFFT}\left(\bigodot_{i=1}^{n} \mathrm{FFT}\Big(s^{(l)}(\boldsymbol{x}) \Big) \right), \tag{19}$$

where $\bigodot_{i=1}^n$ stands for the Hadamard product, which is the component wise product. We also define the estimator:

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, \quad \widehat{\mathrm{K}_m^T}(\boldsymbol{x}, \boldsymbol{y}) = \phi_m^T(\boldsymbol{x})^\top \phi_m^T(\boldsymbol{y}). \tag{20}$$

Result 5 (Unbiasedness and mean square error)**.** *For every integer $m \ge 1$, the estimator $\widehat{\mathrm{K}_m^T}$ is an unbiased estimator of $\mathrm{K}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\top \boldsymbol{y})^n$ and the mean square error (MSE) satisfies the inequality:*

$$\mathrm{MSE}\Big[\widehat{\mathrm{K}_m^T}(\boldsymbol{x}, \boldsymbol{y}) \Big] \le \frac{1}{m} \Big[(\boldsymbol{x}^\top \boldsymbol{y})^{2n} + \|\boldsymbol{x}\|^{2n} \, \|\boldsymbol{y}\|^{2n} \Big]. \tag{21}$$

Proof. See [1]. □

1.2.2 Absolute and Relative Errors

Result 6 (Expectation of relative and absolute errors)**.** *For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, the expected absolute error of the Tensor Sketch estimator satisfies:*

$$\mathbb{E}\Big[\big| \widehat{\mathrm{K}_m^T}(\boldsymbol{x}, \boldsymbol{y}) - \mathrm{K}(\boldsymbol{x}, \boldsymbol{y}) \big| \Big] \le \sqrt{\frac{1}{m} \Big[(\boldsymbol{x}^\top \boldsymbol{y})^{2n} + \|\boldsymbol{x}\|^{2n} \, \|\boldsymbol{y}\|^{2n} \Big]}. \tag{22}$$

Moreover, if $\mathrm{K}(\boldsymbol{x}, \boldsymbol{y}) \neq 0$, define

$$\cos \theta_{xy} = \frac{\boldsymbol{x}^\top \boldsymbol{y}}{\|\boldsymbol{x}\| \, \|\boldsymbol{y}\|},$$

then, the expected relative error is bounded by:

$$\frac{\mathbb{E}\big[\big| \widehat{\mathrm{K}_m^T}(\boldsymbol{x}, \boldsymbol{y}) - \mathrm{K}(\boldsymbol{x}, \boldsymbol{y}) \big| \big]}{\big| \mathrm{K}(\boldsymbol{x}, \boldsymbol{y}) \big|} \le \sqrt{\frac{1}{m} \Big[1 + \cos^{-2n} \theta_{xy} \Big]}. \tag{23}$$

Proof. Since $\widehat{\mathrm{K}_m^T}$ is unbiased, its mean squared error equals its variance, which satisfies

$$\mathrm{MSE}\Big[\widehat{\mathrm{K}_m^T}(\boldsymbol{x},\boldsymbol{y})\Big] = \mathrm{Var}\Big[\widehat{\mathrm{K}_m^T}(\boldsymbol{x},\boldsymbol{y})\Big] \leq \frac{1}{m}\Big[(\boldsymbol{x}^\top\boldsymbol{y})^{2n} + \|\boldsymbol{x}\|^{2n}\,\|\boldsymbol{y}\|^{2n}\Big].$$

Jensen's inequality then gives the bound on the expected absolute error, and dividing by $\big|\mathrm{K}(\boldsymbol{x},\boldsymbol{y})\big|$ yields the relative-error bound:

$$\begin{aligned}\frac{\mathbb{E}\big[\big|\widehat{\mathrm{K}_m^T}(\boldsymbol{x},\boldsymbol{y}) - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\big|\big]}{\big|\mathrm{K}(\boldsymbol{x},\boldsymbol{y})\big|} &\leq \frac{1}{\big|\mathrm{K}(\boldsymbol{x},\boldsymbol{y})\big|}\sqrt{\frac{1}{m}\Big[(\boldsymbol{x}^\top\boldsymbol{y})^{2n} + \|\boldsymbol{x}\|^{2n}\,\|\boldsymbol{y}\|^{2n}\Big]} \\ &= \sqrt{\frac{1}{m}\Big[1 + \cos^{-2n}\theta_{xy}\Big]}.\end{aligned} \tag{24}$$

□

Result 7 (Probability error bounds)**.** *Let $\epsilon > 0$. Then for any $\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^d$,*

$$\mathbb{P}\Big(\big|\widehat{\mathrm{K}_m^T}(\boldsymbol{x},\boldsymbol{y}) - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\big| < \epsilon\Big) \;\geq\; 1 \;-\; \frac{(\boldsymbol{x}^\top\boldsymbol{y})^{2n} \;+\; \|\boldsymbol{x}\|^{2n}\,\|\boldsymbol{y}\|^{2n}}{\epsilon^2\, m}.$$

Moreover, if $\mathrm{K}(\boldsymbol{x},\boldsymbol{y}) \neq 0$*,*

$$\mathbb{P}\Big(\big|\widehat{\mathrm{K}_m^T}(\boldsymbol{x},\boldsymbol{y}) - \mathrm{K}(\boldsymbol{x},\boldsymbol{y})\big| < \epsilon\,\big|\mathrm{K}(\boldsymbol{x},\boldsymbol{y})\big|\Big) \geq 1 \;-\; \frac{1 + \cos^{-2n}\theta_{xy}}{\epsilon^2\, m}.$$

Proof. Apply Chebyshev's inequality to the variance bound obtained in (21), first with threshold ϵ for the absolute-error bound, then with threshold $\epsilon\,|\,\mathrm{K}(\boldsymbol{x},\boldsymbol{y})|$ and using $|\,\mathrm{K}(\boldsymbol{x},\boldsymbol{y})|^2 = (\boldsymbol{x}^\top\boldsymbol{y})^{2n}$ to obtain the relative-error bound. □

1.3 Efficiency

Let L be the sequence length, d the input dimension, n the polynomial degree, and m the number of random features. We compare the cost of forming the full $L \times L$ with the different approaches.

For the exact polynomial kernel, computing $\big(\boldsymbol{x}_i^\top\boldsymbol{x}_j\big)^n$ for all (i,j) requires $O(d)$ per dot–product and $O(\log(n))$ for the exponentiation (using divide and conquer method for example), for a total of complexity of $O\big(L^2\, d\log(n)\big)$.

For the Sum–decomposing Monte Carlo (Section 1), each $\phi_m^S(\boldsymbol{x}) \in \mathbb{R}^m$ costs $O(m\,n)$ multiplications, so for L inputs give a complexity of $O\big(L\,m\,n\big)$. However, computing the attention matrix costs $O\big(L^2\,m\big)$. The overall complexity is $O\big(L\,m\,n + L^2\,m\big)$.

For the Tensor Sketch (Section 1.2, we have the following:

- n hash/sketch passes: $O(n\,d)$ per vector, total $O(L\,n\,d)$.
- n FFTs of length m: $O(n\,m\log m)$ per vector, total $O(L\,n\,m\log m)$.
- one IFFT of length m per vector: $O(m\log m)$, total $O(L\,m\log m)$.

The multiplication of the matrix is the same as in the first approach. Thus the overall complexity is: $O\big(L^2\,m + L\,n\,d + L\,n\,m\log m + L\,m\log m\big)$.

In contrast to the $O(L^2 d\log(n))$ exact cost, both Monte Carlo methods reduce the dependence on $d\log(n)$ by a dependence on m (and $n\log m$ for Tensor Sketch). When $m \ll d$ and n moderate, this yields substantial computational savings.

2 Problem 3: Linearization of the Feedforward Layers in NNs

Consider a feedforward layer of the form

$$\mathbf{y} \;=\; f(\boldsymbol{W}\mathbf{x}), \tag{25}$$

where $\boldsymbol{W} \in \mathbb{R}^{d\times d}$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. Assume f is a continuous function. We propose two variants: a deterministic (biased) polynomial approximation and an unbiased Monte Carlo variant. We note that if f is the GELU function, we have:

$$\forall x \in \mathbb{R}, \quad f(x) = \frac{x}{2}\left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right]. \tag{26}$$

2.1 Bernstein Polynomial Approximation Approach

2.1.1 Set up

From viewing our data from a physical point of view, we can assume that our data is bounded and thus lives in a compact set $\mathcal{X} \subset \mathbb{R}^d$. Also,

$$\forall \boldsymbol{x} \in \mathbb{R}^d, \quad f(\boldsymbol{W}\boldsymbol{x}) = (f(\boldsymbol{w}_i\boldsymbol{x}))_{1\le i\le d}\,, \tag{27}$$

where the $(\boldsymbol{w}_i)_{1\le i\le d} \in (\mathbb{R}^{1\times d})^d$ stand for the rows of $\boldsymbol{W}$. Since f is continuous, for each $i \in [\![1, d]\!], \mathcal{X} \to \mathbb{R}$, $\boldsymbol{x} \mapsto f(\boldsymbol{w}_i\boldsymbol{x})$ is continuous on a compact set. Thus, this function attains its extrema, and we can define

$$a_i \;=\; \min_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{w}_i\boldsymbol{x}, \quad b_i \;=\; \max_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{w}_i\boldsymbol{x}. \tag{28}$$

Hence, for all $\boldsymbol{x} \in \mathcal{X}$,

$$\boldsymbol{w}_i\boldsymbol{x} \in [a_i, b_i].$$

Now, we wish to approximate the scalar mapping $t \mapsto f(t)$ on the interval $[a_i, b_i]$ by a polynomial of degree n. To that end, we introduce the affine change of variable

$$\theta = \frac{t - a_i}{b_i - a_i}, \quad \theta \in [0, 1]. \tag{29}$$

On $[0, 1]$, if g is a continuous function, the classical Bernstein polynomials of degree n approximating g are

$$B_n(\theta) = \sum_{k=0}^{n} \binom{n}{k} \theta^k (1-\theta)^{n-k} g\left(\frac{k}{n}\right), \quad k = 0, 1, \ldots, n. \tag{30}$$

Pulling back to $[a_i, b_i]$, we define the Bernstein approximation of f by

$$\forall t \in [a_i, b_i], \quad B_{n,i}(t) \;=\; \sum_{k=0}^{n} f\Big(a_i + \frac{k}{n}(b_i - a_i)\Big)\binom{n}{k}\Big(\frac{t - a_i}{b_i - a_i}\Big)^k\Big(1 - \frac{t - a_i}{b_i - a_i}\Big)^{n-k}. \tag{31}$$

By construction, $B_{n,i}$ is a polynomial of degree at most n in t. Moreover, one may rewrite it in monomial form. Let $(\alpha_{n,i,k})_{1\le k\le n}$ be its coefficients on the canonical basis of $\mathbb{R}_n[X]$. We have

$$\forall t \in [a_i, b_i], \quad B_{n,i}(t) = \sum_{k=0}^{n} \alpha_{n,i,k} t^k. \tag{32}$$

Since the proof of the Weierstrass theorem asserts that the sequence $B_{n,i}[f]$ converges uniformly to f on $[a_i, b_i]$, we know that

$$\lim_{n\to\infty} \sup_{\boldsymbol{x}\in\mathcal{X}} |B_{n,i}(\boldsymbol{w}_i\boldsymbol{x}) - f(\boldsymbol{w}_i\boldsymbol{x})| = 0 \text{ with } B_{n,i}(\boldsymbol{w}_i\boldsymbol{x}) = \sum_{k=0}^{n} \alpha_{n,i,k}(\boldsymbol{w}_i\boldsymbol{x})^k. \tag{33}$$

From Problem 2 (see 1), we can have a linear unbiased approximation of $\boldsymbol{w}_i\boldsymbol{x}$. For every $k \in [\![1, n]\!]$ and for $m_k \geq 1$, we define $\phi_k : \mathbb{R}^d \mapsto \mathbb{R}^{m_k}$ such that

$$\forall \boldsymbol{x} \in \mathcal{X}, \quad \mathbb{E}\left[\phi_k(\boldsymbol{w}_i)^\top \phi_k(\boldsymbol{x})\right] = (\boldsymbol{w}_i\boldsymbol{x})^k. \tag{34}$$

Adjusting the m_k according to k is justified by Problem 2, where we saw that the MSE increases when the degree of the polynomial kernel increases and decreases when the dimension of the output space increases. Having small m_k for small values of k allows the model to have bigger values where it counts, *i.e.*, for large values of k, for example. Thus, we have:

$$\forall \boldsymbol{x} \in \mathcal{X}, \quad \mathbb{E}\left[\sum_{k=0}^{n} \alpha_{n,i,k}\phi_k(\boldsymbol{w}_i)^\top \phi_k(\boldsymbol{x})\right] = \sum_{k=0}^{n} \alpha_{n,i,k}\mathbb{E}\left[\phi_k(\boldsymbol{w}_i)^\top \phi_k(\boldsymbol{x})\right] = B_{n,i}(\boldsymbol{w}_i\boldsymbol{x}). \tag{35}$$

Now, we can define for all $n \geq 1, i \in [\![1, d]\!]$:

$$\Phi_i^n(\boldsymbol{W}) = \left[\alpha_{n,i,0},\ \alpha_{n,i,1}\phi_1(\boldsymbol{w}_i)^\top,\ \alpha_{n,i,2}\phi_2(\boldsymbol{w}_i)^\top,\ \ldots,\ \alpha_{n,i,n}\phi_n(\boldsymbol{w}_i)^\top\right] \in \mathbb{R}^{1\times 1+\sum_{k=1}^n m_k}, \tag{36}$$

and

$$\Phi^n(\boldsymbol{W}) = \begin{bmatrix}\Phi_1^n(\boldsymbol{W})\\ \Phi_2^n(\boldsymbol{W})\\ \vdots\\ \Phi_d^n(\boldsymbol{W})\end{bmatrix} \in \mathbb{R}^{d\times(1+\sum_{k=1}^n m_k)}; \quad \forall \boldsymbol{x} \in \mathcal{X},\ \Psi^n(\boldsymbol{x}) = \begin{bmatrix}1\\ \phi_1(\boldsymbol{x})\\ \vdots\\ \phi_n(\boldsymbol{x})\end{bmatrix} \in \mathbb{R}^{(1+\sum_{k=1}^n m_k)}. \tag{37}$$

Result 8. *For all $n \geq 1$, the following holds:*

$$\Phi^n(\boldsymbol{W})\Psi^n(\boldsymbol{x}) = \begin{bmatrix}B_{n,1}(\boldsymbol{w}_1\boldsymbol{x})\\ B_{n,2}(\boldsymbol{w}_2\boldsymbol{x})\\ \vdots\\ B_{n,d}(\boldsymbol{w}_d\boldsymbol{x})\end{bmatrix}. \tag{38}$$

Moreover,

$$\lim_{n\to\infty} ||\mathbb{E}\left[\Phi^n(\boldsymbol{W})\Psi^n(\boldsymbol{x})\right] - f(\boldsymbol{W}\boldsymbol{x})||_{\mathcal{X},\infty} = 0. \tag{39}$$

Proof. For 38, see the explanation above. For 39, we note that:

$$\forall n \geq 1, \quad ||\mathbb{E}\left[\Phi^n(\boldsymbol{W})\Psi^n(\boldsymbol{x})\right] - f(\boldsymbol{W}\boldsymbol{x})||_{\mathcal{X},\infty} = \max_{1\leq i\leq d}\sup_{\boldsymbol{x}\in\mathcal{X}} |\mathbb{E}\left[\Phi_i^n(\boldsymbol{W})\Psi^n(\boldsymbol{x})\right] - f(\boldsymbol{W}\boldsymbol{x})_i| \tag{40}$$

$$= \max_{1\leq i\leq d}\sup_{\boldsymbol{x}\in\mathcal{X}} |B_{n,i}(\boldsymbol{w}_i\boldsymbol{x}) - f(\boldsymbol{w}_i\boldsymbol{x})|. \tag{41}$$

By taking the limit and using 33, we obtain the desired result. □

Since the equality is asymptotic, we cannot have a fully unbiased approximation. Each time we increase the value of n, we reduce the supremum of the error of the Bernstein polynomials. It is important to note that this error is deterministic and arises due to the approximation of f. The stochastic aspect here lies in the work to linearize the exponent in the Bernstein polynomials.

Before discussing the variance or the mean square error of this model, it is important to note that this approach can be applied to any continuous function f and has many parameters. The first one, n, is the degree of the "constant" error that we want. Increasing n will result in high-exponent terms to approximate, which can lead to higher variance. Furthermore, it will increase the dimensions of the output spaces for Φ and Ψ. The other parameters that we can tune are the $(m_k)_{1\leq k\leq n}$. As mentioned before, if we want to minimize the output dimension of Ψ, we can adjust these to increase the m_k as k increases. According to Problem 2 (at least for the two types of ϕ functions we chose), this will allow the model to control the variance of the estimation.

2.1.2 Error Study and Parameter Selection

Before examining the variance introduced by the stochastic approach, we first quantify the approximation error from the Bernstein polynomial.

Result 9 (Bernstein Approximation Error)**.** *For each $i \in [\![1, d]\!]$, if f is a $\mathcal{C}^2$ function and $0 \in [a_i, b_i]$, then for all $t \in [a_i, b_i]$, the approximation error satisfies:*

$$|B_{n,i}(t) - f(t)| \leq \frac{(b_i - a_i)^2}{8n} \max_{\xi \in [a_i, b_i]} |f''(\xi)|. \tag{42}$$

Furthermore, when f is the GELU function, we obtain:

$$|B_{n,i}(t) - f(t)| \leq \frac{(b_i - a_i)^2}{8n} \max_{\xi \in [a_i, b_i]} \left| \frac{1}{\sqrt{2\pi}} e^{-\xi^2/2} \left(1 - \frac{\xi^2}{2}\right) \right| \tag{43}$$

Proof. Following [2], for any $\mathcal{C}^2$ function g approximated by Bernstein polynomials on $[0, 1]$, we have:

$$\forall x \in [0, 1], \quad |B_n(x) - g(x)| \leq \frac{1}{8n} \sup_{\xi \in [0,1]} |g''(\xi)|. \tag{44}$$

Applying the transformation $t = a_i + x(b_i - a_i)$ to $[a_i, b_i]$ yields (42). For the GELU function, using (26), we obtain:

$$\forall x \in \mathbb{R}, \quad f''(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \left(1 - \frac{x^2}{2}\right). \tag{45}$$

Substituting this expression for f'' completes the proof. □

Remark The maximum can be computed numerically for any given interval. For non-$\mathcal{C}^2$ functions, alternative formulas like those in [3] may be used. When $0 \notin [a_i, b_i]$, the interval can be adjusted by either decreasing a_i or increasing b_i.

For the variance analysis, we derive exact expressions using the GELU-specific coefficients.

Result 10 (Total Approximation Error)**.** *For any $n \geq 1$, the mean squared error decomposes as:*

$$\mathbb{E}\left[\|\Phi^n(\boldsymbol{W})\Psi^n(\boldsymbol{x}) - f(\boldsymbol{W}\boldsymbol{x})\|^2\right] \leq \sum_{i=1}^{d} \left[\frac{C_i^2}{n^2} + \sum_{k=1}^{n} \frac{\alpha_{n,i,k}^2}{m_k} \left((\boldsymbol{w}_i \boldsymbol{x})^{2k} + \|\boldsymbol{w}_i\|^{2k} \|\boldsymbol{x}\|^{2k} \right) \right], \tag{46}$$

where $C_i = \dfrac{(b_i - a_i)^2}{8\sqrt{2\pi}} \sup\limits_{\xi \in [a_i, b_i]} e^{-\xi^2/2} |1 - \xi^2/2|$.

Proof. We employ a bias-variance decomposition. The bias is bounded using (43). For the variance term, leveraging the independence of feature maps across degrees, we have for each $1 \leq i \leq d$:

$$\mathrm{Var}\left(\sum_{k=1}^{n} \alpha_{n,i,k} \phi_k(\boldsymbol{w}_i)^\top \phi_k(\boldsymbol{x}) \right) = \sum_{k=1}^{n} \alpha_{n,i,k}^2 \,\mathrm{Var}\left(\phi_k(\boldsymbol{w}_i)^\top \phi_k(\boldsymbol{x}) \right).$$

Assuming the ϕ mappings are generated via the tensor approach described in (1.2), we bound the variance term as:

$$\forall \boldsymbol{x} \in \mathcal{X}, \quad \mathrm{Var}\left(\phi_k(\boldsymbol{w}_i)^\top \phi_k(\boldsymbol{x}) \right) \leq \frac{1}{m_k} \left((\boldsymbol{w}_i \boldsymbol{x})^{2k} + \|\boldsymbol{w}_i\|^{2k} \|\boldsymbol{x}\|^{2k} \right). \tag{47}$$

The final result follows by summing over all i. □

We present the following error decomposition:

$$\mathcal{E}(n,\{m_k\}) = \underbrace{\frac{d}{64n^2}\max_{1\le i\le d}(b_i-a_i)^4\|f''\|_\infty^2}_{\text{Bias}^2} + \underbrace{\sum_{i=1}^{d}\sum_{k=1}^{n}\frac{\alpha_{n,i,k}^2}{m_k}\left((\boldsymbol{w}_i\boldsymbol{x})^{2k}+\|\boldsymbol{w}_i\|^{2k}\|\boldsymbol{x}\|^{2k}\right)}_{\text{Variance}}. \tag{48}$$

We propose the following parameter selection strategy. For a target error $\epsilon > 0$, the parameters should scale as:

$$n \sim \epsilon^{-1/2}, \tag{49}$$

$$m_k \sim \frac{nd}{\epsilon}\cdot\alpha_{n,i,k}^2\max_{1\le i\le d}\|\boldsymbol{w}_i\|^{2k}\sup_{\boldsymbol{x}\in\mathcal{X}}\|\boldsymbol{x}\|^{2k}. \tag{50}$$

Algorithm 1 implements this procedure.

Algorithm 1 Adaptive Parameter Selection

```
Output: Parameters $(n, \{m_k\}_{k=1}^n)$
Estimate interval bounds:
    $a_i \leftarrow \min_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{w}_i\boldsymbol{x}, \quad b_i \leftarrow \max_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{w}_i\boldsymbol{x} \quad \forall i \in \{1,\dots,d\}$
Compute Bernstein coefficients $\alpha_{n,i,k}$ for $i \in \{1,\dots,d\}$, $k \in \{0,\dots,n\}$
Set polynomial degree:
    $n \leftarrow \left\lceil \frac{\sqrt{d}\max_i(b_i-a_i)^2\|f''\|_\infty}{8\epsilon^{1/2}} \right\rceil$
for $k = 1$ to $n$ do
    Set feature dimension:
        $m_k \leftarrow \left\lceil \frac{nd}{\epsilon}\max_{1\le i\le d}\alpha_{n,i,k}^2\max_{1\le i\le d}\|\boldsymbol{w}_i\|^{2k}\sup_{\boldsymbol{x}\in\mathcal{X}}\|\boldsymbol{x}\|^{2k} \right\rceil$
end for
return $(n, \{m_k\}_{k=1}^n)$
```

Result 11 (Complexity Bound)**.** *The total dimension $M = \sum_{k=1}^n m_k$ satisfies:*

$$M = \mathcal{O}\left(\epsilon^{-3/2}\cdot d^{3/2}\cdot\max_i(b_i-a_i)^2\cdot\sum_{k=1}^{n}(\max_i\alpha_{n,i,k}^2\max_{1\le i\le d}\|\boldsymbol{w}_i\|^{2k}\sup_{\boldsymbol{x}\in\mathcal{X}}\|\boldsymbol{x}\|^{2k})\right).$$

Proof. Clear from Algorithm 1. □

This bound guarantees the error control for the Bernstein approximation of any $\mathcal{C}^2$ function. However, we note that the dimensionality introduced by $\{m_k\}_k$ may become significantly large, increasing computational costs too much. We now examine an unbiased approximation alternative.

2.2 Unbiased Taylor–composition approach

2.2.1 Explanation

Assume the activation f admits an absolutely convergent Taylor expansion on an open interval containing every $\boldsymbol{w}_i\boldsymbol{x}$ (*i.e.*, $[\min_{1\le i\le d} a_i, \max_{1\le i\le d} b_i] \subseteq I$) that can occur:

$$\forall z \in I, \qquad f(z) = \sum_{k=0}^{\infty} a_k\, z^k, \qquad a_k = \frac{f^{(k)}(0)}{k!}. \tag{51}$$

If f is the GELU function, we have

$$\forall x \in \mathbb{R}, \quad f(x) = \frac{x}{2} + \frac{x}{2}\operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) = \frac{x}{2} + \frac{x}{\sqrt{2\pi}}\sum_{k=0}^{\infty}\frac{(-1)^k}{2k+1}\frac{x^{2k+1}}{2^k k!}. \tag{52}$$

In this case, since the first term is of degree one, we can keep it, and it will be linear. The goal is to have an unbiased approximation of $\frac{1}{\sqrt{2\pi}}\sum_{k=0}^{\infty}\frac{(-1)^k}{2k+1}\frac{(\boldsymbol{w}_i\boldsymbol{x})^{2(n+1)}}{2^n n!}$.

Now, let's assume that we want an unbiased approximation of any

$$f : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}, \quad (\boldsymbol{w}, \boldsymbol{x}) \mapsto f(\boldsymbol{w}\boldsymbol{x}) = \sum_{k=0}^{\infty} a_k (\boldsymbol{w}\boldsymbol{x})^k \tag{53}$$

such that,

$$A = \sum_{k=0}^{\infty} |a_k| < \infty. \tag{54}$$

For each output coordinate i, we keep the random feature maps ϕ_k introduced previously, so that $\mathbb{E}[\phi_k(\boldsymbol{w}_i)^\top \phi_k(\boldsymbol{x})] = (\boldsymbol{w}_i\boldsymbol{x})^k$. The key idea is to turn the infinite series into a single random term through the composition method: sample a random index K with a probability mass function (pmf) $p = \{p_k\}_{k\geq 0}$.

Let $K \sim p$ be independent of all feature maps. We also define ϕ_0 as the constant 1 function. Define, for every $i \in \{1, \ldots, d\}$,

$$\widehat{f}_i(\boldsymbol{x}) \;=\; \frac{a_K}{p_K}\,\phi_K(\boldsymbol{w}_i)^\top \phi_K(\boldsymbol{x}). \tag{55}$$

Since $\mathbb{E}[\phi_K(\boldsymbol{w}_i)^\top \phi_K(\boldsymbol{x}) \mid K] = (\boldsymbol{w}_i\boldsymbol{x})^K$, we have the unbiasedness property

$$\mathbb{E}\big[\widehat{f}_i(\boldsymbol{x})\big] \;=\; \sum_{k=0}^{\infty} p_k\,\frac{a_k}{p_k}\,\mathbb{E}[\phi_K(\boldsymbol{w}_i)^\top \phi_K(\boldsymbol{x})] \;=\; \sum_{k=0}^{\infty}\frac{a_k}{p_k}\,p_k\,(\boldsymbol{w}_i\boldsymbol{x})^k \;=\; f\big(\boldsymbol{w}_i\boldsymbol{x}\big). \tag{56}$$

Stacking the d coordinates gives an estimator $\widehat{f}(\boldsymbol{W}\boldsymbol{x}) \in \mathbb{R}^d$ that is exactly unbiased.

For the choice of the sampling distribution, a practical option is to take $K + 1 \sim \operatorname{Geo}(\rho)$, *i.e.*,

$$\forall k \in \mathbb{N}, \quad p_k = (1-\rho)^k \rho,$$

where $0 < \rho < 1$. We thus have $\mathbb{E}[K] = \frac{1}{\rho} + 1$. For the choice of ρ, we can try to take the value of K where we have the best approximation (least variance) for $(\boldsymbol{w}_i\boldsymbol{x})^k$. Other distributions, such as the Poisson, can also be chosen.

2.2.2 Variance bound

Result 12 (Variance of the estimator)**.** *For every $i \in [\![1, d]\!]$, and for every $\boldsymbol{x} \in \mathcal{X}$, the following holds:*

$$\operatorname{Var}(\widehat{f}_i(\boldsymbol{x})) \;=\; \sum_{k=1}^{\infty}\frac{a_k^2}{p_k}\operatorname{Var}\Big(\phi_k(\boldsymbol{w}_i)^\top \phi_k(\boldsymbol{x})\Big) \;+\; \sum_{k=0}^{\infty}\frac{a_k^2}{p_k}(\boldsymbol{w}_i\boldsymbol{x})^{2k} \;-\; f(\boldsymbol{w}_i\boldsymbol{x})^2. \tag{57}$$

Moreover, if the $(\phi_k)_k$ are generated using the tensor approach in Problem 2, we can derive an upper bound on the variance:

$$\operatorname{Var}(\widehat{f}_i(\boldsymbol{x})) \;\leq\; \frac{a_0^2}{p_0} + \sum_{k=1}^{\infty}\frac{a_k^2}{p_k}\Big[(\boldsymbol{w}_i\boldsymbol{x})^{2k} + \frac{(\boldsymbol{w}_i\boldsymbol{x})^{2k} + \|\boldsymbol{w}_i\|^{2k}\,\|\boldsymbol{x}\|^{2k}}{m_k}\Big]. \tag{58}$$

Proof. Fix $i \in [\![1, d]\!]$ and an input $\boldsymbol{x} \in \mathcal{X}$. For a fixed index k,

$$\operatorname{Var}(\widehat{f}_i(\boldsymbol{x}) \mid K = k) = \frac{a_k^2}{p_k^2} \operatorname{Var}\left(\phi_k(\boldsymbol{w}_i)^\top \phi_k(\boldsymbol{x})\right). \tag{59}$$

Taking the expectation over K gives

$$\mathbb{E}\big[\operatorname{Var}(\widehat{f}_i(\boldsymbol{x}) \mid K)\big] \;=\; \sum_{k=0}^{\infty} p_k \frac{a_k^2}{p_k^2} \operatorname{Var}\left(\phi_k(\boldsymbol{w}_i)^\top \phi_k(\boldsymbol{x})\right) \;=\; \sum_{k=1}^{\infty} \frac{a_k^2}{p_k} \operatorname{Var}\left(\phi_k(\boldsymbol{w}_i)^\top \phi_k(\boldsymbol{x})\right). \tag{60}$$

We also have $\mathbb{E}[\widehat{f}_i(\boldsymbol{x}) \mid K = k] = \dfrac{a_k}{p_k} \mathbb{E}[\phi_k(\boldsymbol{w}_i)^\top \phi_k(\boldsymbol{x})] = \dfrac{a_k}{p_k} (\boldsymbol{w}_i \boldsymbol{x})^k$. Hence,

$$\operatorname{Var}\big(\mathbb{E}[\widehat{f}_i(\boldsymbol{x}) \mid K]\big) \;=\; \sum_{k=0}^{\infty} p_k \Big(\frac{a_k}{p_k} (\boldsymbol{w}_i \boldsymbol{x})^k\Big)^2 \;-\; \Big(\sum_{k=0}^{\infty} a_k (\boldsymbol{w}_i \boldsymbol{x})^k\Big)^2. \tag{61}$$

The second (negative) term is just $f(\boldsymbol{w}_i \boldsymbol{x})^2$; using the law of total variance, we obtain:

$$\operatorname{Var}(\widehat{f}_i(\boldsymbol{x})) \;=\; \sum_{k=1}^{\infty} \frac{a_k^2}{p_k} \operatorname{Var}\left(\phi_k(\boldsymbol{w}_i)^\top \phi_k(\boldsymbol{x})\right) \;+\; \sum_{k=0}^{\infty} \frac{a_k^2}{p_k} (\boldsymbol{w}_i \boldsymbol{x})^{2k} \;-\; f(\boldsymbol{w}_i \boldsymbol{x})^2. \tag{62}$$

If the $(\phi_k)_k$ are generated using the tensor feature maps from Problem 2, we can use the previous upper bound 21 to derive:

$$\operatorname{Var}(\widehat{f}_i(\boldsymbol{x})) \;\le\; \frac{a_0^2}{p_0} + \sum_{k=1}^{\infty} \frac{a_k^2}{p_k} \Big[(\boldsymbol{w}_i \boldsymbol{x})^{2k} + \frac{(\boldsymbol{w}_i \boldsymbol{x})^{2k} + \|\boldsymbol{w}_i\|^{2k} \, \|\boldsymbol{x}\|^{2k}}{m_k}\Big]. \tag{63}$$

□

3 Problem 4: Smoothed version of the local attention matrix

In this problem we no longer distinguish queries and keys, i.e. we set $q_i = k_i \in \mathbb{R}^{d_{QK}}$ for every position $i = 1, \dots, L$. The ground-truth (masked) attention weights are

$$\boldsymbol{A}_{i,j} = \begin{cases} \exp\left(\frac{\boldsymbol{q}_i^\top \boldsymbol{q}_j}{\sqrt{d_{QK}}}\right) & \text{if } \|\boldsymbol{q}_i - \boldsymbol{q}_j\|_1 \le \delta, \\ 0 & \text{otherwise,} \end{cases} \qquad i, j \in \{1, \dots, L\}, \tag{4}$$

where $\delta > 0$ is a locality hyper-parameter.

3.1 Hadamard factorization of the mask

First, note that

$$\forall (i,j) \in [\![1, L]\!]^2, \quad A_{i,j} = \exp\left(\frac{\boldsymbol{q}_i^\top \boldsymbol{q}_j}{\sqrt{d_{QK}}}\right) \mathbb{1}(\|\boldsymbol{q}_i - \boldsymbol{q}_j\|_1 \le \delta). \tag{64}$$

$$\boldsymbol{M}_1 = \left(\exp\left(\frac{\boldsymbol{q}_i^\top \boldsymbol{q}_j}{\sqrt{d_{QK}}}\right)\right)_{1 \le i,j \le L}, \quad \boldsymbol{M}_2 = \left(\mathbb{1}(\|\boldsymbol{q}_i - \boldsymbol{q}_j\|_1 \le \delta)\right)_{1 \le i,j \le L}. \tag{65}$$

We now have $\boldsymbol{A} = \boldsymbol{M}_1 \odot \boldsymbol{M}_2$, where $\odot$ denotes the Hadamard product. We already know how to approximate $\boldsymbol{M}_1$, which corresponds to the softmax kernel. Specifically, let m be a positive integer. Following [4], we define

$$\phi_m^+ : \mathbb{R}^L \to \mathbb{R}^m, \quad \boldsymbol{x} \mapsto \exp\left(-\frac{\|\boldsymbol{x}\|^2}{2\, d_{QK}^{1/2}}\right) \exp\left(\frac{w^\top \boldsymbol{x}}{d_{QK}^{1/4}}\right), \tag{66}$$

where $w \sim \mathcal{N}(0, \boldsymbol{I}_d)$. Then

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^L, \quad \mathbb{E}\left[\phi_m^+(\boldsymbol{x})^\top \phi_m^+(\boldsymbol{y})\right] = \exp\left(\frac{\boldsymbol{x}^\top \boldsymbol{y}}{\sqrt{d_{QK}}}\right). \tag{67}$$

Hence

$$\boldsymbol{M}_1 = \mathbb{E}\left[\phi_m^+(\boldsymbol{Q})\, \phi_m^+(\boldsymbol{Q})^\top\right]. \tag{68}$$

Now, we want to approximate $\boldsymbol{M}_2$ in the same way. We use an approach similar to the first theorem in [5], employing the Fourier transform for a shift-invariant kernel. First, let $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{y}$, and define

$$k : \boldsymbol{r} \mapsto \mathbb{1}_{\{\|\boldsymbol{r}\|_1 \le \delta\}} = \int_{\mathbb{R}^d} \hat{k}(\boldsymbol{\omega})\, e^{i\, \boldsymbol{\omega}^\top \boldsymbol{r}}\, d\boldsymbol{\omega} \quad \text{a.e.,} \tag{69}$$

where $\hat{k}$ is its inverse Fourier transform defined by

$$\hat{k} : \boldsymbol{\omega} \mapsto \frac{1}{(2\pi)^d} \int_{\|\boldsymbol{r}\|_1 \le \delta} e^{-\, i\, \boldsymbol{\omega}^\top \boldsymbol{r}}\, d\boldsymbol{r}. \tag{70}$$

We remark that this expression for k holds everywhere except at discontinuities, i.e. where $\|\boldsymbol{r}\|_1 = \delta$. The Fourier transform exists because this indicator function is in L^1. Since $\hat{k}$ need not be positive, we separate it into a sign function and an absolute value:

$$\forall \boldsymbol{\omega} \in \mathbb{R}^d, \quad \hat{k}(\boldsymbol{\omega}) = \operatorname{sign}\left(\hat{k}(\boldsymbol{\omega})\right) \left|\hat{k}(\boldsymbol{\omega})\right|. \tag{71}$$

By sampling $\omega_1, \dots, \omega_m$ from the density proportional to $|\hat{k}(\omega)|$ on $\mathbb{R}^d$, we define

$$\phi_m^1(\boldsymbol{q}) = \left(\tfrac{1}{\sqrt{m}} \sqrt{|\hat{k}(\omega_\ell)|}\, e^{i\, \omega_\ell^\top \boldsymbol{q}}\right)_{1 \le \ell \le m}, \tag{72}$$

$$\phi_m^2(\boldsymbol{q}) = \left(\tfrac{1}{\sqrt{m}} \operatorname{sign}(\hat{k}(\omega_\ell)) \sqrt{|\hat{k}(\omega_\ell)|}\, e^{-\, i\, \omega_\ell^\top \boldsymbol{q}}\right)_{1 \le \ell \le m}. \tag{73}$$

By the Monte Carlo method,

$$\boldsymbol{M}_2 = \mathbb{E}\left[\phi_m^1(\boldsymbol{Q})\, \phi_m^2(\boldsymbol{Q})^\top\right]. \tag{74}$$

3.2 Transformation of the Hadamard product into a standard product

Now, we define

$$U = V = \phi_m^+(\boldsymbol{Q}) \in \mathbb{R}^{L\times m}, \quad X = \phi_m^1(\boldsymbol{Q}) \in \mathbb{R}^{L\times m}, \quad Y = \phi_m^2(\boldsymbol{Q}) \in \mathbb{R}^{L\times m}. \tag{75}$$

Let $i, j \in \{1, \dots, L\}$. Then

$$\begin{aligned} A_{i,j} &= \mathbb{E}\Big[\big(\phi_m^+(\boldsymbol{q}_i)\,\phi_m^+(\boldsymbol{q}_j)^\top\big) \odot \big(\phi_m^1(\boldsymbol{q}_i)\,\phi_m^2(\boldsymbol{q}_j)^\top\big)\Big]_{i,j} & (76)\\ &= \mathbb{E}\big[(UV^\top)_{i,j}\,(XY^\top)_{i,j}\big] & (77)\\ &= \mathbb{E}\Big[\sum_{k=1}^{m} U_{ik}V_{jk} \sum_{\ell=0}^{m} X_{i\ell}Y_{j\ell}\Big] & (78)\\ &= \mathbb{E}\Big[\sum_{1\le k,\ell\le m} U_{ik}V_{jk}X_{i\ell}Y_{j\ell}\Big]. & (79)\end{aligned}$$

Now, suppose $(w_1, \dots, w_m) \overset{\text{iid}}{\sim} \mathcal{N}(0, \boldsymbol{I}_d)$ and $(\omega_1, \dots, \omega_m) \overset{\text{iid}}{\sim}$ density $\propto |\hat{k}(\omega)|$ on $\mathbb{R}^d$. A Monte Carlo estimator for (79) is obtained by selecting m pairs (k, ℓ) uniformly from $\{1, \dots, m\}^2$ and averaging. Since the (w_k) and (ω_ℓ) are independent, we may generate m samples of each and sum them. Thus,

$$A_{ij} = \mathbb{E}\Big[\sum_{1\le k\le m} U_{ik}V_{jk}X_{ik}Y_{jk}\Big] = \mathbb{E}\Big[\big((U\odot X)(V\odot Y)^\top\big)_{i,j}\Big]. \tag{80}$$

Therefore, we define $\boldsymbol{Q'}$ and $\boldsymbol{K'}$ as follows:

$$Q' = \big[Q'_{i\ell}\big]_{1\le i\le L,\,1\le \ell\le m}, \quad K' = \big[K'_{j\ell}\big]_{1\le j\le L,\,1\le \ell\le m},$$

where for each $\ell = 1, \dots, m$ we draw

$$w_\ell \sim \mathcal{N}(0, I_{d_{QK}}), \quad \omega_\ell \sim \text{density } \propto |\hat{k}(\omega)|,$$

and set

$$Q'_{i\ell} = \frac{1}{\sqrt{m}} \exp\Big(-\frac{\|\boldsymbol{q}_i\|^2}{2\,d_{QK}^{1/2}}\Big) \exp\Big(\frac{w_\ell^\top \boldsymbol{q}_i}{d_{QK}^{1/4}}\Big) \sqrt{|\hat{k}(\omega_\ell)|}\; e^{i\,\omega_\ell^\top \boldsymbol{q}_i}, \tag{81}$$

$$K'_{j\ell} = \frac{1}{\sqrt{m}} \exp\Big(-\frac{\|\boldsymbol{q}_j\|^2}{2\,d_{QK}^{1/2}}\Big) \exp\Big(\frac{w_\ell^\top \boldsymbol{q}_j}{d_{QK}^{1/4}}\Big) \operatorname{sign}\big(\hat{k}(\omega_\ell)\big) \sqrt{|\hat{k}(\omega_\ell)|}\; e^{-i\,\omega_\ell^\top \boldsymbol{q}_j}. \tag{82}$$

Result 13. *The matrices $\boldsymbol{Q'}$ and $\boldsymbol{K'}$ are defiend such that their product gives an unbiased approximation of $\boldsymbol{A}$, i.e.*

$$\boldsymbol{A} = \mathbb{E}[\boldsymbol{Q'}(\boldsymbol{K'})^\top]. \tag{83}$$

Proof. See above. □

References

[1] N. Pham and R. Pagh, "Fast and scalable polynomial kernels via explicit feature maps," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2013, pp. 239–247. [Online]. Available: `https://chbrown.github.io/kdd-2013-usb/kdd/p239.pdf`

[2] H. Gzyl and J. L. Palacios, "The Weierstrass approximation theorem and large deviations," *Amer. Math. Monthly*, vol. 104, no. 7, pp. 650–653, 1997. [Online]. Available: `https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e3442f80e8bd89722a5f3326fd53d89321bd3f14`

[3] Z. Guan, "Iterated Bernstein polynomial approximations," arXiv:0909.0684 [math.CA], 2009. [Online]. Available: `https://arxiv.org/abs/0909.0684`

[4] K. Choromanski *et al.*, "Rethinking attention with performers," *arXiv preprint arXiv:2009.14794*, 2020. [Online]. Available: `https://arxiv.org/abs/2009.14794`

[5] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Adv. Neural Inf. Process. Syst.*, vol. 20, 2007, pp. 1177–1184. [Online]. Available: `https://papers.nips.cc/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf`