Leverage Score Sampling for Tensor Product Matrices in Input Sparsity Time

David Woodruff
CMU

Amir Zandieh
MPII

Kernel Methods

- Widely used in machine learning, statistics, and control
- Real-world applications:
  - **Hyperparameter Tuning of Deep Neural Networks:** e.g. Google Vizier
  - **Reinforcement Learning: Bayesian Optimization** [Srinivas et. al'09]
  - **Neural Tangent Kernel:** evolution of neural nets during training can be described by kernel methods [Jacot et. al'18]

Kernel Methods

- Learn a nonlinear function $f : \mathbb{R}^d \to \mathbb{R}$ from noisy samples

$$\gamma_i = f(x_i) + \varepsilon_i \text{ for } i = 1,2,\ldots n$$

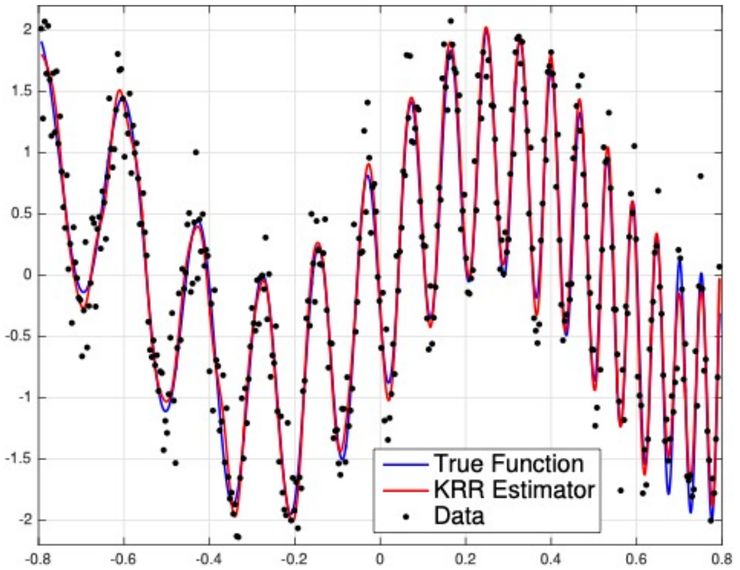
  - ε_i are i.i.d. Gaussian noise

- **Kernel Ridge Regression** is a simple and yet powerful solution

  - Given a kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$

$$\tilde{f}(x) = \sum_{i=1}^{n} \alpha_i \; k(x, x_i)$$

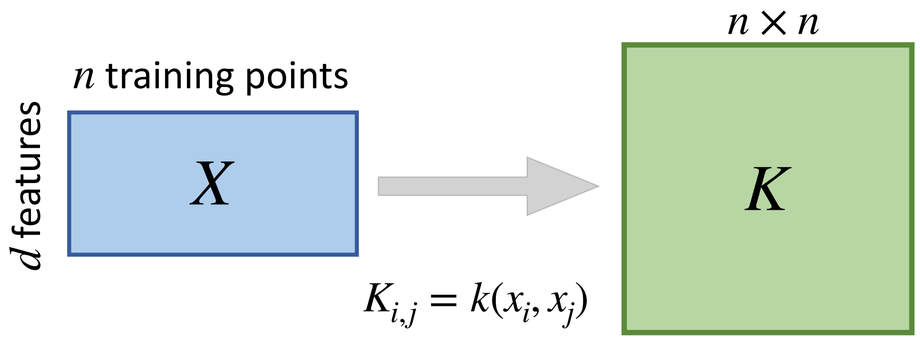
$$\alpha = \arg\min_{\beta \in \mathbb{R}^n} \left\| K\,\beta - \gamma \right\|_2^2 + \lambda\, \beta^\top K\, \beta$$



6

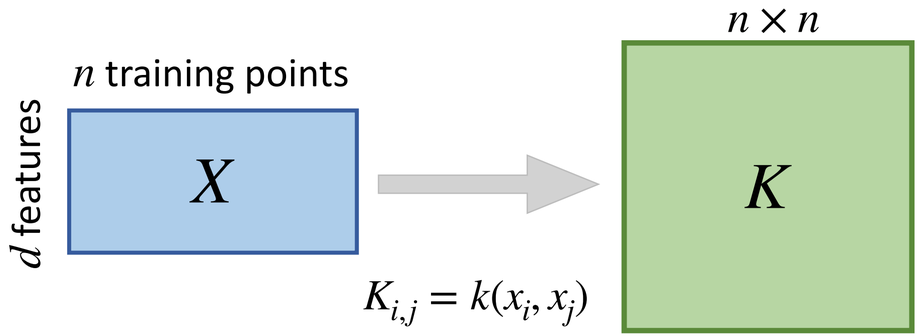
Scalability of Kernel Methods

- Kernel methods are expensive



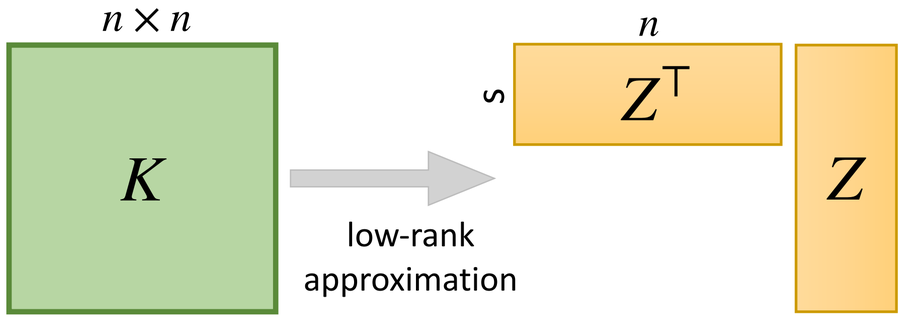
Scalability of Kernel Methods

- Kernel methods are expensive

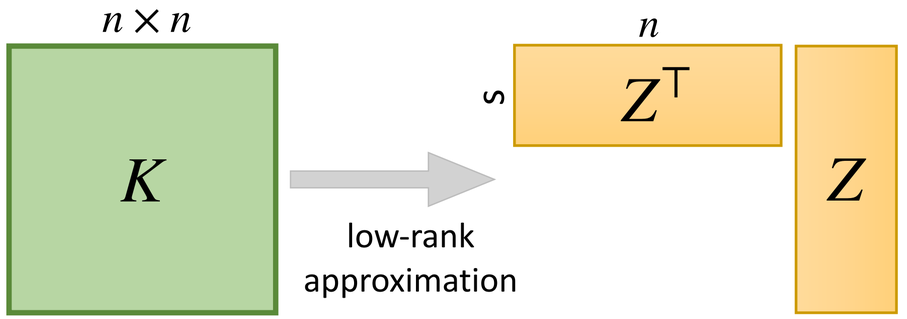


  - Computing all kernel entries takes $\Omega\left(n\ \mathtt{nnz}(X) + n^2\right)$ time
  - Even writing it down takes $\Omega\left(n^2\right)$ time and memory
  - A single iteration of a linear system solver takes $\Omega\left(n^2\right)$ time
  - For $n = 100{,}000$ storing K requires $80GB$!

Classic Solution: Dimensionality Reduction



Classic Solution: Dimensionality Reduction



- Storing Z uses $O(ns)$ space and computing $Z^\top Z$ α takes $O(ns)$ time
- Orthogonalisation, eigen-decomposition, and pseudo-inversion of $Z^\top Z$ all take just $O\left(ns^2\right)$ time

Efficient Low-Rank Approximation?

- Direct eigen-decomposition, or even approximation via Krylov subspace methods are out of question since they at least require fully forming K !

Efficient Low-Rank Approximation?

- Direct eigen-decomposition, or even approximation via Krylov subspace methods are out of question since they at least require fully forming K !
- Many faster methods proposed: Nyström Method [WS'01, MM'17], Random Features [RR'08, AKMMVZ'17, ZHASKS'21], Oblivious Sketching [ANW'14, AKKPVWZ'20, SWYZ'21], Leverage Score Sampling [WZ'20]

Efficient Low-Rank Approximation?

- Direct eigen-decomposition, or even approximation via Krylov subspace methods are out of question since they at least require fully forming K !
- Many faster methods proposed: Nyström Method [WS'01, MM'17], Random Features [RR'08, AKMMVZ'17, ZHASKS'21], Oblivious Sketching [ANW'14, AKKPVWZ'20, SWYZ'21], Leverage Score Sampling [WZ'20]
- Our approach: kernel low-rank approximation based on leverage score sampling

Kernel Feature Map

- Any kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ defines a lifting $\phi : \mathbb{R}^d \to \mathbb{R}^D$ such that

$$k(x, y) = \phi(x)^\top \phi(y)$$

- The kernel computes the inner product between the lifted data points

Kernel Feature Map

- Any kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ defines a lifting $\phi : \mathbb{R}^d \to \mathbb{R}^D$ such that

$$k(x, y) = \phi(x)^\top \phi(y)$$

- The kernel computes the inner product between the lifted data points
- $$K = \Phi^\top \Phi$$
- Φ is a feature matrix with n columns whose i^{th} column is $\phi(x_i)$

Polynomial Kernel

- One of our main focuses is the **Polynomial Kernel**

$$k(x, y) = \left(x^\top y\right)^q$$

- Lifting function: $\phi(x) = x^{\otimes q} \in \mathbb{R}^{d^q}$

Polynomial Kernel

- One of our main focuses is the **Polynomial Kernel**

$$k(x, y) = \left(x^\top y\right)^q$$

- Lifting function: $\phi(x) = x^{\otimes q} \in \mathbb{R}^{d^q}$

$$\Phi = X^{\otimes q} \in \mathbb{R}^{d^q \times n}$$

- Goal: design a sampler $\Pi \in \mathbb{R}^{s \times d^q}$ such that $\Pi\ X^{\otimes q}$ is efficiently computable without needing to form $X^{\otimes q}$ explicitly

Approximation Guarantee

- **Subspace Embedding:** for every $\varepsilon, \lambda > 0$ and any dataset matrix $X \in \mathbb{R}^{d \times n}$, if $\Phi = X^{\otimes q}$ is the the degree-q Polynomial feature matrix, w.h.p.

$$\frac{\Phi^\top \Phi + \lambda I}{1+\varepsilon} \preceq \Phi^\top \Pi^\top \Pi \, \Phi + \lambda I \preceq \frac{\Phi^\top \Phi + \lambda I}{1-\varepsilon}$$

- Want:
  - Number of samples at most statistical dimension $s_\lambda = \mathtt{tr}\left(K(K+\lambda I)^{-1}\right)$
  - Total time to find Π and compute $\Pi \, \Phi$, at most $\mathcal{O}(\mathtt{nnz}(X))$

Prior Work

- Ahle, Kapralov, Knudsen, Pagh, Velingker, Woodruff, **Z**'20: Oblivious Subspace Embedding (OSE) of the Polynomial kernel
  - Target dimension $s \approx \varepsilon^{-2} q^4 s_\lambda$
  - Time to apply the sketch $\Pi\ X^{\otimes q}$ is $\varepsilon^{-2} q^5 \cdot \mathtt{nnz}(X)$

Prior Work

- Ahle, Kapralov, Knudsen, Pagh, Velingker, Woodruff, **Z**'20: Oblivious Subspace Embedding (OSE) of the Polynomial kernel
  - Target dimension $s \approx \varepsilon^{-2} q^4 s_\lambda$
  - Time to apply the sketch $\Pi\ X^{\otimes q}$ is $\varepsilon^{-2} q^5 \cdot \mathtt{nnz}(X)$

- **suboptimal target dimension**
- **multiplicative q^5/ε^2 factor in runtime**

Prior Work

- Ahle, Kapralov, Knudsen, Pagh, Velingker, Woodruff, **Z**'20: Oblivious Subspace Embedding (OSE) of the Polynomial kernel
  - Target dimension $s \approx \varepsilon^{-2} q^4 s_\lambda$
  - Time to apply the sketch $\Pi\ X^{\otimes q}$ is $\varepsilon^{-2} q^5 \cdot \mathtt{nnz}(X)$
- Song, Woodruff, Yu, Zhang'21: OSE for the degree-q Polynomial kernel
  - Target dimension $s \approx n/\varepsilon^2$
  - Time to apply the sketch $\Pi\ X^{\otimes q}$ is $\varepsilon^{-2} q^2 n^2 + nd$

Prior Work

- Ahle, Kapralov, Knudsen, Pagh, Velingker, Woodruff, **Z**'20: Oblivious Subspace Embedding (OSE) of the Polynomial kernel
  - Target dimension $s \approx \varepsilon^{-2} q^4 s_\lambda$
  - Time to apply the sketch $\Pi\ X^{\otimes q}$ is $\varepsilon^{-2} q^5 \cdot \mathrm{nnz}(X)$

- **suboptimal target dimension**
- **quadratic dependence on n in runtime**

- Song, Woodruff, Yu, Zhang'21: OSE for the degree-q Polynomial kernel
  - Target dimension $s \approx n/\varepsilon^2$
  - Time to apply the sketch $\Pi\ X^{\otimes q}$ is $\varepsilon^{-2} q^2 n^2 + nd$

Prior Work

- Woodruff, **Z**'20: Leverage Score Sampling with OSE guarantee for the degree-q Polynomial kernel
  - Target dimension $s \approx s_\lambda/\varepsilon^2$
  - Sampling time $\Pi\ X^{\otimes q}$ is $q^{2.5} \cdot \texttt{nnz}(X)$
- This method showed that $\texttt{poly}(1/\varepsilon)$ factors can be decoupled from the leading term of the runtime

Prior Work

- Woodruff, **Z**'20: Leverage Score Sampling with OSE guarantee for the degree-q Polynomial kernel
  - Target dimension $s \approx s_\lambda/\varepsilon^2$
  - Sampling time $\Pi\ X^{\otimes q}$ is $q^{2.5} \cdot \mathtt{nnz}(X)$
- This method showed that $\mathtt{poly}(1/\varepsilon)$ factors can be decoupled from the leading term of the runtime

- **multiplicative $q^{2.5}$ factor in runtime**

Main Result

- **Theorem 1.** For every $X \in \mathbb{R}^{d \times n}$, if s_λ is the statistical dimension of the degree-q polynomial kernel on this dataset, then there exists an algorithm that outputs a sampling matrix $\Pi \in \mathbb{R}^{s \times d^q}$ with $s \approx s_\lambda / \varepsilon^2$ samples using $\mathcal{O}\left(\min\{q \cdot \mathtt{nnz}(X), nd\}\right)$ runtime such that w.h.p.

$$\frac{X^{\otimes q\top} X^{\otimes q} + \lambda I}{1+\varepsilon} \preceq X^{\otimes q\top} \Pi^\top \Pi \; X^{\otimes q} + \lambda I \preceq \frac{X^{\otimes q\top} X^{\otimes q} + \lambda I}{1-\varepsilon}$$

Implications for Other Kernels

- For datasets with ℓ_2 radius $r > 0$:
- OSE for the Gaussian kernel as well as any dot-product kernels with rapidly convergent Taylor expansion in time $\mathcal{O}\left(\min\left\{r^2 \cdot \mathrm{nnz}(X), nd\right\}\right)$