Randomized SVD, CUR Decomposition, and SPSD Matrix Approximation

Shusen Wang

Outline

- CX Decomposition & Approximate SVD
- CUR Decomposition
- SPSD Matrix Approximation

CX Decomposition

- Given any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$
- The CX decomposition of $\mathbf{A}$
  1. Sketching: $\mathbf{C} = \mathbf{AP} \in \mathbb{R}^{m \times c}$
  2. Find $\mathbf{X}$ such that $\mathbf{A} \approx \mathbf{CX}$
     - E.g. $\mathbf{X}^\star = \mathrm{argmin}_{\mathbf{X}} \|\mathbf{A} - \mathbf{CX}\|_{\mathrm{F}}^2 = \mathbf{C}^\dagger \mathbf{A}$
     - It costs $O(mnc)$

CX Decomposition

- Let the sketching matrix $\mathbf{P} \in \mathbb{R}^{n \times c}$ be defined in the table.
- $\min_{\text{rank}(\mathbf{X}) \leq k} ||\mathbf{A} - \mathbf{C}\mathbf{X}||_F^2 \leq (1 + \epsilon) \; ||\mathbf{A} - \mathbf{A}_k||_F^2$

	Uniform sampling	Leverage score sampling	Gaussian projection	SRHT	Count sketch
c $\geq$	$O\left(\nu k \left(\log k + \frac{1}{\epsilon}\right)\right)$	$O\left(k \left(\log k + \frac{1}{\epsilon}\right)\right)$	$O\left(\frac{k}{\epsilon}\right)$	$O\left((k + \log n)\left(\log k + \frac{1}{\epsilon}\right)\right)$	$O\left(k^2 + \frac{k}{\epsilon}\right)$

ν is the column coherence of $\mathbf{A}_\mathrm{k}$

CX Decomposition ⇔ Approximate SVD

- CX decomposition ⇔ approximate SVD

$$\mathbf{A} \approx \mathbf{C}\mathbf{X}$$

CX Decomposition ⇔ Approximate SVD

- CX decomposition ⇔ approximate SVD

$$\mathbf{A} \approx \mathbf{C}\mathbf{X} = \mathbf{U}_{\mathrm{C}}\boldsymbol{\Sigma}_{\mathrm{C}}\mathbf{V}_{\mathrm{C}}^{\mathrm{T}}\mathbf{X}$$

SVD: $\mathbf{C} = \mathbf{U}_{\mathrm{C}}\, \boldsymbol{\Sigma}_{\mathrm{C}}\mathbf{V}_{\mathrm{C}}^{\mathrm{T}} \in \mathbb{R}^{m \times c}$

Time cost: $O(mc^2)$

CX Decomposition ⇔ Approximate SVD

- CX decomposition ⇔ approximate SVD

$$\mathbf{A} \approx \mathbf{C}\mathbf{X} = \mathbf{U}_\mathrm{C}\mathbf{\Sigma}_\mathrm{C}\mathbf{V}_\mathrm{C}^\mathrm{T}\mathbf{X} = \mathbf{U}_\mathrm{C}\mathbf{Z}$$

Let $\mathbf{\Sigma}_\mathrm{C}\mathbf{V}_\mathrm{C}^\mathrm{T}\mathbf{X} = \mathbf{Z} \in \mathbb{R}^{c \times n}$

SVD: $\mathbf{C} = \mathbf{U}_\mathrm{C}\,\mathbf{\Sigma}_\mathrm{C}\mathbf{V}_\mathrm{C}^\mathrm{T} \in \mathbb{R}^{m \times c}$

Time cost: $O(mc^2 + nc^2)$

CX Decomposition ⇔ Approximate SVD

- CX decomposition ⇔ approximate SVD

$$\mathbf{A} \approx \mathbf{C}\mathbf{X} = \mathbf{U}_\mathrm{C}\mathbf{\Sigma}_\mathrm{C}\mathbf{V}_\mathrm{C}^\mathrm{T}\mathbf{X} = \mathbf{U}_\mathrm{C}\mathbf{Z} = \mathbf{U}_\mathrm{C}\mathbf{U}_\mathrm{Z}\mathbf{\Sigma}_\mathrm{Z}\mathbf{V}_\mathrm{Z}^\mathrm{T}$$

Let $\mathbf{\Sigma}_\mathrm{C}\mathbf{V}_\mathrm{C}^\mathrm{T}\mathbf{X} = \mathbf{Z} \in \mathbb{R}^{c\times n}$

SVD: $\mathbf{C} = \mathbf{U}_\mathrm{C}\,\mathbf{\Sigma}_\mathrm{C}\mathbf{V}_\mathrm{C}^\mathrm{T} \in \mathbb{R}^{m\times c}$

SVD: $\mathbf{Z} = \mathbf{U}_\mathrm{Z}\mathbf{\Sigma}_\mathrm{Z}\mathbf{V}_\mathrm{Z}^\mathrm{T} \in \mathbb{R}^{c\times n}$

Time cost: $O(mc^2 + nc^2 + nc^2)$

CX Decomposition ⇔ Approximate SVD

- CX decomposition ⇔ approximate SVD

$m \times s$ matrix with orthonormal columns

$$\mathbf{A} \approx \mathbf{C}\mathbf{X} = \mathbf{U}_{\mathrm{C}} \boldsymbol{\Sigma}_{\mathrm{C}} \mathbf{V}_{\mathrm{C}}^{\mathrm{T}} \mathbf{X} = \mathbf{U}_{\mathrm{C}} \mathbf{Z} = \mathbf{U}_{\mathrm{C}} \mathbf{U}_{\mathrm{Z}} \boldsymbol{\Sigma}_{\mathrm{Z}} \mathbf{V}_{\mathrm{Z}}^{\mathrm{T}}$$

$s \times n$ matrix with orthonormal rows

Let $\boldsymbol{\Sigma}_{\mathrm{C}} \mathbf{V}_{\mathrm{C}}^{\mathrm{T}} \mathbf{X} = \mathbf{Z} \in \mathbb{R}^{c \times n}$

diagonal matrix

SVD: $\mathbf{C} = \mathbf{U}_{\mathrm{C}} \boldsymbol{\Sigma}_{\mathrm{C}} \mathbf{V}_{\mathrm{C}}^{\mathrm{T}} \in \mathbb{R}^{m \times c}$

SVD: $\mathbf{Z} = \mathbf{U}_{\mathrm{Z}} \boldsymbol{\Sigma}_{\mathrm{Z}} \mathbf{V}_{\mathrm{Z}}^{\mathrm{T}} \in \mathbb{R}^{c \times n}$

Time cost: $O(mc^2 + nc^2 + nc^2 + mc^2)$

CX Decomposition ⇔ Approximate SVD

- CX decomposition ⇔ approximate SVD
- Done! Approximate rank c SVD: $\mathbf{A} \approx (\mathbf{U}_\mathrm{C}\mathbf{U}_\mathrm{Z})\mathbf{\Sigma}_\mathrm{Z}\mathbf{V}_\mathrm{Z}^\mathrm{T}$

$m \times s$ matrix with orthonormal columns

$$\mathbf{A} \;\approx\; \mathbf{CX} \;=\; \mathbf{U}_\mathrm{C}\mathbf{\Sigma}_\mathrm{C}\mathbf{V}_\mathrm{C}^\mathrm{T}\mathbf{X} \;=\; \mathbf{U}_\mathrm{C}\mathbf{Z} \;=\; \mathbf{U}_\mathrm{C}\mathbf{U}_\mathrm{Z}\mathbf{\Sigma}_\mathrm{Z}\mathbf{V}_\mathrm{Z}^\mathrm{T}$$

$s \times n$ matrix with orthonormal rows

diagonal matrix

Time cost: $O(mc^2 + nc^2 + nc^2 + mc^2) = O(mc^2 + nc^2)$

CX Decomposition $\Leftrightarrow$ Approximate SVD

- CX decomposition $\Leftrightarrow$ approximate SVD

- Given $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{C} \in \mathbb{R}^{m \times c}$, the approximate SVD costs
  - $O(mnc)$ time
  - $O(mc + nc)$ memory

CX Decomposition

- The CX decomposition of $\mathbf{A} \in \mathbb{R}^{m \times n}$
  - Optimal solution: $\mathbf{X}^{\star} = \mathrm{argmin}_{\mathbf{X}} \left|\left|\mathbf{A} - \mathbf{CX}\right|\right|_{\mathbf{F}}^{2} = \mathbf{C}^{\dagger}\mathbf{A}$
  - How to make it more efficient?

CX Decomposition

- The CX decomposition of $\mathbf{A} \in \mathbb{R}^{m \times n}$
  - Optimal solution: $\mathbf{X}^\star = \boxed{\mathrm{argmin}_{\mathbf{X}} \|\mathbf{A} - \mathbf{CX}\|_{\mathrm{F}}^2} = \mathbf{C}^\dagger \mathbf{A}$
  - How to make it more efficient?

A regression problem!

Fast CX Decomposition

- Fast CX [Drineas, Mahoney, Muthukrishnan, 2008][Clarkson & Woodruff, 2013]
  - Draw another sketching matrix $\mathbf{S} \in \mathbb{R}^{m \times s}$
  - Compute $\widetilde{\mathbf{X}} = \mathrm{argmin}_{\mathbf{X}} \left|\left|\mathbf{S}^T(\mathbf{A} - \mathbf{CX})\right|\right|_{\mathrm{F}}^2 = \left(\mathbf{S}^{\mathrm{T}}\mathbf{C}\right)^{\dagger}\left(\mathbf{S}^{\mathrm{T}}\mathbf{A}\right)$
  - Time cost: $O(ncs) + \mathrm{TimeOfSketch}$
  - When $s = \tilde{O}(c/\epsilon)$,

$$\left|\left|\mathbf{A} - \mathbf{C}\widetilde{\mathbf{X}}\right|\right|_{\mathrm{F}}^2 \leq (1+\epsilon) \cdot \min_{\mathbf{X}} \left|\left|\mathbf{A} - \mathbf{CX}\right|\right|_{\mathrm{F}}^2$$

Outline

- CX Decomposition & Approximate SVD
- CUR Decomposition
- SPSD Matrix Approximation

CUR Decomposition

- Sketching
  - $\mathbf{C} = \mathbf{A}\mathbf{P}_{\mathbf{C}} \in \mathbb{R}^{m \times c}$
  - $\mathbf{R} = \mathbf{P}_{\mathbf{R}}^{\mathrm{T}}\mathbf{A} \in \mathbb{R}^{r \times n}$
- Find $\mathbf{U}$ such that $\mathbf{CUR} \approx \mathbf{A}$
- CUR $\Leftrightarrow$ Approximate SVD
  - In the same way as "*CX* $\Leftrightarrow$ *Approximate SVD*"

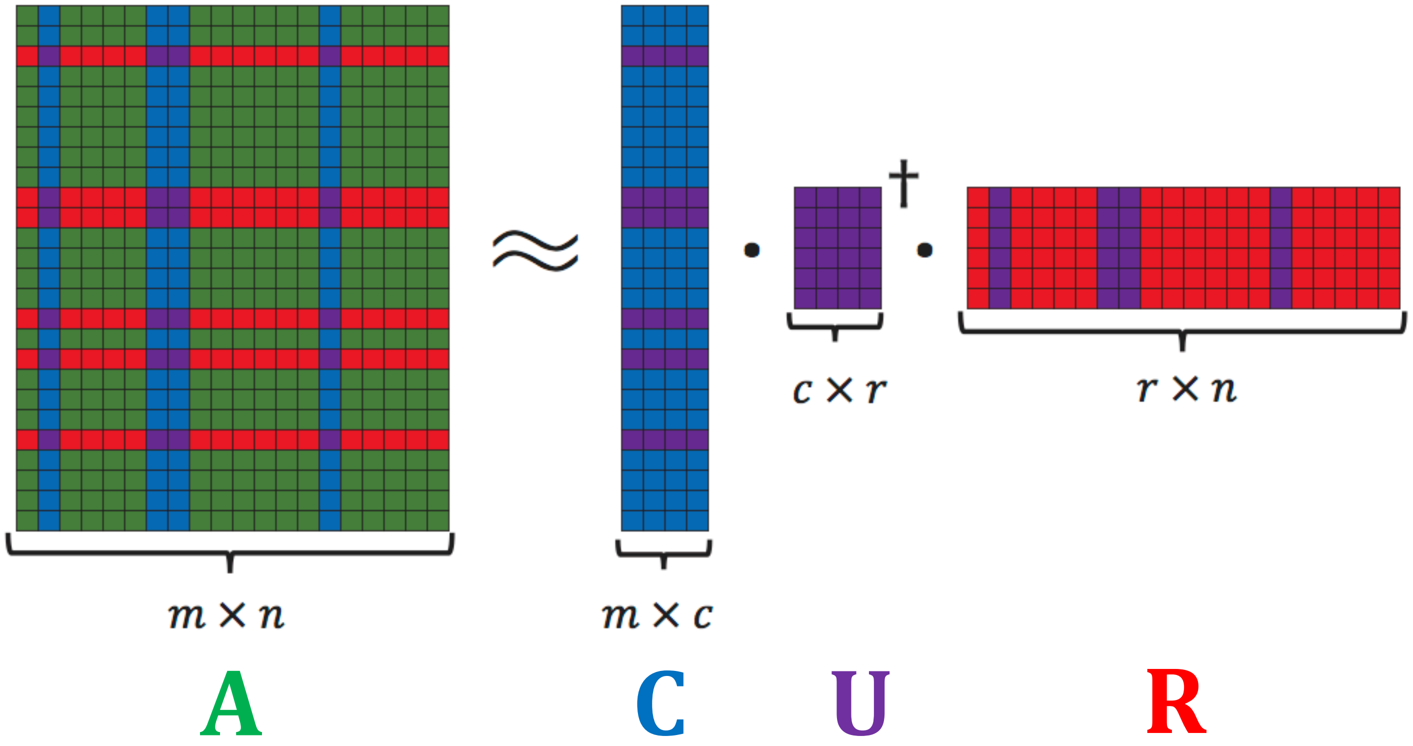
CUR Decomposition

- Sketching
  - $\mathbf{C} = \mathbf{A}\mathbf{P}_{\mathbf{C}} \in \mathbb{R}^{m \times c}$
  - $\mathbf{R} = \mathbf{P}_{\mathbf{R}}^{\mathrm{T}}\mathbf{A} \in \mathbb{R}^{r \times n}$
- Find $\mathbf{U}$ such that $\mathbf{CUR} \approx \mathbf{A}$
- CUR $\Leftrightarrow$ Approximate SVD
  - In the same way as "*CX*$\Leftrightarrow$ *Approximate SVD*"
- 3 types of $\mathbf{U}$

CUR Decomposition

- Type 1 [Drineas, Mahoney, Muthukrishnan, 2008]:

$$\mathbf{U} = \left(\mathbf{P}_{\mathbf{R}}^{T}\mathbf{A}\mathbf{P}_{\mathbf{C}}\right)^{\dagger}$$



CUR Decomposition

- Type 1 [Drineas, Mahoney, Muthukrishnan, 2008]:

$$\mathbf{U} = \left(\mathbf{P}_{\mathbf{R}}^{T} \mathbf{A} \mathbf{P}_{\mathbf{C}}\right)^{\dagger}$$

- Recall the fast CX decomposition

$$\mathbf{A} \approx \mathbf{C}\tilde{\mathbf{X}} = \mathbf{C}\left(\mathbf{P}_{\mathbf{R}}^{T}\mathbf{C}\right)^{\dagger}\left(\mathbf{P}_{\mathbf{R}}^{T}\mathbf{A}\right)$$

CUR Decomposition

- Type 1 [Drineas, Mahoney, Muthukrishnan, 2008]:

$$\mathbf{U} = \left(\mathbf{P}_{\mathbf{R}}^{T}\mathbf{A}\mathbf{P}_{\mathbf{C}}\right)^{\dagger}$$

- Recall the fast CX decomposition

$$\mathbf{A} \approx \mathbf{C}\widetilde{\mathbf{X}} = \mathbf{C}\left(\mathbf{P}_{\mathbf{R}}^{T}\mathbf{C}\right)^{\dagger}\left(\mathbf{P}_{\mathbf{R}}^{T}\mathbf{A}\right) = \mathbf{C}\mathbf{U}\mathbf{R}$$

CUR Decomposition

- Type 1 [Drineas, Mahoney, Muthukrishnan, 2008]:

$$\mathbf{U} = \left(\mathbf{P}_{\mathbf{R}}^{T}\mathbf{A}\mathbf{P}_{\mathbf{C}}\right)^{\dagger}$$

- Recall the fast CX decomposition

$$\mathbf{A} \approx \mathbf{C}\widetilde{\mathbf{X}} = \mathbf{C}\left(\mathbf{P}_{\mathbf{R}}^{T}\mathbf{C}\right)^{\dagger}\left(\mathbf{P}_{\mathbf{R}}^{T}\mathbf{A}\right) = \mathbf{CUR}$$

- They're equivalent: $\mathbf{C}\,\widetilde{\mathbf{X}} = \mathbf{C}\,\mathbf{U}\,\mathbf{R}$

CUR Decomposition

- Type 1 [Drineas, Mahoney, Muthukrishnan, 2008]:

$$\mathbf{U} = \left(\mathbf{P}_{\mathbf{R}}^{T}\mathbf{A}\mathbf{P}_{\mathbf{C}}\right)^{\dagger}$$

- Recall the fast CX decomposition

$$\mathbf{A} \approx \mathbf{C}\widetilde{\mathbf{X}} = \mathbf{C}\left(\mathbf{P}_{\mathbf{R}}^{T}\mathbf{C}\right)^{\dagger}\left(\mathbf{P}_{\mathbf{R}}^{T}\mathbf{A}\right) = \mathbf{CUR}$$

- They're equivalent: $\mathbf{C}\,\widetilde{\mathbf{X}} = \mathbf{C}\,\mathbf{U}\,\mathbf{R}$
- Require $c = \tilde{O}\left(\frac{k}{\epsilon}\right)$ and $r = \tilde{O}\left(\frac{c}{\epsilon}\right)$ such that

$$\left|\left|\mathbf{A} - \mathbf{CUR}\right|\right|_{\mathrm{F}}^{2} \leq (1+\epsilon)\left|\left|\mathbf{A} - \mathbf{A}_{k}\right|\right|_{\mathrm{F}}^{2}$$

CUR Decomposition

- Type 1 [Drineas, Mahoney, Muthukrishnan, 2008]:

$$\mathbf{U} = \left(\mathbf{P}_{\mathbf{R}}^{\mathbf{T}} \mathbf{A} \mathbf{P}_{\mathbf{C}}\right)^{\dagger}$$

- Efficient
  - $\mathrm{O}(rc^2) + \mathrm{TimeOfSketch}$
- Loose bound
  - Sketch size $\propto \epsilon^{-2}$
- Bad empirical performance

CUR Decomposition

- Type 2: Optimal CUR

$$\mathbf{U}^\star = \min_{\mathbf{U}} \|\mathbf{A} - \mathbf{CUR}\|_F^2 = \mathbf{C}^\dagger \mathbf{A} \mathbf{R}^\dagger$$

CUR Decomposition

- Type 2: Optimal CUR

$$\mathbf{U}^{\star} = \min_{\mathbf{U}} \left|\left|\mathbf{A} - \mathbf{CUR}\right|\right|_F^2 = \mathbf{C}^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}$$

- Theory [W & Zhang, 2013], [Boutsidis & Woodruff, 2014]:
  - $\mathbf{C}$ and $\mathbf{R}$ are selected by the adaptive sampling algorithm
  - $c = O\left(\frac{k}{\epsilon}\right)$ and $r = O\left(\frac{k}{\epsilon}\right)$
  - $\left|\left|\mathbf{A} - \mathbf{CUR}\right|\right|_F^2 \leq (1 + \epsilon) \left|\left|\mathbf{A} - \mathbf{A}_k\right|\right|_{\mathrm{F}}^2$

CUR Decomposition

- Type 2: Optimal CUR

$$\mathbf{U}^{\star} = \min_{\mathbf{U}} \left|\left|\mathbf{A} - \mathbf{CUR}\right|\right|_F^2 = \mathbf{C}^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}$$

- Inefficient
  - $\mathrm{O}(mnc) + \mathrm{TimeOfSketch}$

CUR Decomposition

- Type 3: Fast CUR [W, Zhang, Zhang, 2015]
  - Draw 2 sketching matrices $\mathbf{S_C}$ and $\mathbf{S_R}$
  - Solve the problem

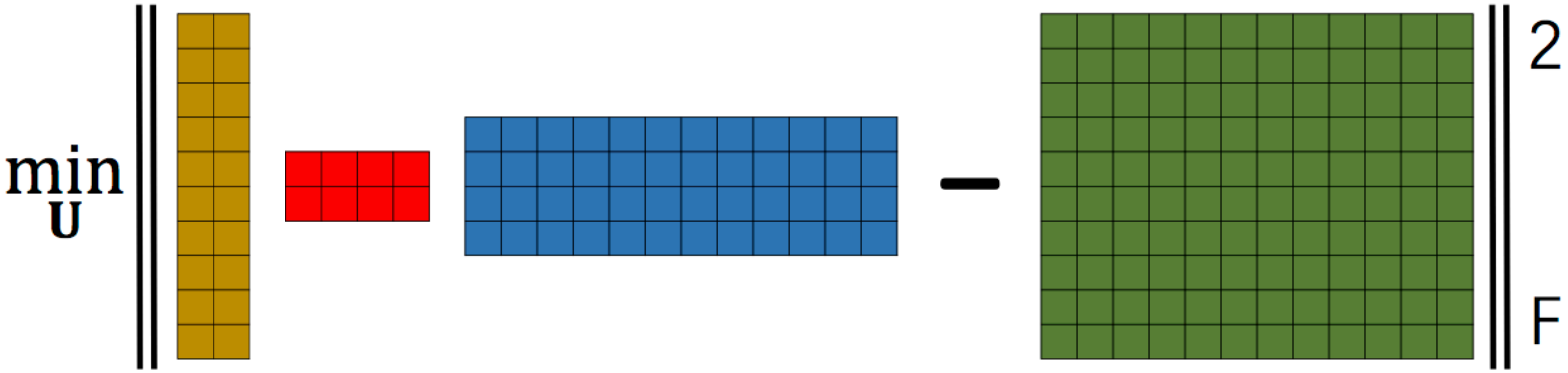
$$\widetilde{\mathbf{U}} = \min_{\mathbf{U}} \left|\left| \boldsymbol{S}_{\boldsymbol{C}}^{T}(\mathbf{A} - \mathbf{CUR})\mathbf{S_R} \right|\right|_{F}^{2} = \left(\mathbf{S}_{\mathbf{C}}^{\mathrm{T}}\mathbf{C}\right)^{\dagger}\left(\mathbf{S}_{\mathbf{C}}^{T}\mathbf{A}\boldsymbol{S}_{\boldsymbol{R}}\right)(\mathbf{R}\mathbf{S_R})^{\dagger}$$

- Intuition?

CUR Decomposition

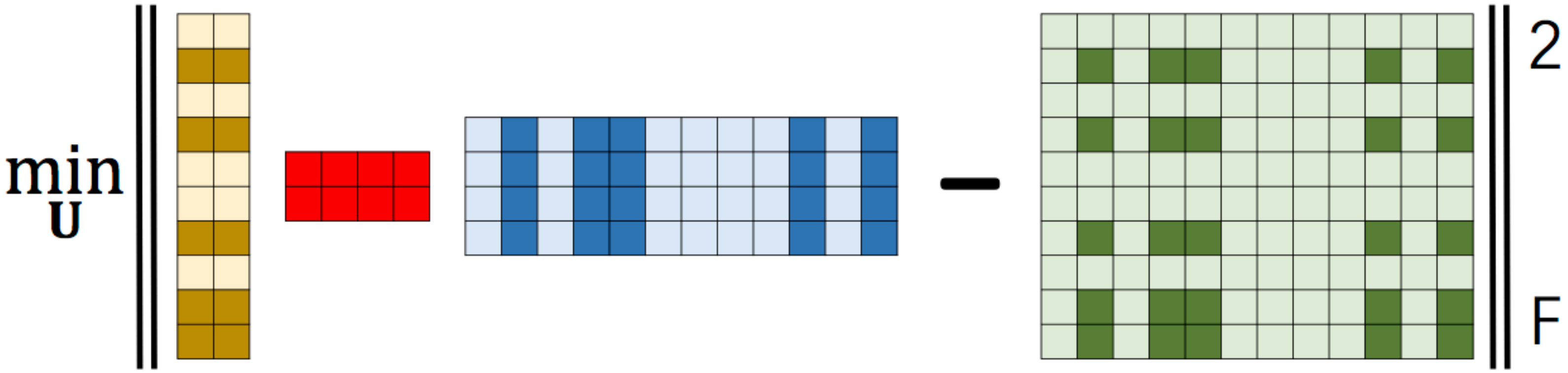
- The optimal $\mathbf{U}$ matrix is obtained by the optimization problem

$$\mathbf{U}^{\star} = \min_{\mathbf{U}} \left|\left|\mathbf{CUR} - \mathbf{A}\right|\right|_F^2$$



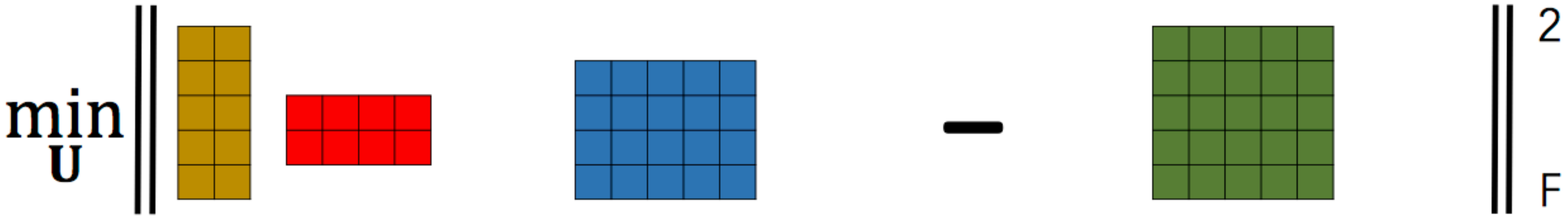
CUR Decomposition

- Approximately solve the optimization problem, e.g. by column selection



CUR Decomposition

- Solve the small scale problem



CUR Decomposition

- Type 3: Fast CUR [W, Zhang, Zhang, 2015]
  - Draw 2 sketching matrices $\mathbf{S_C} \in \mathbb{R}^{m \times s_c}$ and $\mathbf{S_R} \in \mathbb{R}^{n \times s_r}$
  - Solve the problem

$$\tilde{\mathbf{U}} = \min_{\mathbf{U}} \left|\left|\mathbf{S}_C^T(\mathbf{A} - \mathbf{CUR})\mathbf{S_R}\right|\right|_F^2 = \left(\mathbf{S}_\mathbf{C}^\mathrm{T}\mathbf{C}\right)^\dagger\left(\mathbf{S}_\mathbf{C}^T\mathbf{A}\mathbf{S}_R\right)(\mathbf{R}\mathbf{S_R})^\dagger$$

- Theory
  - $s_c = O\left(\frac{c}{\epsilon}\right)$ and $s_r = O\left(\frac{r}{\epsilon}\right)$
  - $\left|\left|\mathbf{A} - \mathbf{C}\tilde{\mathbf{U}}\mathbf{R}\right|\right|_F^2 \leq (1 + \epsilon) \cdot \min_{\mathbf{U}} \left|\left|\mathbf{A} - \mathbf{CUR}\right|\right|_F^2$

CUR Decomposition

- Type 3: Fast CUR [W, Zhang, Zhang, 2015]
  - Draw 2 sketching matrices $\mathbf{S_C} \in \mathbb{R}^{m \times s_c}$ and $\mathbf{S_R} \in \mathbb{R}^{n \times s_r}$
  - Solve the problem

$$\tilde{\mathbf{U}} = \min_{\mathbf{U}} \left\| \mathbf{S}_{\boldsymbol{C}}^{T} (\mathbf{A} - \mathbf{CUR}) \mathbf{S_R} \right\|_F^2 = \left(\mathbf{S}_{\mathbf{C}}^{\mathrm{T}} \mathbf{C} \right)^{\dagger} \left(\mathbf{S}_{\mathbf{C}}^{T} \mathbf{A} \mathbf{S}_{\boldsymbol{R}} \right) (\mathbf{R} \mathbf{S_R})^{\dagger}$$

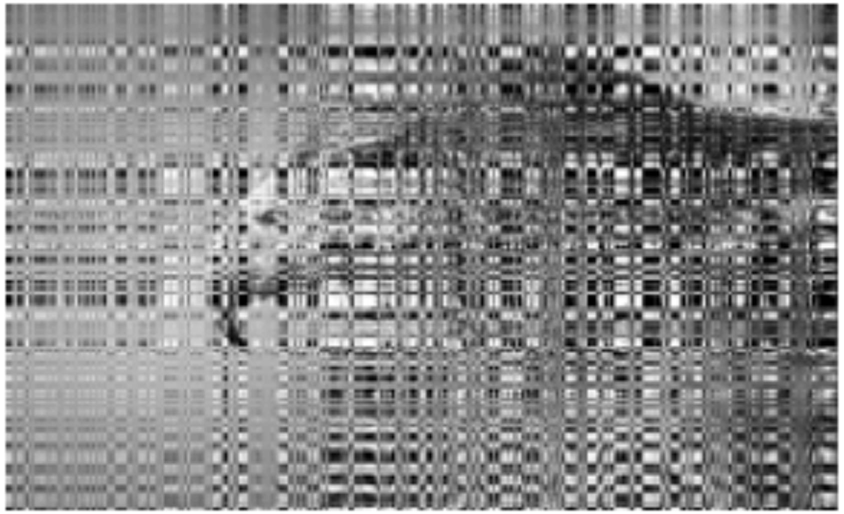
- Efficient
  - $O\big(s_c s_r (c + r)\big) + \mathrm{TimeOfSketch}$
- Good empirical performance

A:
$m = 1920$
$n = 1168$

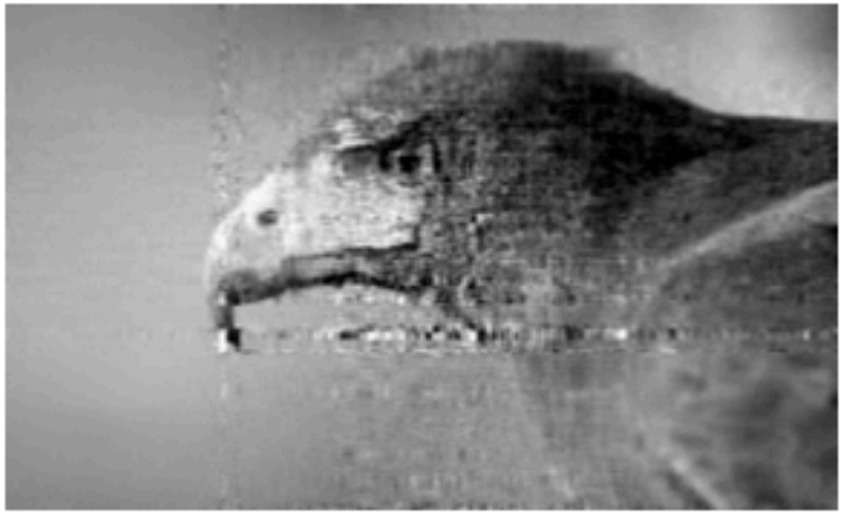
C and **R:**
- $c = r = 100$
- uniform sampling

Original

Type 2: Optimal CUR



Type 1: Fast CX



Type 3: Fast CUR
$s_c = 2c, \qquad s_r = 2r$

Type 3: Fast CUR
$s_c = 4c, \qquad s_r = 4r$

Conclusions

- Approximate truncated SVD
  - CX decomposition
  - CUR decomposition (3 types)
- Fast CUR is the best

Outline

- CX Decomposition & Approximate SVD
- CUR Decomposition
- SPSD Matrix Approximation

Motivation 1: Kernel Matrix

- Given n samples $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^d$ and kernel function $\kappa(\cdot,\cdot)$.
- E.g. Gaussian RBF kernel

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2}{\sigma^2}\right).$$

Motivation 1: Kernel Matrix

- Given n samples $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^d$ and kernel function $\kappa(\cdot,\cdot)$.
- E.g. Gaussian RBF kernel

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left|\left|\mathbf{x}_i - \mathbf{x}_j\right|\right|_2^2}{\sigma^2}\right).$$

- Computing the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$
  - where $k_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$
  - costs $O(n^2 d)$ time

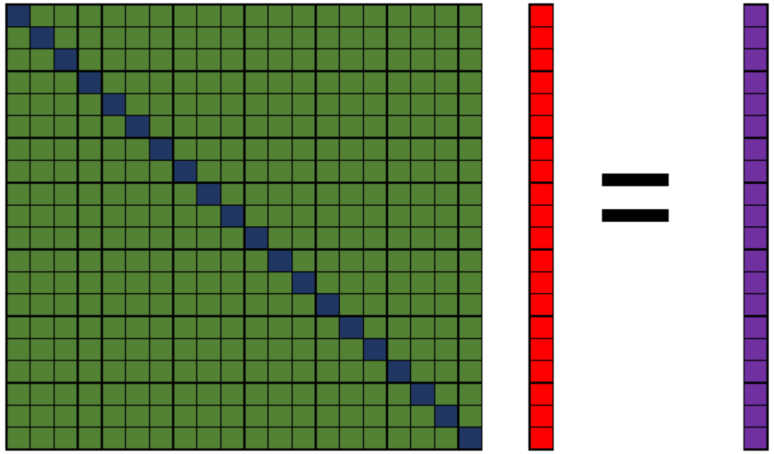
Motivation 2: Matrix Inversion

- Solve the linear system

$$(\mathbf{K} + \alpha \mathbf{I}_n)\mathbf{w} = \mathbf{y}$$

  to find $\mathbf{w} \in \mathbb{R}^n$.

- $\mathbf{K} \in \mathbb{R}^{n \times n}$ is the kernel matrix
- $\mathbf{y} = [y_1, \cdots, y_n] \in \mathbb{R}^n$ contains the labels



Motivation 2: Matrix Inversion

- Solve the linear system
$$(\mathbf{K} + \alpha \mathbf{I}_n)\mathbf{w} = \mathbf{y}$$
  to find $\mathbf{w} \in \mathbb{R}^n$.
- Solution: $\mathbf{w}^\star = (\mathbf{K} + \alpha \mathbf{I}_n)^{-1}\mathbf{y}$

Motivation 2: Matrix Inversion

- Solve the linear system

$$(\mathbf{K} + \alpha \mathbf{I}_n)\mathbf{w} = \mathbf{y}$$

  to find $\mathbf{w} \in \mathbb{R}^n$.
- Solution: $\mathbf{w}^\star = (\mathbf{K} + \alpha \mathbf{I}_n)^{-1}\mathbf{y}$
- It costs
  - $O(n^3)$ time
  - $O(n^2)$ memory.

Motivation 2: Matrix Inversion

- Solve the linear system
$$(\mathbf{K} + \alpha \mathbf{I}_n)\mathbf{w} = \mathbf{y}$$
  to find $\mathbf{w} \in \mathbb{R}^n$.
- Solution: $\mathbf{w}^\star = (\mathbf{K} + \alpha \mathbf{I}_n)^{-1}\mathbf{y}$
- It costs
  - $O(n^3)$ time
  - $O(n^2)$ memory.
- Performed by
  - Kernel ridge regression
  - Least squares kernel SVM

Motivation 3: Eigenvalue Decomposition

- Find the top k ($\ll n$) eigenvectors of $\mathbf{K}$.
- It costs
  - $\tilde{O}(n^2 k)$ time
  - $O(n^2)$ memory.

Motivation 3: Eigenvalue Decomposition

- Find the top k ($\ll n$) eigenvectors of $\mathbf{K}$.
- It costs
  - $\tilde{O}(n^2 k)$ time
  - $O(n^2)$ memory.
- Performed by
  - Kernel PCA (k is the target rank)
  - Manifold learning (k is the target rank)

Computational Challenges

- Time costs
  - Computing kernel matrix: $O(n^2 d)$
  - Matrix inversion: $O(n^3)$
  - Rank k eigenvalue decomposition: $O(n^2 k)$

Computational Challenges

- Time costs
  - Computing kernel matrix: $O(n^2 d)$
  - Matrix inversion: $O(n^3)$
  - Rank k eigenvalue decomposition: $O(n^2 k)$

At least quadratic time!

Computational Challenges

- Time costs
  - Computing kernel matrix: $O(n^2 d)$
  - Matrix inversion: $O(n^3)$
  - Rank k eigenvalue decomposition: $O(n^2 k)$
- Memory costs
  - Inversion and eigenvalue decomposition: $O(n^2)$

Computational Challenges

- Time costs
  - Computing kernel matrix: $O(n^2 d)$
  - Matrix inversion: $O(n^3)$
  - Rank k eigenvalue decomposition: $O(n^2 k)$
- Memory costs
  - Inversion and eigenvalue decomposition: $O(n^2)$
  - Because
    - the numerical algorithms are pass-inefficient
    - ➔ form $\mathbf{K}$ and keep it in memory

Computational Challenges

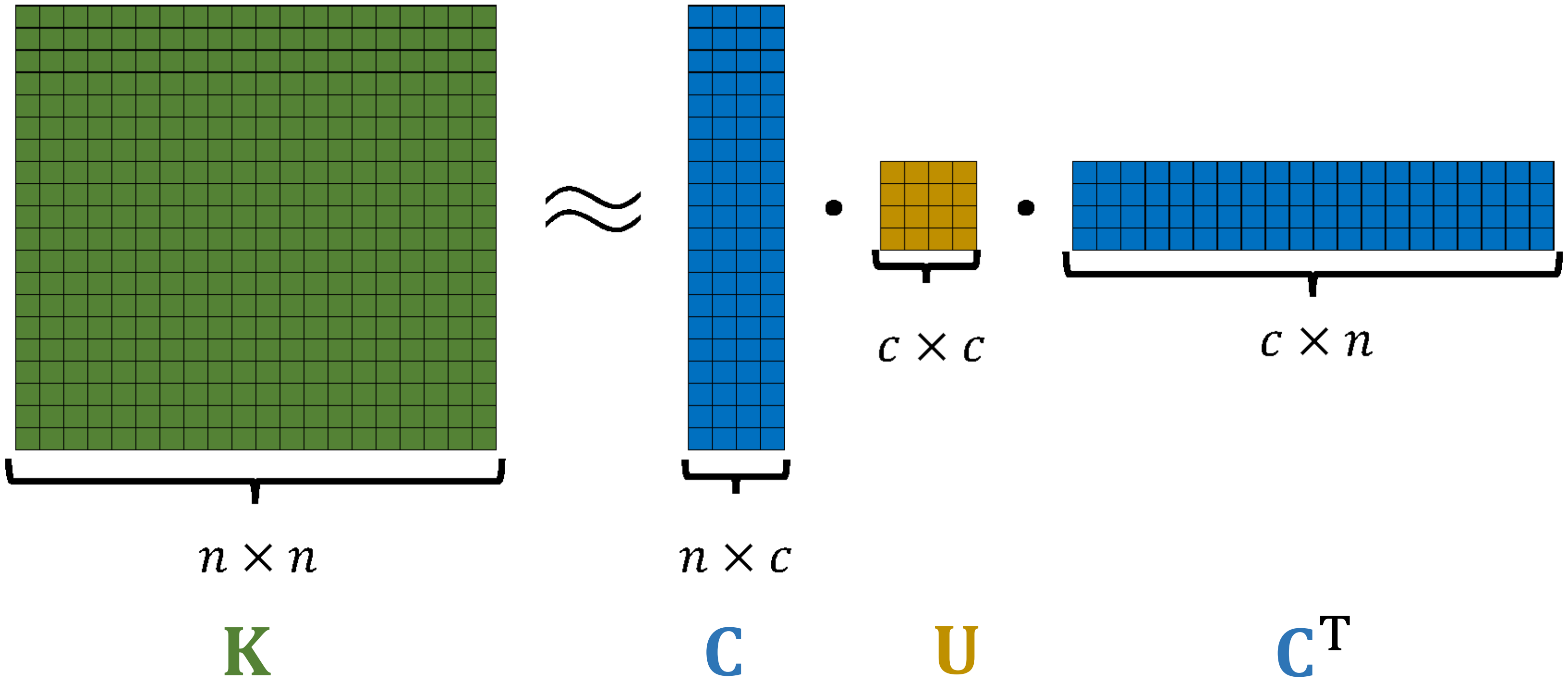
- Time costs
  - Computing kernel matrix: $O(n^2 d)$
  - Matrix inversion: $O(n^3)$
  - Rank k eigenvalue decomposition: $O(n^2 k)$
- Memory costs
  - Inversion and eigenvalue decomposition: $O(n^2)$
  - Because
    - the numerical algorithms are pass-inefficient
    - ➔ form $\mathbf{K}$ and keep it in memory

When $n = 10^5$, the $n \times n$ matrix costs 80GB memory!

How to Speedup?

- Efficiently form the low-rank approximation

$$\mathbf{K} \approx \mathbf{C}\,\mathbf{U}\,\mathbf{C}^{\mathrm{T}}$$



How to Speedup?

- Efficiently form the low-rank approximation

$$\mathbf{K} \approx \mathbf{C}\,\mathbf{U}\,\mathbf{C}^{\mathrm{T}}$$

- Equivalent $\mathbf{K} \approx \mathbf{L}\,\mathbf{L}^{\mathrm{T}}$

Efficient Matrix Inversion

- Solve the linear system $(\mathbf{K} + \alpha \mathbf{I}_n)\mathbf{w} = \mathbf{y}$:

$$\mathbf{w}^\star = (\mathbf{K} + \alpha \mathbf{I}_n)^{-1}\mathbf{y}$$

Efficient Matrix Inversion

- Approximately solve the linear system $(\mathbf{K} + \alpha\mathbf{I}_n)\mathbf{w} = \mathbf{y}$:
  - Replace $\mathbf{K}$ by $\mathbf{L}\mathbf{L}^{\mathrm{T}}$: $\mathbf{w}^\star = (\mathbf{K} + \alpha\mathbf{I}_n)^{-1}\mathbf{y} \approx \left(\mathbf{L}\mathbf{L}^{\mathrm{T}} + \alpha\mathbf{I}_n\right)^{-1}\mathbf{y}$

Efficient Matrix Inversion

- Approximately solve the linear system $(\mathbf{K} + \alpha \mathbf{I}_n)\mathbf{w} = \mathbf{y}$
  - Replace $\mathbf{K}$ by $\mathbf{L}\mathbf{L}^{\mathrm{T}}$: $\quad \mathbf{w}^{\star} = (\mathbf{K} + \alpha \mathbf{I}_n)^{-1}\mathbf{y} \approx \left(\mathbf{L}\mathbf{L}^{\mathrm{T}} + \alpha \mathbf{I}_n\right)^{-1}\mathbf{y}$
  - Expand the inversion by the Woodbury identity

Efficient Matrix Inversion

- Approximately solve the linear system $(\mathbf{K} + \alpha \mathbf{I}_n)\mathbf{w} = \mathbf{y}$
  - Replace $\mathbf{K}$ by $\mathbf{L}\mathbf{L}^{\mathrm{T}}$: $\mathbf{w}^{\star} = (\mathbf{K} + \alpha \mathbf{I}_n)^{-1}\mathbf{y} \approx \left(\mathbf{L}\mathbf{L}^{\mathrm{T}} + \alpha \mathbf{I}_n\right)^{-1}\mathbf{y}$
  - Expand the inversion by the Woodbury identity

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{D}\mathbf{A}^{-1}$$

Efficient Matrix Inversion

- Approximately solve the linear system $(\mathbf{K} + \alpha \mathbf{I}_n)\mathbf{w} = \mathbf{y}$
  - Replace $\mathbf{K}$ by $\mathbf{L}\mathbf{L}^{\mathrm{T}}$: $\quad \mathbf{w}^{\star} = (\mathbf{K} + \alpha \mathbf{I}_n)^{-1}\mathbf{y} \approx \left(\mathbf{L}\mathbf{L}^{\mathrm{T}} + \alpha \mathbf{I}_n\right)^{-1}\mathbf{y}$
  - Expand the inversion by the Woodbury identity

$$\mathbf{w}^{\star} \approx \alpha^{-1}\mathbf{y} + \alpha^{-1}\mathbf{L}(\alpha \mathbf{I} + \mathbf{L}^{T}\mathbf{L})^{-1}\mathbf{L}^{\mathbf{T}}\mathbf{y}$$

Efficient Matrix Inversion

- Approximately solve the linear system $(\mathbf{K} + \alpha \mathbf{I}_n)\mathbf{w} = \mathbf{y}$
  - Replace $\mathbf{K}$ by $\mathbf{L}\mathbf{L}^{\mathrm{T}}$: $\mathbf{w}^{\star} = (\mathbf{K} + \alpha \mathbf{I}_n)^{-1}\mathbf{y} \approx \left(\mathbf{L}\mathbf{L}^{\mathrm{T}} + \alpha \mathbf{I}_n\right)^{-1}\mathbf{y}$
  - Expand the inversion by the Woodbury identity

$$\mathbf{w}^{\star} \approx \alpha^{-1}\mathbf{y} + \alpha^{-1}\mathbf{L}(\alpha\mathbf{I} + \mathbf{L}^{T}\mathbf{L})^{-1}\mathbf{L}^{\mathbf{T}}\mathbf{y}$$

  - Time cost: $O(nc^2)$

Linear in n, much better than $O(n^3)$

Efficient Eigenvalue Decomposition

- Approximately compute the k-eigenvalue decomposition of $\mathbf{K}$
  - SVD: $\mathbf{L} = \mathbf{U_L \Sigma_L V_L}$
  - $\mathbf{K} \approx \mathbf{L L^T} = \mathbf{U_L \Sigma_L^2 U_L^T}$

Efficient Eigenvalue Decomposition

- Approximately compute the k-eigenvalue decomposition of $\mathbf{K}$
  - SVD: $\mathbf{L} = \mathbf{U_L}\mathbf{\Sigma_L}\mathbf{V_L}$
  - $\mathbf{K} \approx \mathbf{L}\mathbf{L}^\mathbf{T} = \mathbf{U_L}\mathbf{\Sigma}_\mathbf{L}^2\mathbf{U}_\mathbf{L}^\mathbf{T}$
  - Approximate k- eigenvalue decomposition of $\mathbf{K}$
    - eigenvectors: the first k vectors in $\mathbf{U_L}$

  - Time cost: $O(nc^2)$

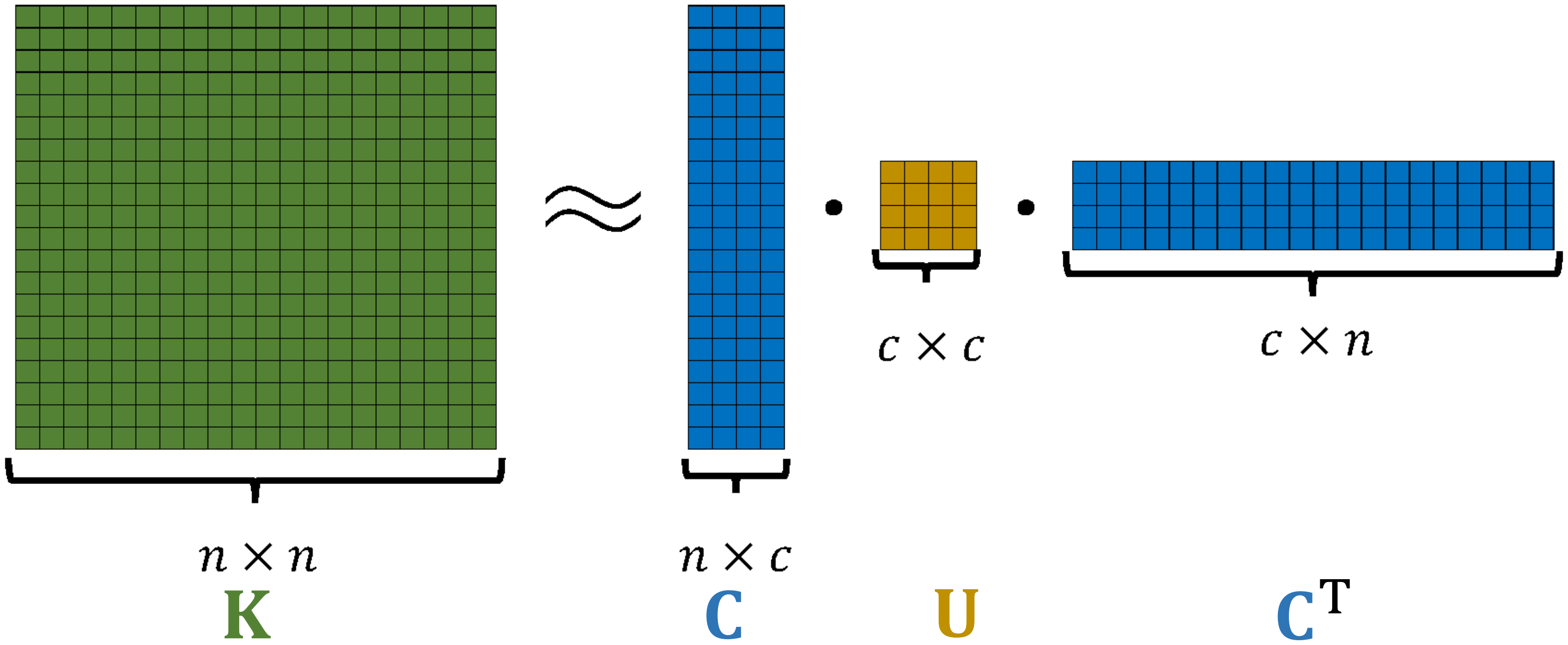
Efficient Eigenvalue Decomposition

- Approximately compute the k-eigenvalue decomposition of $\mathbf{K}$
  - SVD: $\mathbf{L} = \mathbf{U_L \Sigma_L V_L}$
  - $\mathbf{K} \approx \mathbf{L}\mathbf{L}^{\mathbf{T}} = \mathbf{U_L \Sigma_L^2 U_L^T}$
  - Approximate k- eigenvalue decomposition of $\mathbf{K}$
    - eigenvectors: the first k vectors in $\mathbf{U_L}$
  - Time cost: $O(nc^2)$
    - Much lower than $\tilde{O}(n^2 k)$

Sketching Based Models

- How to find such an approximation?

$$\mathbf{K} \approx \mathbf{C}\,\mathbf{U}\,\mathbf{C}^{\mathrm{T}}$$



Sketching Based Models

- How to find such an approximation?

$$\mathbf{K} \approx \mathbf{C}\,\mathbf{U}\,\mathbf{C}^{\mathrm{T}}$$

- Sketching based Methods: $\mathbf{C} = \mathbf{K}\boldsymbol{S} \in \mathbb{R}^{n\times c}$ is a sketch of $\mathbf{K}$.
  - $\boldsymbol{S} \in \mathbb{R}^{n\times c}$ can be column selection or random projection matrix

Sketching Based Models

- How to find such an approximation?

$$\mathbf{K} \approx \mathbf{C}\,\mathbf{U}\,\mathbf{C}^{\mathrm{T}}$$

- Sketching based Methods: $\mathbf{C} = \mathbf{K}\boldsymbol{S} \in \mathbb{R}^{n \times c}$ is a sketch of $\mathbf{K}$.
  - $\boldsymbol{S} \in \mathbb{R}^{n \times c}$ can be column selection or random projection matrix
- Three methods:
  - The prototype model [HMT11, WZ13, WLZ16]
  - The fast model [WZZ15]
  - The Nyström method [WS15, GM13]

The Prototype Model

- Objective: $\mathbf{K} \approx \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}}$
- Minimize the approximation error by

$$\mathbf{U}^{\star} = \underset{\mathbf{U}}{\operatorname{argmin}} \left\| \mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}} \right\|_F^2 = \mathbf{C}^{\dagger}\mathbf{K}\left(\mathbf{C}^{\dagger}\right)^{\mathrm{T}}.$$

The Prototype Model

- Objective: $\mathbf{K} \approx \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}}$
- Minimize the approximation error by

$$\mathbf{U}^{\star} = \operatorname*{argmin}_{\mathbf{U}} \left|\left|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}}\right|\right|_F^2 = \mathbf{C}^{\dagger}\mathbf{K}\left(\mathbf{C}^{\dagger}\right)^{\mathrm{T}}.$$

Extension of the random SVD to SPSD matrix [HMT11]

The Prototype Model

- Objective: $\mathbf{K} \approx \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}}$
- Minimize the approximation error by

$$\mathbf{U}^{\star} = \operatorname*{argmin}_{\mathbf{U}} \left\| \mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}} \right\|_F^2 = \mathbf{C}^{\dagger}\mathbf{K}\left(\mathbf{C}^{\dagger}\right)^{\mathrm{T}}.$$

- Time: $O(n^2 c)$

- The time complexity is nearly the same to the k-eigenvalue decomposition.
- It is much faster than the k- eigenvalue decomposition in practice.

The Prototype Model

- Objective: $\mathbf{K} \approx \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}}$
- Minimize the approximation error by

$$\mathbf{U}^{\star} = \underset{\mathbf{U}}{\mathrm{argmin}} \left\| \mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}} \right\|_F^2 = \mathbf{C}^{\dagger}\mathbf{K}\left(\mathbf{C}^{\dagger}\right)^{\mathrm{T}}.$$

- Time: $O(n^2 c)$
- #Passes: one

The Prototype Model

- Objective: $\mathbf{K} \approx \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}}$
- Minimize the approximation error by

$$\mathbf{U}^{\star} = \underset{\mathbf{U}}{\operatorname{argmin}} \left\| \mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}} \right\|_F^2 = \mathbf{C}^{\dagger}\mathbf{K}\left(\mathbf{C}^{\dagger}\right)^{\mathrm{T}}.$$

- Time: $O(n^2 c)$
- #Passes: one
- Memory: $O(nc)$
  - Put k_{ij} in memory only when it is visited
  - Keep $\mathbf{C}^{\dagger}$ in memory

The Prototype Model

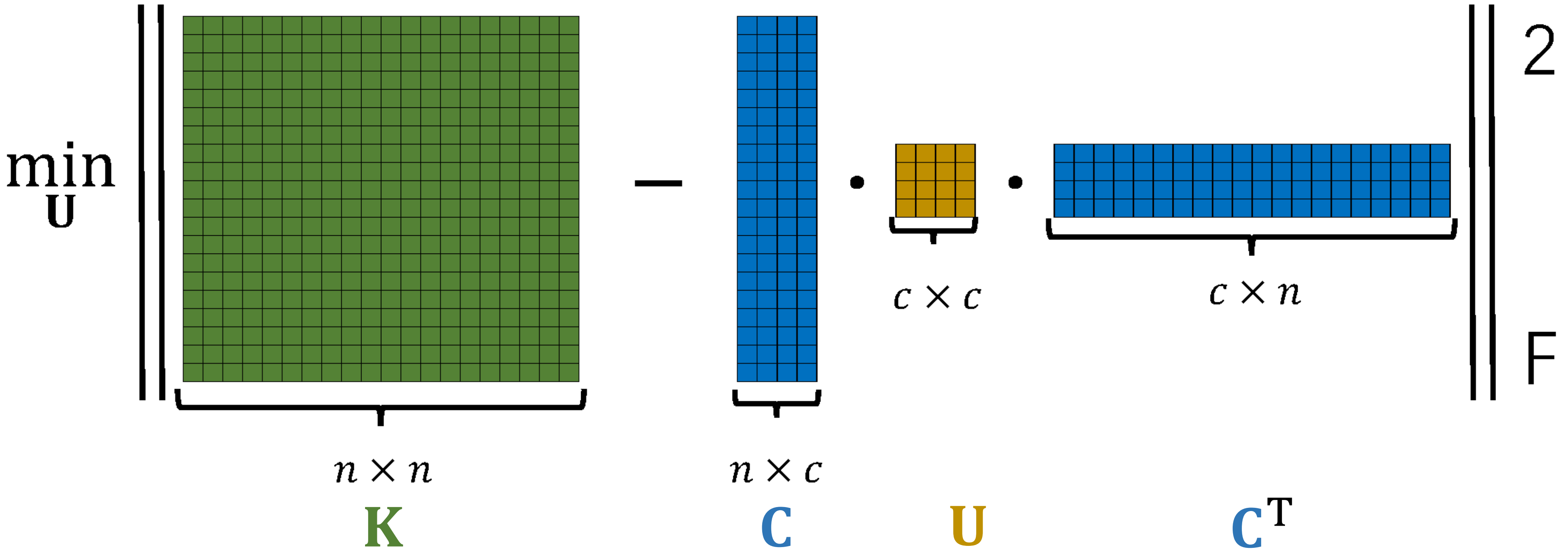
- Error Bound
  - $k \ll n$ is arbitrary integer
  - $\mathbf{P}$ samples $c = O\left(\frac{k}{\epsilon}\right)$ columns by adaptive sampling
  - $\mathbb{E}\left|\left|\mathbf{K} - \mathbf{C}\mathbf{U}^{\star}\mathbf{C}^{\mathrm{T}}\right|\right|_F^2 \leq (1+\epsilon)\left|\left|\mathbf{K} - \mathbf{K}_k\right|\right|_F^2$

The Prototype Model

- Limitations
  - $\mathbf{U}^{\star} = \mathbf{C}^{\dagger}\mathbf{K}\left(\mathbf{C}^{\dagger}\right)^{\mathrm{T}}$
  - Time cost is $O(n^2 c)$
  - Requires observing the whole of $\mathbf{K}$

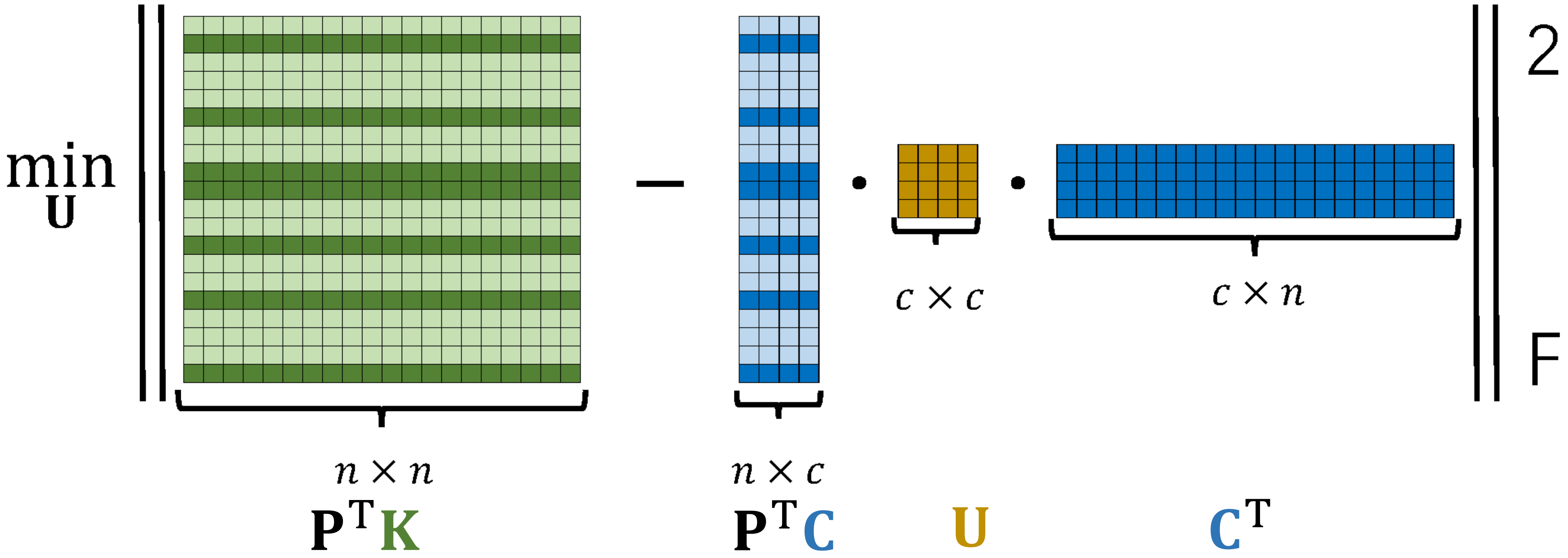
The Prototype Model

- Prototype model: $\mathbf{K} \approx \mathbf{C}\,\mathbf{U}^{\star}\,\mathbf{C}^{\mathrm{T}}$, where
$$\mathbf{U}^{\star} = \underset{\mathbf{U}}{\operatorname{argmin}} \left\|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}}\right\|_{\mathrm{F}}^{2}.$$



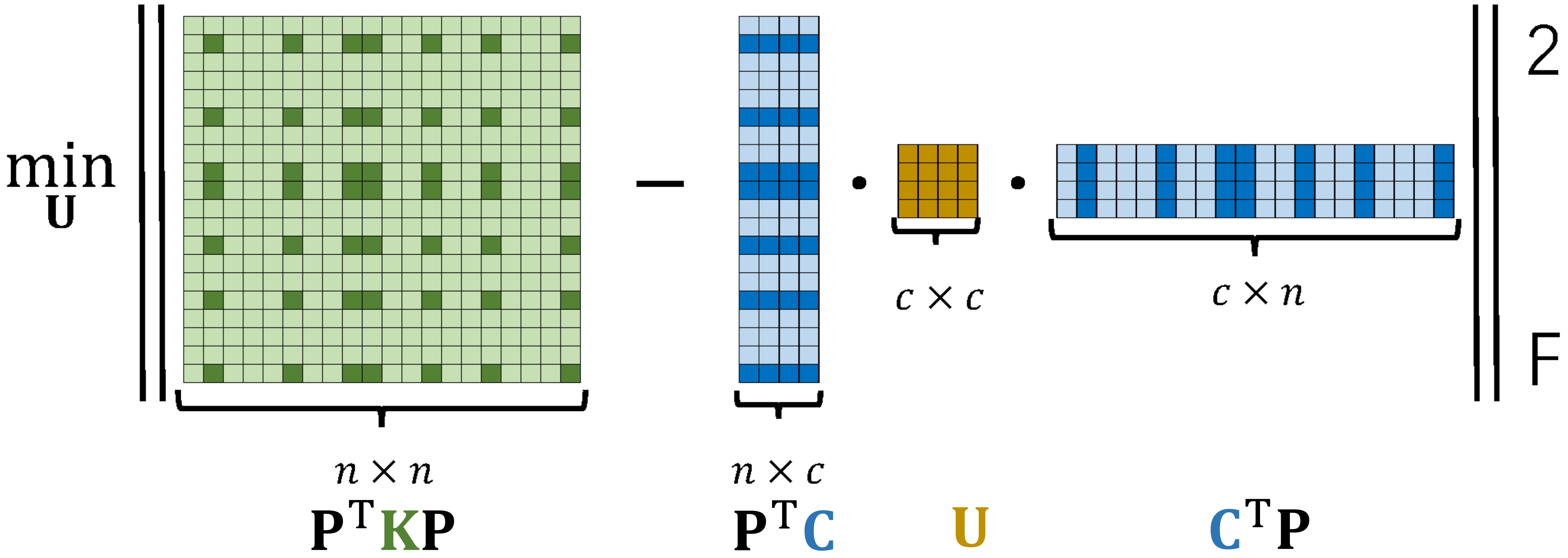
The Fast Model

- Column/row selection
  - Form $\mathbf{P}^T\mathbf{K}\mathbf{P}$ and $\mathbf{P}^T\mathbf{C}$



The Fast Model

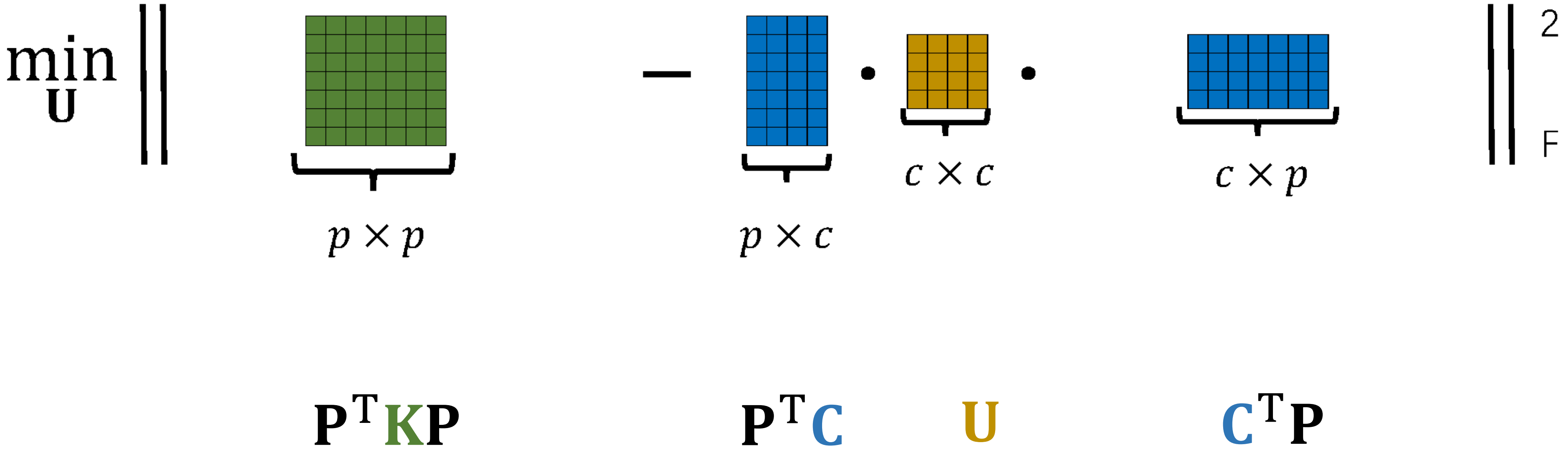
- Column/row selection
  - Form $\mathbf{P}^{\mathrm{T}}\mathbf{K}\mathbf{P}$ and $\mathbf{P}^{\mathrm{T}}\mathbf{C}$



The Fast Model

- $\mathbf{K} \approx \mathbf{C}\,\tilde{\mathbf{U}}\,\mathbf{C}^{\mathrm{T}}$, where

$$\tilde{\mathbf{U}} = \operatorname*{argmin}_{\mathbf{U}} \left\| \mathbf{P}^{\mathrm{T}}\left(\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}}\right)\mathbf{P} \right\|_{F}^{2}.$$



The Fast Model

- Prototype model: $\mathbf{U}^\star = \underset{\mathbf{U}}{\mathrm{argmin}} \left\| \mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}} \right\|_{\mathrm{F}}^{2} = \mathbf{C}^{\dagger}\mathbf{K}\left(\mathbf{C}^{\dagger}\right)^{\mathrm{T}}$
- Fast model: $\tilde{\mathbf{U}} = \underset{\mathbf{U}}{\mathrm{argmin}} \left\| \mathbf{P}^{\mathrm{T}}\left(\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}}\right)\mathbf{P} \right\|_{F}^{\mathbf{2}} = \left(\mathbf{P}^{\mathrm{T}}\mathbf{C}\right)^{\dagger}\left(\mathbf{P}^{\mathrm{T}}\mathbf{K}\mathbf{P}\right)\left(\mathbf{C}^{\mathrm{T}}\mathbf{P}\right)^{\dagger}.$

The Fast Model

- Prototype model: $\mathbf{U}^{\star} = \underset{\mathbf{U}}{\operatorname{argmin}} \left|\left|\mathbf{K} - \mathbf{CUC}^{\mathrm{T}}\right|\right|_{\mathrm{F}}^{2} = \mathbf{C}^{\dagger}\mathbf{K}\left(\mathbf{C}^{\dagger}\right)^{\mathrm{T}}$
- Fast model: $\tilde{\mathbf{U}} = \underset{\mathbf{U}}{\operatorname{argmin}} \left|\left|\mathbf{P}^{\mathrm{T}}\left(\mathbf{K} - \mathbf{CUC}^{\mathrm{T}}\right)\mathbf{P}\right|\right|_{F}^{\mathbf{2}} = \left(\mathbf{P}^{\mathrm{T}}\mathbf{C}\right)^{\dagger}\left(\mathbf{P}^{\mathrm{T}}\mathbf{KP}\right)\left(\mathbf{C}^{\mathrm{T}}\mathbf{P}\right)^{\dagger}.$
- Theory
  - $p = O\left(\sqrt{\frac{nc}{\epsilon}}\right)$
  - $\mathbf{P}$ is column selection matrix (according to the row leverage scores of $\mathbf{C}$)
  - Then $\left|\left|\mathbf{K} - \mathbf{C}\tilde{\mathbf{U}}\mathbf{C}^{\mathrm{T}}\right|\right|_{F}^{2} \leq (1+\epsilon)\left|\left|\mathbf{K} - \mathbf{CU}^{\star}\mathbf{C}^{\mathrm{T}}\right|\right|_{F}^{2}$

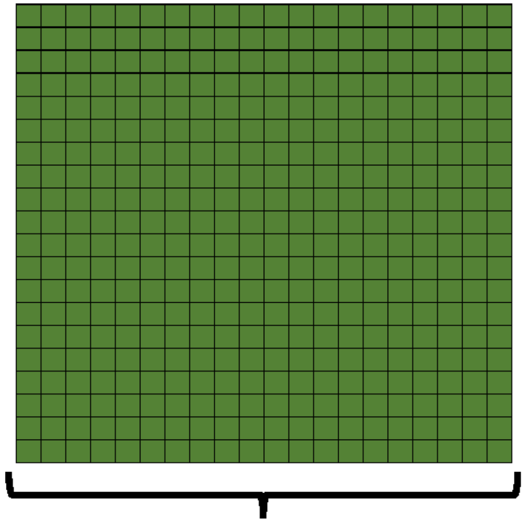
The faster model is nearly as good as the prototype model!

The Fast Model

- Prototype model: $\mathbf{U}^{\star} = \underset{\mathbf{U}}{\operatorname{argmin}} \left|\left|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}}\right|\right|_{\mathrm{F}}^{2} = \mathbf{C}^{\dagger}\mathbf{K}\left(\mathbf{C}^{\dagger}\right)^{\mathrm{T}}$
- Fast model: $\tilde{\mathbf{U}} = \underset{\mathbf{U}}{\operatorname{argmin}} \left|\left|\mathbf{P}^{\mathrm{T}}\left(\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}}\right)\mathbf{P}\right|\right|_{F}^{\mathbf{2}} = \left(\mathbf{P}^{\mathrm{T}}\mathbf{C}\right)^{\dagger}\left(\mathbf{P}^{\mathrm{T}}\mathbf{K}\mathbf{P}\right)\left(\mathbf{C}^{\mathrm{T}}\mathbf{P}\right)^{\dagger}.$
- Theory
  - $p = O\left(\sqrt{\frac{nc}{\epsilon}}\right)$
  - $\mathbf{P}$ is column selection matrix (according to the row leverage scores of $\mathbf{C}$)
  - Then $\left|\left|\mathbf{K} - \mathbf{C}\tilde{\mathbf{U}}\mathbf{C}^{\mathrm{T}}\right|\right|_{F}^{2} \leq (1+\epsilon)\left|\left|\mathbf{K} - \mathbf{C}\mathbf{U}^{\star}\mathbf{C}^{\mathrm{T}}\right|\right|_{F}^{2}$
- Overall time cost: $O(p^2 c + nc^2) = O(nc^3/\epsilon)$

linear in n

The Nyström Method

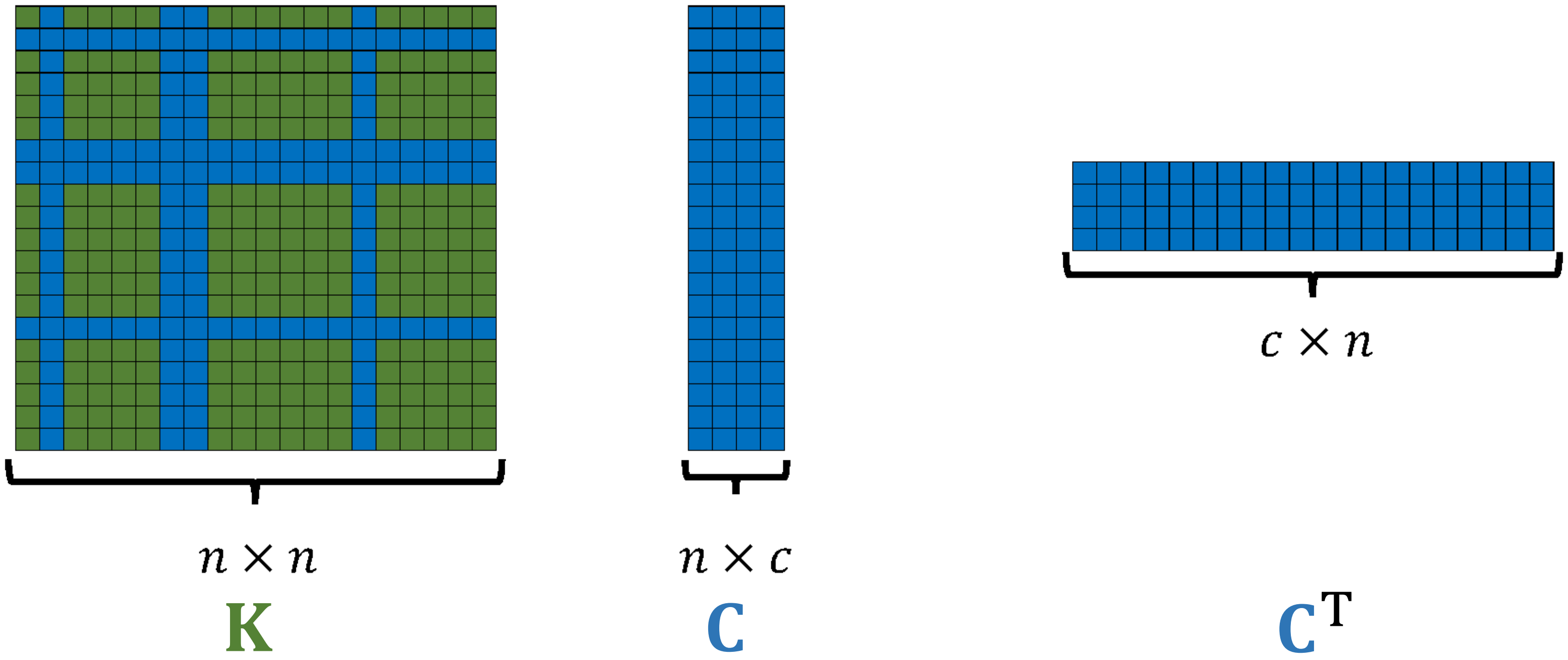


$n \times n$

$\mathbf{K}$

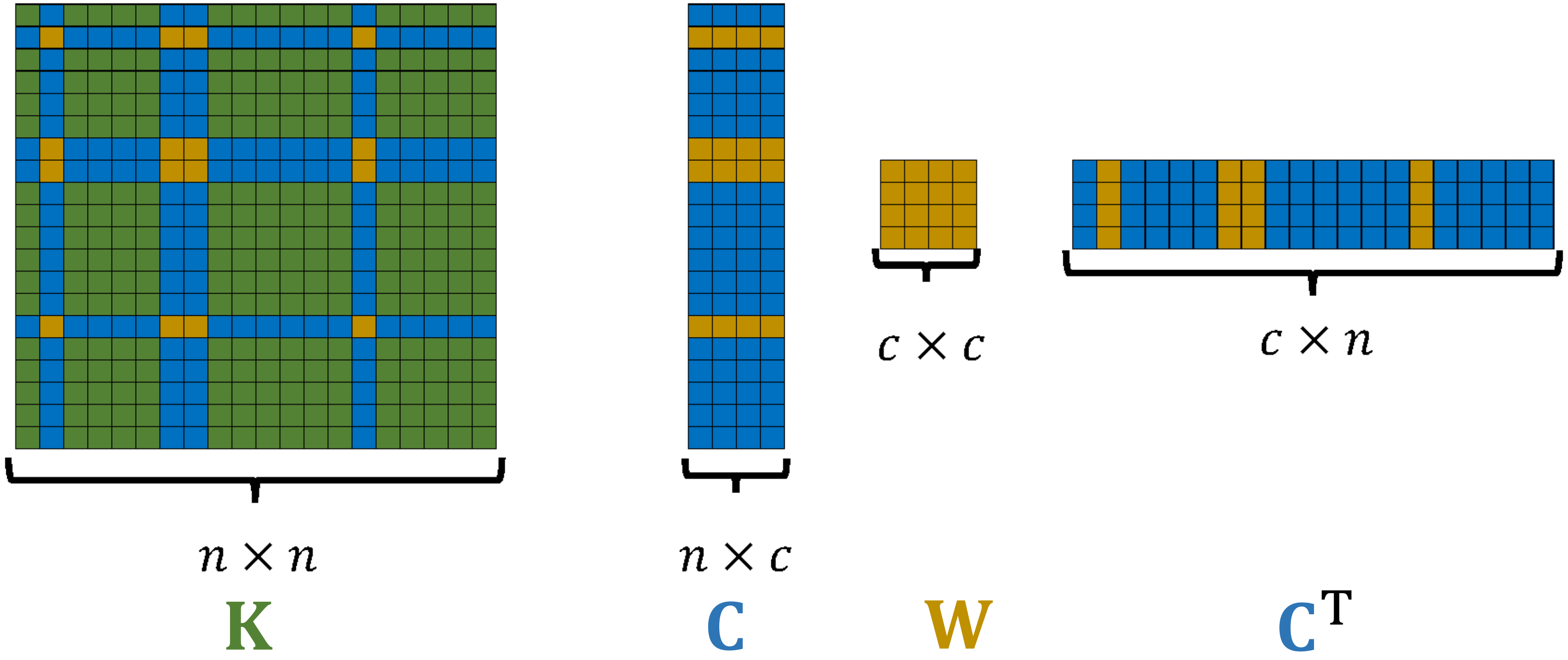
The Nyström Method

- $\mathbf{S}$ $(n \times c)$: column selection matrix
- $\mathbf{C} = \mathbf{KS}$ $(n \times c)$



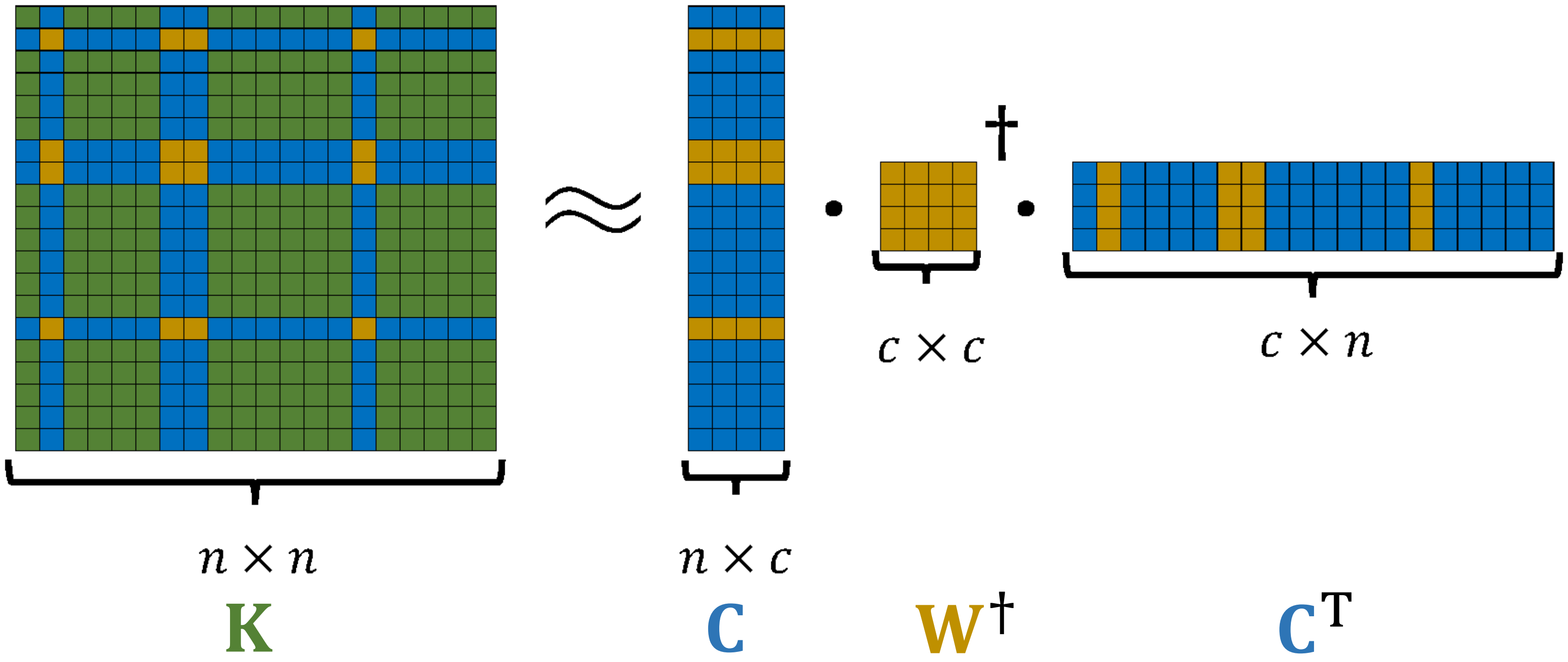
The Nyström Method

- $\mathbf{S}$ $(n \times c)$: column selection matrix
- $\mathbf{C} = \mathbf{KS}$ $(n \times c)$, $\mathbf{W} = \mathbf{S}^{\mathrm{T}}\mathbf{KS} = \mathbf{S}^{\mathrm{T}}\mathbf{C}$ $(c \times c)$



The Nyström Method

- $\mathbf{S}$ $(n \times c)$: column selection matrix
- $\mathbf{C} = \mathbf{K}\mathbf{S}$ $(n \times c)$, $\mathbf{W} = \mathbf{S}^{\mathrm{T}}\mathbf{K}\mathbf{S} = \mathbf{S}^{\mathrm{T}}\mathbf{C}$ $(c \times c)$
- The Nyström method: $\mathbf{K} \approx \mathbf{C}\,\mathbf{W}^{\dagger}\,\mathbf{C}^{\mathrm{T}}$



The Nyström Method

- $\mathbf{S}$ $(n \times c)$: column selection matrix
- $\mathbf{C} = \mathbf{K}\mathbf{S}$ $(n \times c)$, $\mathbf{W} = \mathbf{S}^{\mathrm{T}}\mathbf{K}\mathbf{S} = \mathbf{S}^{\mathrm{T}}\mathbf{C}$ $(c \times c)$
- The Nyström method: $\mathbf{K} \approx \mathbf{C}\,\mathbf{W}^{\dagger}\,\mathbf{C}^{\mathrm{T}}$
- New explanation:
    - Recall the fast model: $\tilde{\mathbf{X}} = \operatorname*{argmin}_{\mathbf{X}} \left|\left| \mathbf{P}^{\mathrm{T}} \left(\mathbf{K} - \mathbf{C}\mathbf{X}\mathbf{C}^{\mathrm{T}} \right) \mathbf{P} \right|\right|_{\mathrm{F}}^{2}$

The Nyström Method

- $\mathbf{S}$ $(n \times c)$: column selection matrix
- $\mathbf{C} = \mathbf{K}\mathbf{S}$ $(n \times c)$, $\mathbf{W} = \mathbf{S}^{\mathrm{T}}\mathbf{K}\mathbf{S} = \mathbf{S}^{\mathrm{T}}\mathbf{C}$ $(c \times c)$
- The Nyström method: $\mathbf{K} \approx \mathbf{C}\,\mathbf{W}^{\dagger}\,\mathbf{C}^{\mathrm{T}}$
- New explanation:
  - Recall the fast model: $\tilde{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \left|\left|\mathbf{P}^{\mathrm{T}}\left(\mathbf{K} - \mathbf{C}\mathbf{X}\mathbf{C}^{\mathrm{T}}\right)\mathbf{P}\right|\right|_{\mathrm{F}}^{2}$
  - Setting $\mathbf{P} = \mathbf{S}$, then

$$\tilde{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \left|\left|\mathbf{S}^{\mathrm{T}}\left(\mathbf{K} - \mathbf{C}\mathbf{X}\mathbf{C}^{\mathrm{T}}\right)\mathbf{S}\right|\right|_{\mathrm{F}}^{2}$$

The Nyström Method

- $\mathbf{S}$ $(n \times c)$: column selection matrix
- $\mathbf{C} = \mathbf{K}\mathbf{S}$ $(n \times c)$, $\mathbf{W} = \mathbf{S}^{\mathrm{T}}\mathbf{K}\mathbf{S} = \mathbf{S}^{\mathrm{T}}\mathbf{C}$ $(c \times c)$
- The Nyström method: $\mathbf{K} \approx \mathbf{C}\,\mathbf{W}^{\dagger}\,\mathbf{C}^{\mathrm{T}}$
- New explanation:
  - Recall the fast model: $\tilde{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \left|\left|\mathbf{P}^{\mathrm{T}}\left(\mathbf{K} - \mathbf{C}\mathbf{X}\mathbf{C}^{\mathrm{T}}\right)\mathbf{P}\right|\right|_{\mathrm{F}}^{2}$
  - Setting $\mathbf{P} = \mathbf{S}$, then

$$\begin{aligned}\tilde{\mathbf{X}} &= \underset{\mathbf{X}}{\operatorname{argmin}} \left|\left|\mathbf{S}^{\mathrm{T}}\left(\mathbf{K} - \mathbf{C}\mathbf{X}\mathbf{C}^{\mathrm{T}}\right)\mathbf{S}\right|\right|_{\mathrm{F}}^{2} \\ &= \left(\mathbf{S}^{\mathrm{T}}\mathbf{C}\right)^{\dagger}\left(\mathbf{S}^{\mathrm{T}}\mathbf{K}\mathbf{S}\right)\left(\mathbf{C}^{\mathrm{T}}\mathbf{S}\right)^{\dagger}\end{aligned}$$

The Nyström Method

- $\mathbf{S}$ $(n \times c)$: column selection matrix
- $\mathbf{C} = \mathbf{KS}$ $(n \times c)$, $\mathbf{W} = \mathbf{S}^{\mathrm{T}}\mathbf{KS} = \mathbf{S}^{\mathrm{T}}\mathbf{C}$ $(c \times c)$
- The Nyström method: $\mathbf{K} \approx \mathbf{C}\,\mathbf{W}^{\dagger}\,\mathbf{C}^{\mathrm{T}}$
- New explanation:
  - Recall the fast model: $\tilde{\mathbf{X}} = \operatorname*{argmin}_{\mathbf{X}} \left\| \mathbf{P}^{\mathrm{T}}(\mathbf{K} - \mathbf{C}\mathbf{X}\mathbf{C}^{\mathrm{T}})\mathbf{P} \right\|_{\mathrm{F}}^{2}$
  - Setting $\mathbf{P} = \mathbf{S}$, then

$$
\begin{aligned}
\tilde{\mathbf{X}} &= \operatorname*{argmin}_{\mathbf{X}} \left\| \mathbf{S}^{\mathrm{T}}(\mathbf{K} - \mathbf{C}\mathbf{X}\mathbf{C}^{\mathrm{T}})\mathbf{S} \right\|_{\mathrm{F}}^{2} \\
&= (\mathbf{S}^{\mathrm{T}}\mathbf{C})^{\dagger}(\mathbf{S}^{\mathrm{T}}\mathbf{K}\mathbf{S})(\mathbf{C}^{\mathrm{T}}\mathbf{S})^{\dagger} \\
&= \mathbf{W}^{\dagger}\mathbf{W}\mathbf{W}^{\dagger}
\end{aligned}
$$

The Nyström Method

- $\mathbf{S}$ $(n \times c)$: column selection matrix
- $\mathbf{C} = \mathbf{K}\mathbf{S}$ $(n \times c)$, $\mathbf{W} = \mathbf{S}^{\mathrm{T}}\mathbf{K}\mathbf{S} = \mathbf{S}^{\mathrm{T}}\mathbf{C}$ $(c \times c)$
- The Nyström method: $\mathbf{K} \approx \mathbf{C}\,\mathbf{W}^{\dagger}\,\mathbf{C}^{\mathrm{T}}$
- New explanation:
  - Recall the fast model: $\tilde{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \left|\left|\mathbf{P}^{\mathrm{T}}(\mathbf{K} - \mathbf{C}\mathbf{X}\mathbf{C}^{\mathrm{T}})\mathbf{P}\right|\right|_{\mathrm{F}}^{2}$
  - Setting $\mathbf{P} = \mathbf{S}$, then

$$\begin{aligned}\tilde{\mathbf{X}} &= \underset{\mathbf{X}}{\operatorname{argmin}} \left|\left|\mathbf{S}^{\mathrm{T}}(\mathbf{K} - \mathbf{C}\mathbf{X}\mathbf{C}^{\mathrm{T}})\mathbf{S}\right|\right|_{\mathrm{F}}^{2} \\ &= (\mathbf{S}^{\mathrm{T}}\mathbf{C})^{\dagger}(\mathbf{S}^{\mathrm{T}}\mathbf{K}\mathbf{S})(\mathbf{C}^{\mathrm{T}}\mathbf{S})^{\dagger} \\ &= \mathbf{W}^{\dagger}\mathbf{W}\mathbf{W}^{\dagger} = \mathbf{W}^{\dagger}\end{aligned}$$

The Nyström Method

- $\mathbf{S}$ $(n \times c)$: column selection matrix
- $\mathbf{C} = \mathbf{KS}$ $(n \times c)$, $\mathbf{W} = \mathbf{S}^{\mathrm{T}}\mathbf{KS} = \mathbf{S}^{\mathrm{T}}\mathbf{C}$ $(c \times c)$
- The Nyström method: $\mathbf{K} \approx \mathbf{C}\,\mathbf{W}^{\dagger}\,\mathbf{C}^{\mathrm{T}}$
- New explanation:
  - Recall the fast model: $\tilde{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \left\| \mathbf{P}^{\mathrm{T}} \left(\mathbf{K} - \mathbf{CXC}^{\mathrm{T}} \right) \mathbf{P} \right\|_{\mathrm{F}}^{2}$
  - Setting $\mathbf{P} = \mathbf{S}$, then

$$\begin{aligned} \tilde{\mathbf{X}} &= \underset{\mathbf{X}}{\operatorname{argmin}} \left\| \mathbf{S}^{\mathrm{T}} \left(\mathbf{K} - \mathbf{CXC}^{\mathrm{T}} \right) \mathbf{S} \right\|_{\mathrm{F}}^{2} \\ &= \left(\mathbf{S}^{\mathrm{T}}\mathbf{C} \right)^{\dagger} \left(\mathbf{S}^{\mathrm{T}}\mathbf{KS} \right) \left(\mathbf{C}^{\mathrm{T}}\mathbf{S} \right)^{\dagger} \\ &= \mathbf{W}^{\dagger}\mathbf{W}\mathbf{W}^{\dagger} = \mathbf{W}^{\dagger} \end{aligned}$$

- The Nystrom method is special instance of the fast model.

The Nyström Method

- $\mathbf{S}$ $(n \times c)$: column selection matrix
- $\mathbf{C} = \mathbf{KS}$ $(n \times c)$, $\mathbf{W} = \mathbf{S}^{\mathrm{T}}\mathbf{KS} = \mathbf{S}^{\mathrm{T}}\mathbf{C}$ $(c \times c)$
- The Nyström method: $\mathbf{K} \approx \mathbf{C}\,\mathbf{W}^{\dagger}\,\mathbf{C}^{\mathrm{T}}$
- New explanation:
  - Recall the fast model: $\tilde{\mathbf{X}} = \underset{\mathbf{X}}{\mathrm{argmin}} \left|\left|\mathbf{P}^{\mathrm{T}}\left(\mathbf{K} - \mathbf{CXC}^{\mathrm{T}}\right)\mathbf{P}\right|\right|_{\mathrm{F}}^{2}$
  - Setting $\mathbf{P} = \mathbf{S}$, then

$$\tilde{\mathbf{X}} = \underset{\mathbf{X}}{\mathrm{argmin}} \left|\left|\mathbf{S}^{\mathrm{T}}\left(\mathbf{K} - \mathbf{CXC}^{\mathrm{T}}\right)\mathbf{S}\right|\right|_{\mathrm{F}}^{2}$$

$$= \left(\mathbf{S}^{\mathrm{T}}\mathbf{C}\right)^{\dagger}\left(\mathbf{S}^{\mathrm{T}}\mathbf{KS}\right)\left(\mathbf{C}^{\mathrm{T}}\mathbf{S}\right)^{\dagger}$$

$$= \mathbf{W}^{\dagger}\mathbf{W}\mathbf{W}^{\dagger} = \mathbf{W}^{\dagger}$$

- The Nystrom method is special instance of the fast model.
- It is approximate solution to the prototype model

The Nyström Method

- Cost
  - Time: $O(nc^2)$
  - Memory: $O(nc)$

The Nyström Method

- Cost
  - Time: $O(nc^2)$
  - Memory: $O(nc)$

$\boxed{\text{Very efficient!}}$

The Nyström Method

- Cost
  - Time: $O(nc^2)$
  - Memory: $O(nc)$
- Error bound: weak

$\boxed{\text{Very efficient!}}$

Comparisons

- $\mathbf{C} = \mathbf{KS} \in \mathbb{R}^{n \times c}$, $\mathbf{W} = \mathbf{S}^{\mathrm{T}}\mathbf{KS} = \mathbf{S}^{\mathrm{T}}\mathbf{C} \in \mathbb{R}^{c \times c}$
- SPSD matrix approximation: $\mathbf{K} \approx \mathbf{CUC}^{\mathrm{T}}$
  - The prototype model: $\mathbf{U} = \mathbf{C}^{\dagger}\mathbf{K}\left(\mathbf{C}^{\dagger}\right)^{\mathrm{T}}$
  - The fast model: $\mathbf{U} = \left(\mathbf{P}^{\mathrm{T}}\mathbf{C}\right)^{\dagger}(\mathbf{P}^{\mathrm{T}}\mathbf{KP})\left(\mathbf{C}^{\mathrm{T}}\mathbf{P}\right)^{\dagger}$
  - The Nyström method: $\mathbf{U} = \mathbf{W}^{\dagger}$

Comparisons

- $\mathbf{C} = \mathbf{KS} \in \mathbb{R}^{n \times c}$, $\mathbf{W} = \mathbf{S}^{\mathrm{T}}\mathbf{KS} = \mathbf{S}^{\mathrm{T}}\mathbf{C} \in \mathbb{R}^{c \times c}$
- SPSD matrix approximation: $\mathbf{K} \approx \mathbf{CUC}^{\mathrm{T}}$
  - The prototype model: $\mathbf{U} = \mathbf{C}^{\dagger}\mathbf{K}\left(\mathbf{C}^{\dagger}\right)^{\mathrm{T}}$
  - The fast model: $\mathbf{U} = \left(\mathbf{P}^{\mathrm{T}}\mathbf{C}\right)^{\dagger}\left(\mathbf{P}^{\mathrm{T}}\mathbf{KP}\right)\left(\mathbf{C}^{\mathrm{T}}\mathbf{P}\right)^{\dagger}$
  - The Nyström method: $\mathbf{U} = \mathbf{W}^{\dagger}$

When $\mathbf{P} = \mathbf{I}_n$, the prototype model $\Leftrightarrow$ the fast model

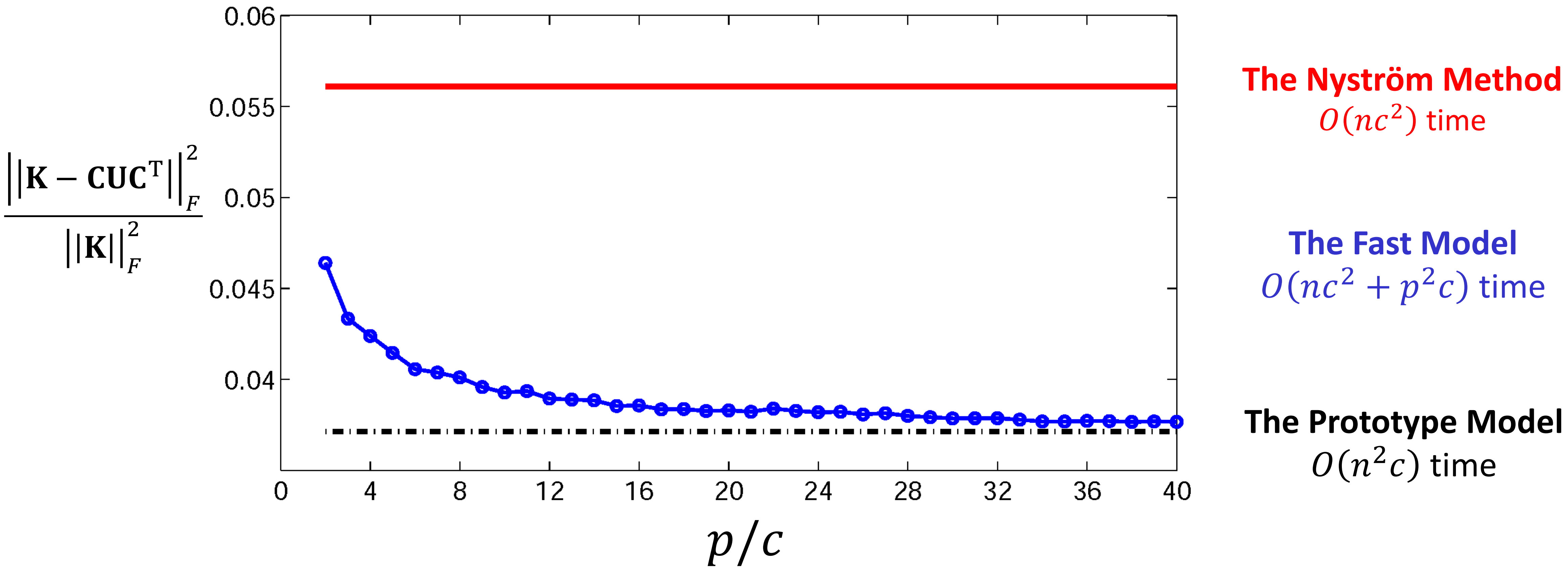
Comparisons

- $\mathbf{C} = \mathbf{K}\mathbf{S} \in \mathbb{R}^{n \times c}$, $\mathbf{W} = \mathbf{S}^{\mathrm{T}}\mathbf{K}\mathbf{S} = \mathbf{S}^{\mathrm{T}}\mathbf{C} \in \mathbb{R}^{c \times c}$
- SPSD matrix approximation: $\mathbf{K} \approx \mathbf{C}\mathbf{U}\mathbf{C}^{\mathrm{T}}$
  - The prototype model: $\mathbf{U} = \mathbf{C}^{\dagger}\mathbf{K}\left(\mathbf{C}^{\dagger}\right)^{\mathrm{T}}$
  - The fast model: $\mathbf{U} = \left(\mathbf{P}^{\mathrm{T}}\mathbf{C}\right)^{\dagger}(\mathbf{P}^{\mathrm{T}}\mathbf{K}\mathbf{P})\left(\mathbf{C}^{\mathrm{T}}\mathbf{P}\right)^{\dagger}$
  - The Nyström method: $\mathbf{U} = \mathbf{W}^{\dagger}$

When $\mathbf{P} = \mathbf{S}$, the Nyström method $\Leftrightarrow$ the fast model

Comparisons

- $c = 150$, $n = 100c$, vary p from $2c$ to $40c$



Conclusions

- Motivations
  - Avoid forming the kernel matrix
  - Avoid inversion/decomposition
- Prototype model, fast model, Nystrom
  - They have connections
  - The fast model and Nystrom are practical