RECENT ADVANCES IN POSITIVE SEMIDEFINITE MATRIX APPROXIMATION

Cameron Musco (Microsoft Research)

Simons Workshop on RandNLA & Applications

POSITIVE SEMIDEFINITE MATRICES

Symmetric $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive semidefinite if it has all non-negative eigenvalues.

POSITIVE SEMIDEFINITE MATRICES

Symmetric $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive semidefinite if it has all non-negative eigenvalues.

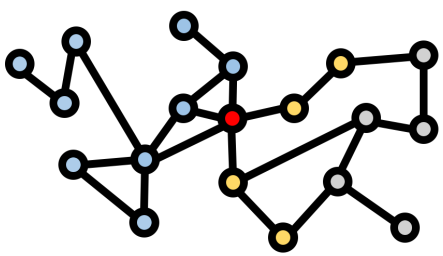
$$\lambda_i(\mathbf{A}) \geq 0, \forall\, i \iff \mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^n.$$

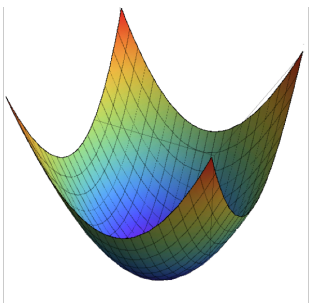
POSITIVE SEMIDEFINITE MATRICES

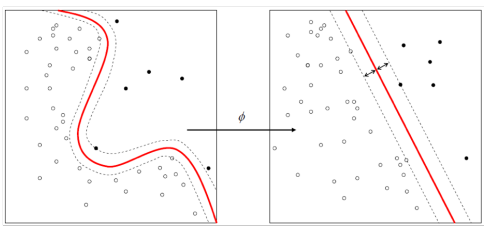
Symmetric $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive semidefinite if it has all non-negative eigenvalues.

$$\lambda_i(\mathbf{A}) \geq 0, \forall\, i \iff \mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^n.$$

- Include graph Laplacians, Gram matrices and kernel matrices, covariance matrices, Hessians for convex functions, etc.







POSITIVE SEMIDEFINITE MATRICES

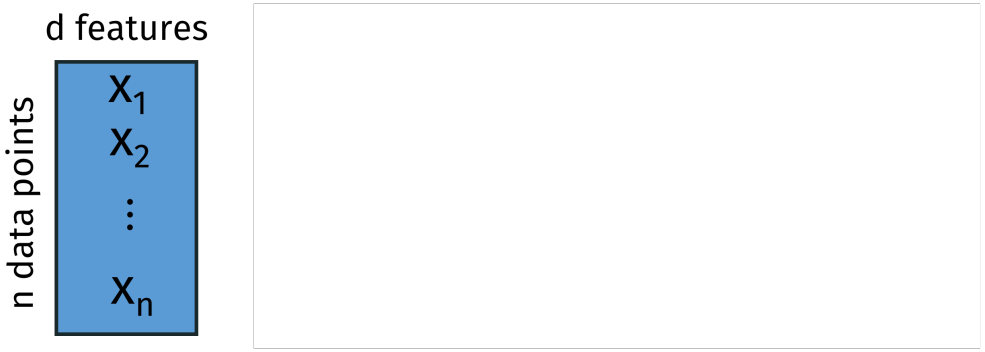
- Massive, dense positive semidefinite matrices are often generated from much less massive datasets.

POSITIVE SEMIDEFINITE MATRICES

- Massive, dense positive semidefinite matrices are often generated from much less massive datasets.
- E.g. for kernel methods like kernel SVM, kernel ridge regression (Gaussian process regresssion, kriging), kernel distance measures (maximum mean discrepancy).

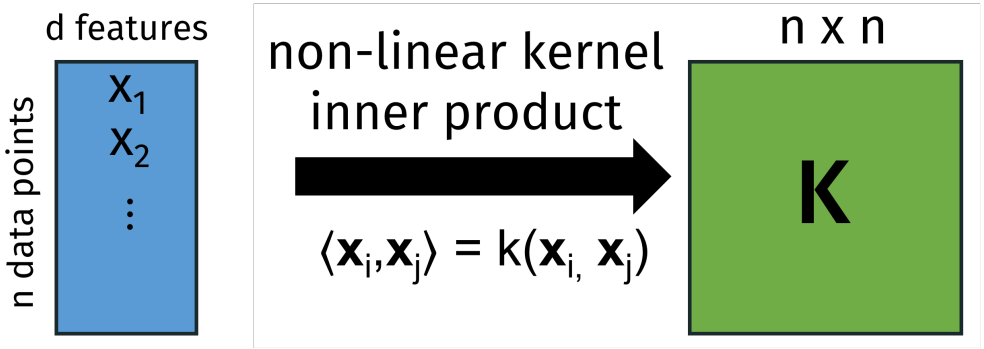
POSITIVE SEMIDEFINITE MATRICES

- Massive, dense positive semidefinite matrices are often generated from much less massive datasets.
- E.g. for kernel methods like kernel SVM, kernel ridge regression (Gaussian process regresssion, kriging), kernel distance measures (maximum mean discrepancy).



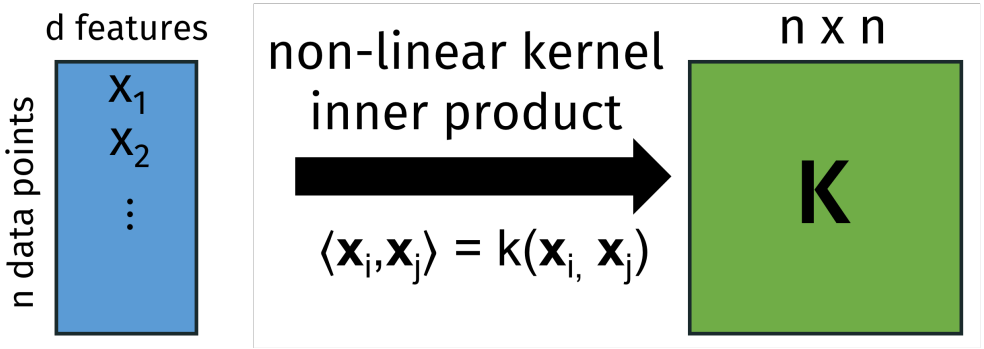
POSITIVE SEMIDEFINITE MATRICES

- Massive, dense positive semidefinite matrices are often generated from much less massive datasets.
- E.g. for kernel methods like kernel SVM, kernel ridge regression (Gaussian process regresssion, kriging), kernel distance measures (maximum mean discrepancy).



POSITIVE SEMIDEFINITE MATRICES

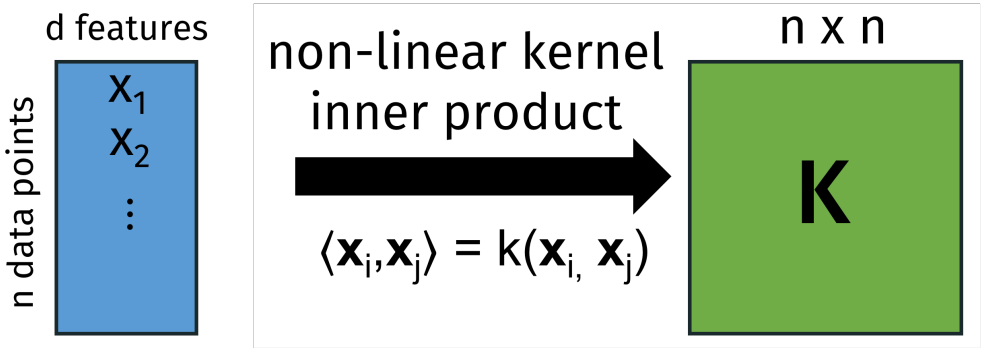
- Massive, dense positive semidefinite matrices are often generated from much less massive datasets.
- E.g. for kernel methods like kernel SVM, kernel ridge regression (Gaussian process regresssion, kriging), kernel distance measures (maximum mean discrepancy).



- Even writing down **K** or performing a single iteration of an iterative solver takes $\Omega(n^2)$ time.

POSITIVE SEMIDEFINITE MATRICES

- Massive, dense positive semidefinite matrices are often generated from much less massive datasets.
- E.g. for kernel methods like kernel SVM, kernel ridge regression (Gaussian process regresssion, kriging), kernel distance measures (maximum mean discrepancy).



- Even writing down $\mathbf{K}$ or performing a single iteration of an iterative solver takes $\Omega(n^2)$ time.
- For $n = 100,000$, $\mathbf{K}$ has 10 billion entries. Takes 80 GB of storage if each is a `double`.

HANDLING LARGE PSD MATRICES

HANDLING LARGE PSD MATRICES

Option 1: Massive computation/parallelization.

HANDLING LARGE PSD MATRICES

Option 1: Massive computation/parallelization.

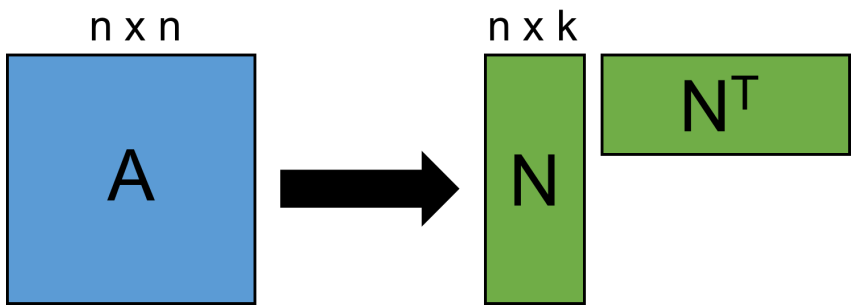
Option 2: Develop effective approximation methods.

HANDLING LARGE PSD MATRICES

Option 1: Massive computation/parallelization.

Option 2: Develop effective approximation methods.

- E.g., exploit low-rank structure.



PSD MATRIX APPROXIMATION

- Traditional low-rank approximation methods (full SVD, Krylov subspace methods, random projection) run in $\Omega(\text{nnz}(\mathbf{A}))$ time ($\Omega(n^2)$ for dense matrices).

PSD MATRIX APPROXIMATION

- Traditional low-rank approximation methods (full SVD, Krylov subspace methods, random projection) run in $\Omega(\text{nnz}(\mathbf{A}))$ time ($\Omega(n^2)$ for dense matrices).
- Many $o(n^2)$ (i.e. sublinear) time methods have been studied:
  - Incomplete Cholesky factorization (Fine & Scheinberg '02, Bach & Jordan '02).
  - Entrywise sampling (Achlioptas, McSherry, & Schölkopf '01).
  - Nyström approximation (Williams & Seeger '01, Drineas & Mahoney '05, Gittens & Mahoney '13).
  - Random Fourier features (Rahimi & Recht '07).

PSD MATRIX APPROXIMATION

- Traditional low-rank approximation methods (full SVD, Krylov subspace methods, random projection) run in $\Omega(\text{nnz}(\mathbf{A}))$ time ($\Omega(n^2)$ for dense matrices).
- Many $o(n^2)$ (i.e. sublinear) time methods have been studied:
  - Incomplete Cholesky factorization (Fine & Scheinberg '02, Bach & Jordan '02).
  - Entrywise sampling (Achlioptas, McSherry, & Schölkopf '01).
  - Nyström approximation (Williams & Seeger '01, Drineas & Mahoney '05, Gittens & Mahoney '13).
  - Random Fourier features (Rahimi & Recht '07).

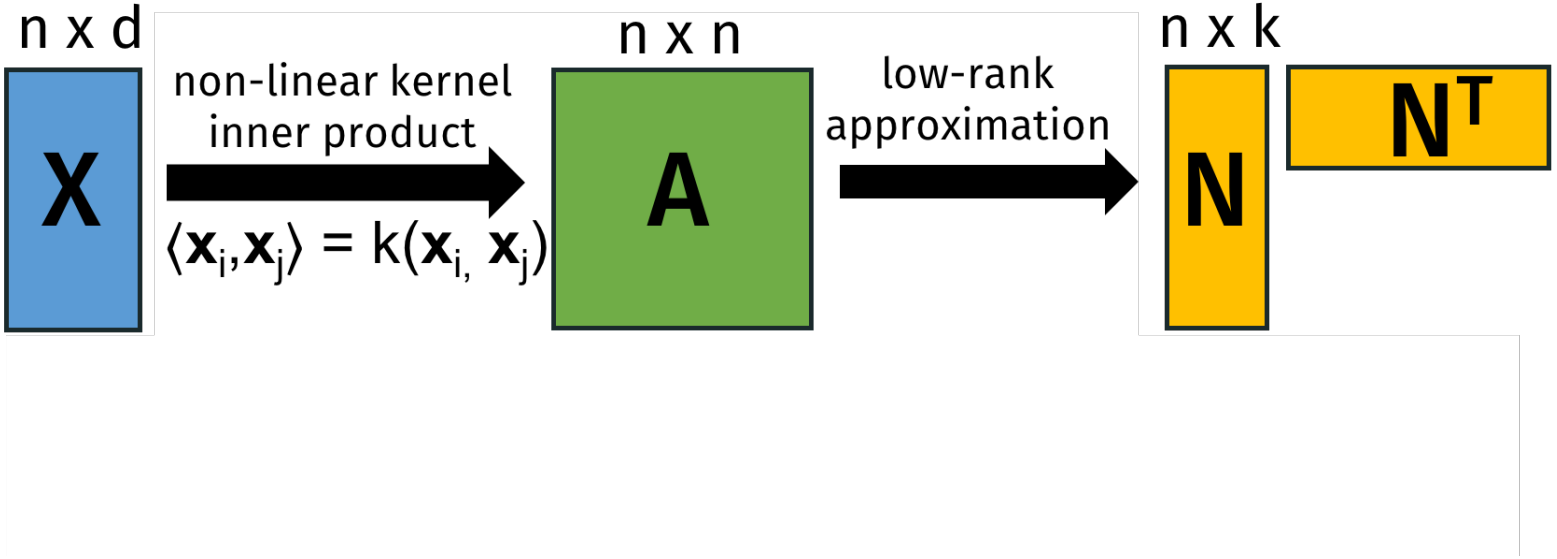
PSD MATRIX APPROXIMATION

- Traditional low-rank approximation methods (full SVD, Krylov subspace methods, random projection) run in $\Omega(\text{nnz}(\mathbf{A}))$ time ($\Omega(n^2)$ for dense matrices).
- Many $o(n^2)$ (i.e. sublinear) time methods have been studied:
  - Incomplete Cholesky factorization (Fine & Scheinberg '02, Bach & Jordan '02).
  - Entrywise sampling (Achlioptas, McSherry, & Schölkopf '01).
  - Nyström approximation (Williams & Seeger '01, Drineas & Mahoney '05, Gittens & Mahoney '13).
  - Random Fourier features (Rahimi & Recht '07).
- A variety of approximation bounds, some under assumptions. Nothing as strong as e.g., $\|\mathbf{A} - \mathbf{N}\mathbf{N}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2$.

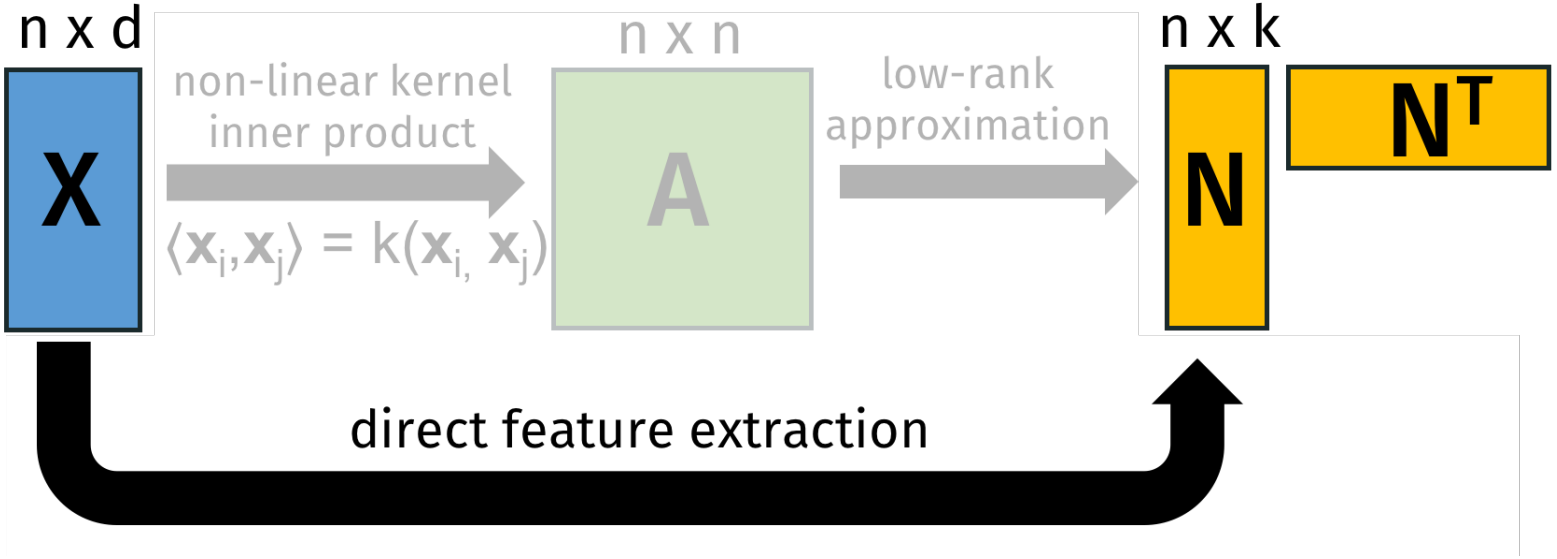
Our Goal: Give strong worst case approximation bounds for sublinear time (i.e., $o(n^2)$) methods. Build a new toolkit for PSD matrix algebra in the process.

PSD MATRIX APPROXIMATION

Our Goal: Give strong worst case approximation bounds for sublinear time (i.e., $o(n^2)$) methods. Build a new toolkit for PSD matrix algebra in the process.



Our Goal: Give strong worst case approximation bounds for sublinear time (i.e., $o(n^2)$) methods. Build a new toolkit for PSD matrix algebra in the process.



SUBLINEAR TIME BARRIER FOR GENERAL MATRICES

For general matrices, $\Omega(\text{nnz}(A))$ time is required for any non-trivial matrix approximation.

SUBLINEAR TIME BARRIER FOR GENERAL MATRICES

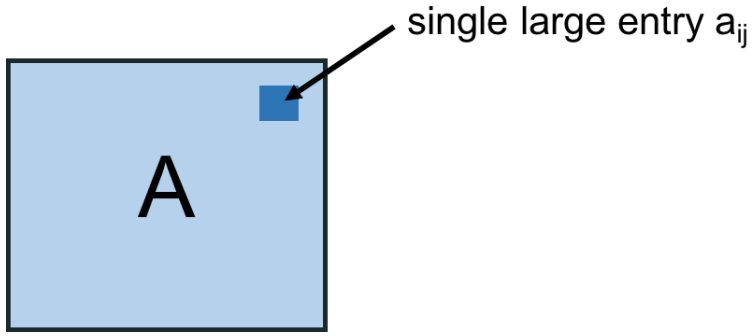
For general matrices, $\Omega(\text{nnz}(A))$ time is required for any non-trivial matrix approximation.

- Randomly place a single entry that dominates A's Frobenius norm.

SUBLINEAR TIME BARRIER FOR GENERAL MATRICES

For general matrices, $\Omega(\text{nnz}(A))$ time is required for any non-trivial matrix approximation.

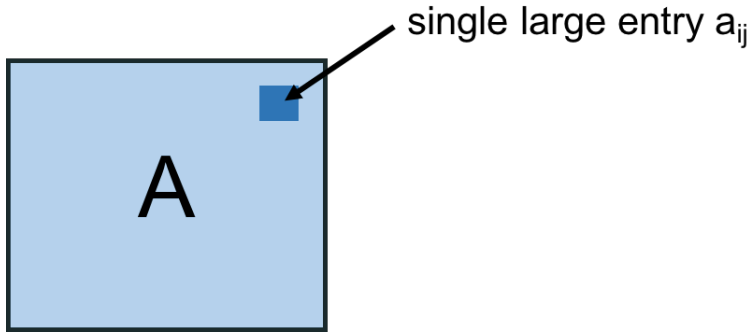
- Randomly place a single entry that dominates A's Frobenius norm.



SUBLINEAR TIME BARRIER FOR GENERAL MATRICES

For general matrices, $\Omega(\text{nnz}(A))$ time is required for any non-trivial matrix approximation.

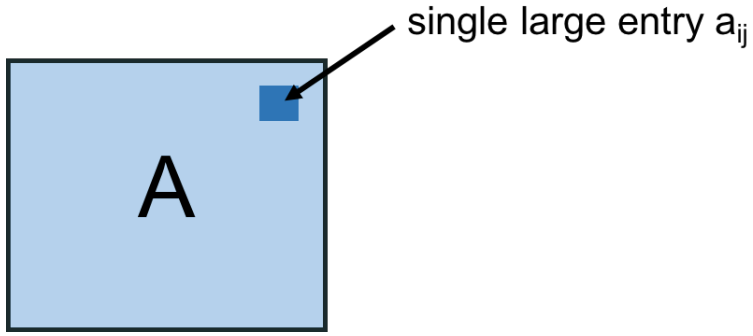
- Randomly place a single entry that dominates A's Frobenius norm.
- Finding it with constant probability requires reading at least a constant fraction of the non-zero entries in A (i.e., $\Omega(\text{nnz}(A))$ time).



SUBLINEAR TIME BARRIER FOR GENERAL MATRICES

For general matrices, $\Omega(\text{nnz}(\mathbf{A}))$ time is required for any non-trivial matrix approximation.

- Randomly place a single entry that dominates $\mathbf{A}$'s Frobenius norm.
- Finding it with constant probability requires reading at least a constant fraction of the non-zero entries in $\mathbf{A}$ (i.e., $\Omega(\text{nnz}(\mathbf{A}))$ time).



- Rules out, e.g., for any $\Delta < 1$, approximation of the form:

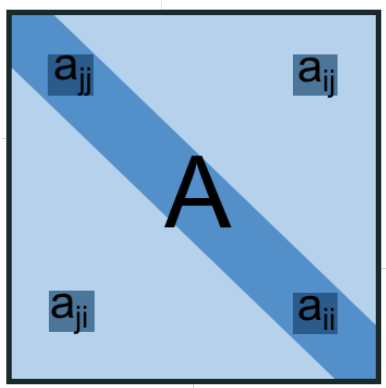
$$\|\mathbf{A} - \mathbf{N}\mathbf{N}^T\|_F \leq \Delta \|\mathbf{A}\|_F^2.$$

WHAT ABOUT FOR PSD MATRICES?

Observation: For PSD $\mathbf{A}$, we have for any entry $\mathbf{a}_{ij}$:

$$\mathbf{a}_{ij} \leq \max(\mathbf{a}_{ii}, \mathbf{a}_{jj})$$

since otherwise $(\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{A}(\mathbf{e}_i - \mathbf{e}_j) < 0$.



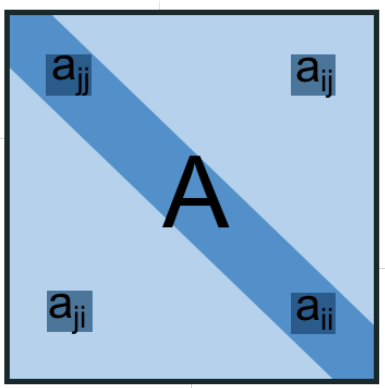
WHAT ABOUT FOR PSD MATRICES?

Observation: For PSD $\mathbf{A}$, we have for any entry $\mathbf{a}_{ij}$:

$$\mathbf{a}_{ij} \leq \max(\mathbf{a}_{ii}, \mathbf{a}_{jj})$$

since otherwise $(\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{A}(\mathbf{e}_i - \mathbf{e}_j) < 0$.

- So we can find any 'hidden' heavy entry by looking at its corresponding diagonal entries.



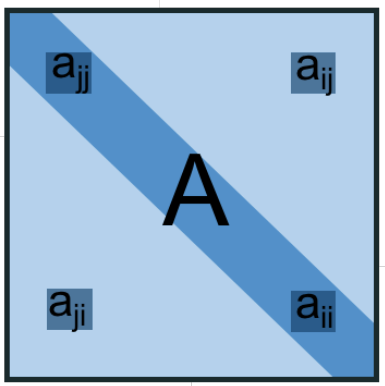
WHAT ABOUT FOR PSD MATRICES?

Observation: For PSD $\mathbf{A}$, we have for any entry $\mathbf{a}_{ij}$:

$$\mathbf{a}_{ij} \leq \max(\mathbf{a}_{ii}, \mathbf{a}_{jj})$$

since otherwise $(\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{A}(\mathbf{e}_i - \mathbf{e}_j) < 0$.

- So we can find any 'hidden' heavy entry by looking at its corresponding diagonal entries.



Question: How can we use additional structure arising from PSD-ness to achieve non-trivial approximation in sublinear time?

EVERY PSD MATRIX IS A GRAM MATRIX

Very Simple Fact: Every PSD matrix $\mathbf{A} \in R^{n \times n}$ can be written as $\mathbf{B}^T\mathbf{B}$ for some $\mathbf{B} \in \mathbb{R}^{n \times n}$.

EVERY PSD MATRIX IS A GRAM MATRIX

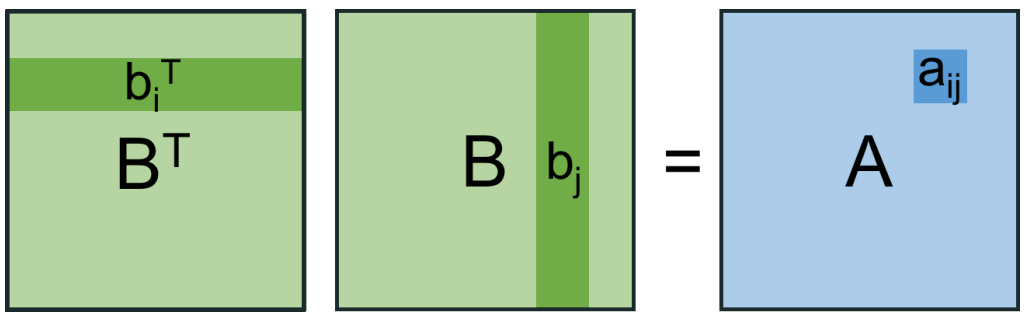
Very Simple Fact: Every PSD matrix $\mathbf{A} \in R^{n \times n}$ can be written as $\mathbf{B}^T\mathbf{B}$ for some $\mathbf{B} \in \mathbb{R}^{n \times n}$.

- $\mathbf{B}$ can be any matrix square root of $\mathbf{A}$, e.g. if we let $\mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$ be the eigendecomposition of $\mathbf{A}$, we can set $\mathbf{B} = \mathbf{\Sigma}^{1/2}\mathbf{V}^T$.

EVERY PSD MATRIX IS A GRAM MATRIX

Very Simple Fact: Every PSD matrix $\mathbf{A} \in R^{n \times n}$ can be written as $\mathbf{B}^T\mathbf{B}$ for some $\mathbf{B} \in \mathbb{R}^{n \times n}$.

- $\mathbf{B}$ can be any matrix square root of $\mathbf{A}$, e.g. if we let $\mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$ be the eigendecomposition of $\mathbf{A}$, we can set $\mathbf{B} = \mathbf{\Sigma}^{1/2}\mathbf{V}^T$.
- Letting $\mathbf{b}_1, ..., \mathbf{b}_n$ be the columns of $\mathbf{B}$, the entries of $\mathbf{A}$ contain every pairwise dot product $\mathbf{a}_{ij} = \mathbf{b}_i^T\mathbf{b}_j$.



EVERY PSD MATRIX IS A GRAM MATRIX

The fact that $\mathbf{A}$ is a Gram matrix places a variety of geometric constraints on its entries.

EVERY PSD MATRIX IS A GRAM MATRIX

The fact that $\mathbf{A}$ is a Gram matrix places a variety of geometric constraints on its entries.

- The heavy diagonal observation is just one example. By Cauchy-Schwarz:

$$\mathbf{a}_{ij} = \mathbf{b}_i^T \mathbf{b}_j \leq \sqrt{(\mathbf{b}_i^T \mathbf{b}_i) \cdot (\mathbf{b}_j^T \mathbf{b}_j)} = \sqrt{\mathbf{a}_{ii} \cdot \mathbf{a}_{jj}} \leq \max(\mathbf{a}_{ii}, \mathbf{a}_{jj}).$$

EVERY PSD MATRIX IS A GRAM MATRIX

The fact that $\mathbf{A}$ is a Gram matrix places a variety of geometric constraints on its entries.

- The heavy diagonal observation is just one example. By Cauchy-Schwarz:

$$\mathbf{a}_{ij} = \mathbf{b}_i^T \mathbf{b}_j \leq \sqrt{(\mathbf{b}_i^T \mathbf{b}_i) \cdot (\mathbf{b}_j^T \mathbf{b}_j)} = \sqrt{\mathbf{a}_{ii} \cdot \mathbf{a}_{jj}} \leq \max(\mathbf{a}_{ii}, \mathbf{a}_{jj}).$$

Another View: $\mathbf{A}$ contains a lot of information about the column span of $\mathbf{B}$ in a very compressed form – with every pairwise dot product stored as $\mathbf{a}_{ij}$.

FACTOR MATRIX LOW-RANK APPROXIMATION

Good News: It is possible to find a low-rank approximation of $\mathbf{B}$ using $o(n^2)$ column dot products, i.e. $o(n^2)$ accesses to $\mathbf{A}$!

Good News: It is possible to find a low-rank approximation of B using $o(n^2)$ column dot products, i.e. $o(n^2)$ accesses to A!

What does this buy us? B has the same (right) singular vectors as A, and its singular values are given by $\sigma_i(\mathsf{B}) = \sqrt{\sigma_i(\mathsf{A})}$.

Good News: It is possible to find a low-rank approximation of $\mathbf{B}$ using $o(n^2)$ column dot products, i.e. $o(n^2)$ accesses to $\mathbf{A}$!

What does this buy us? $\mathbf{B}$ has the same (right) singular vectors as $\mathbf{A}$, and its singular values are given by $\sigma_i(\mathbf{B}) = \sqrt{\sigma_i(\mathbf{A})}$.

- The top k singular vectors are the same for the two matrices, so low-rank approximation of $\mathbf{B}$ is closely related to that of $\mathbf{A}$.

Good News: It is possible to find a low-rank approximation of B using $o(n^2)$ column dot products, i.e. $o(n^2)$ accesses to A!

What does this buy us? B has the same (right) singular vectors as A, and its singular values are given by $\sigma_i(\mathsf{B}) = \sqrt{\sigma_i(\mathsf{A})}$.

- The top k singular vectors are the same for the two matrices, so low-rank approximation of B is closely related to that of A.
- E.g. an optimal low-rank approximation for B gives an optimal low-rank approximation for $\mathsf{A} = \mathsf{B}^T\mathsf{B}$.

Low-rank approximation via adaptive sampling

Theorem (Deshpande, Vempala '06)

For any $\mathsf{B} \in \mathbb{R}^{n \times n}$, *there exists a subset of* $\tilde{O}(k^2/\epsilon)$ *columns whose span contains* $\mathsf{Z} \in \mathbb{R}^{n \times k}$ *satisfying:*

$$\|\mathsf{B} - \mathsf{Z}\mathsf{Z}^T\mathsf{B}\|_F \leq (1+\epsilon)\|\mathsf{B} - \mathsf{B}_k\|_F$$

Theorem (Deshpande, Vempala '06)

For any $\mathbf{B} \in \mathbb{R}^{n \times n}$, *there exists a subset of* $\tilde{O}(k^2/\epsilon)$ *columns whose span contains* $\mathbf{Z} \in \mathbb{R}^{n \times k}$ *satisfying:*

$$\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F \leq (1+\epsilon)\|\mathbf{B} - \mathbf{B}_k\|_F$$

Adaptive Sampling

Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n}\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$.

Theorem (Deshpande, Vempala '06)

For any $\mathbf{B} \in \mathbb{R}^{n \times n}$, *there exists a subset of* $\tilde{O}(k^2/\epsilon)$ *columns whose span contains* $\mathbf{Z} \in \mathbb{R}^{n \times k}$ *satisfying:*

$$\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F \leq (1+\epsilon)\|\mathbf{B} - \mathbf{B}_k\|_F$$

Adaptive Sampling

Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n}\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$.

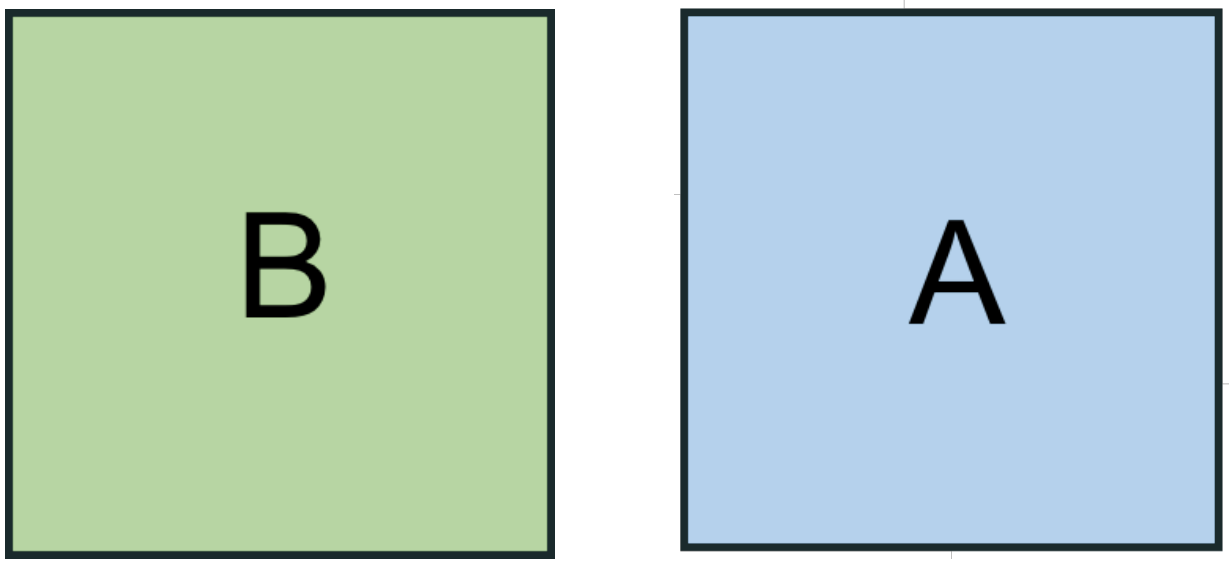
Adaptive Sampling

Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n} \|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$.



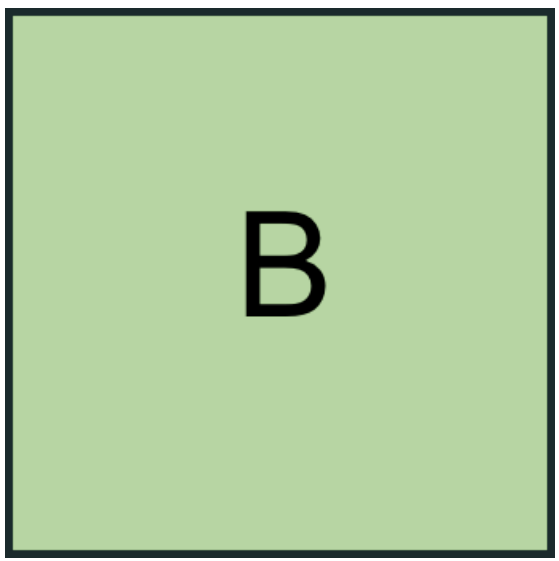
Adaptive Sampling

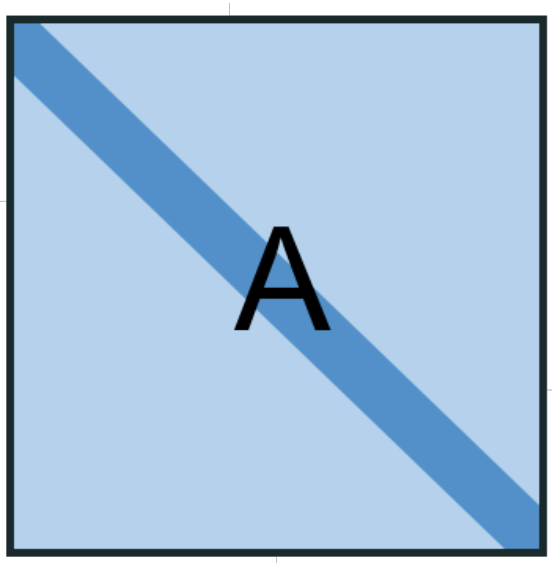
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n}\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2} = \frac{\|\mathbf{b}_i\|^2}{\sum_{i=1}^{n}\|\mathbf{b}_i\|^2} = \frac{\mathbf{a}_{ii}}{\mathrm{tr}(\mathbf{A})}$.





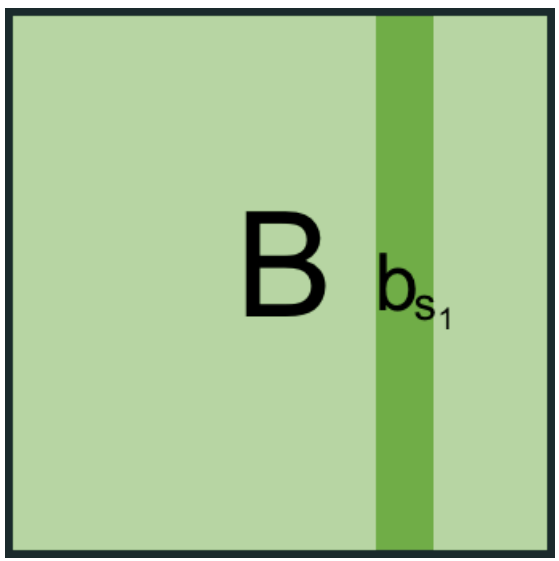
Adaptive Sampling

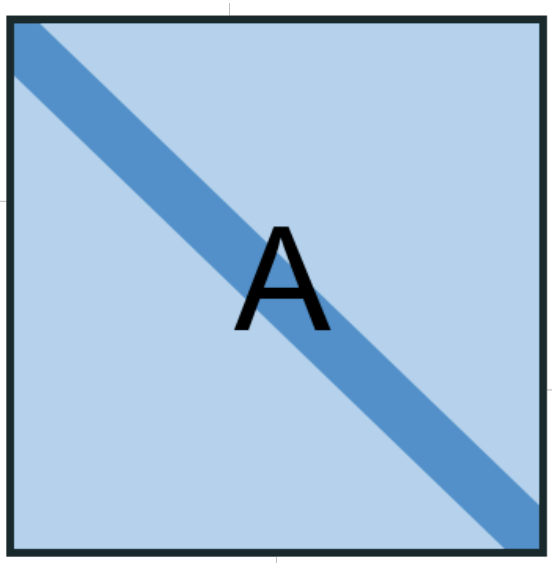
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n}\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2} = \frac{\|\mathbf{b}_i\|^2}{\sum_{i=1}^{n}\|\mathbf{b}_i\|^2} = \frac{\mathbf{a}_{ii}}{\mathrm{tr}(\mathbf{A})}$.





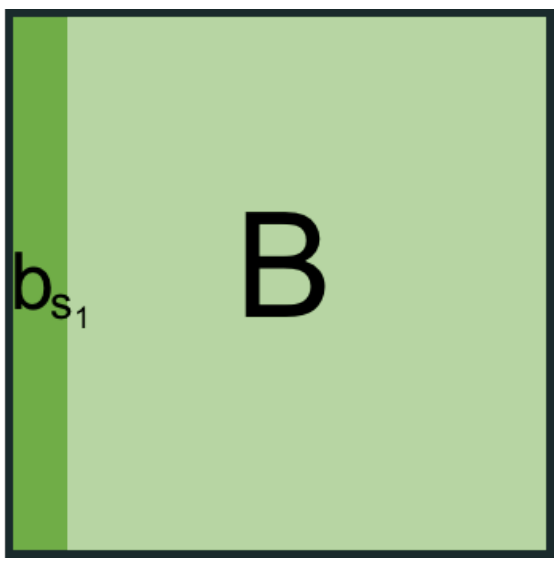
Adaptive Sampling

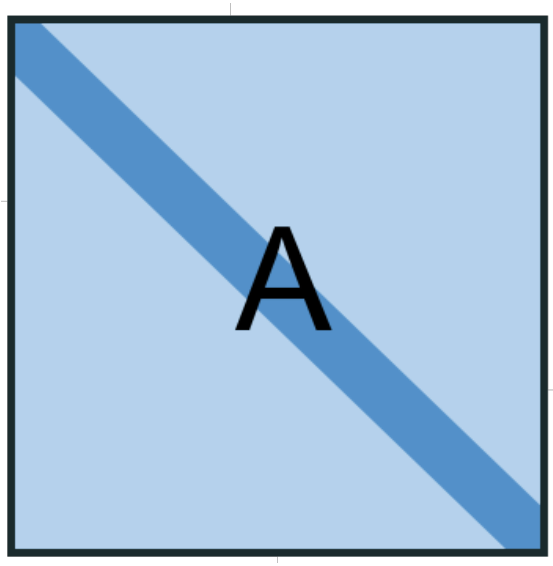
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n}\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2} = \frac{\|\mathbf{b}_i\|^2}{\sum_{i=1}^{n}\|\mathbf{b}_i\|^2} = \frac{a_{ii}}{\mathrm{tr}(A)}$.





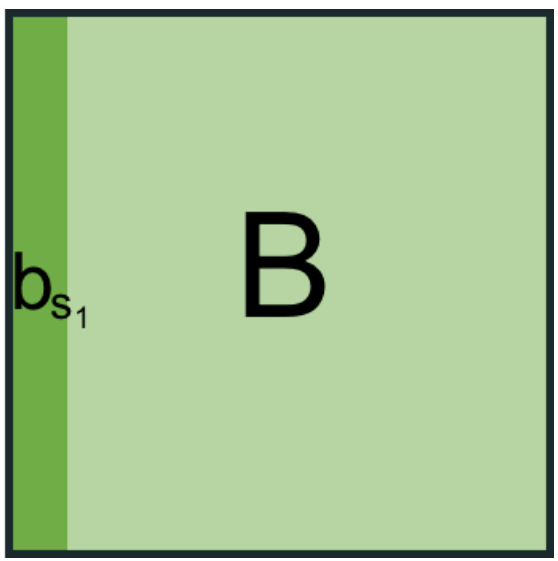
Adaptive Sampling

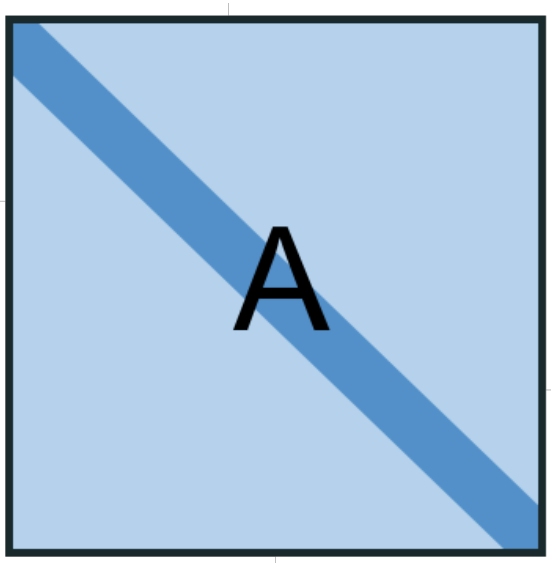
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n} \|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$.





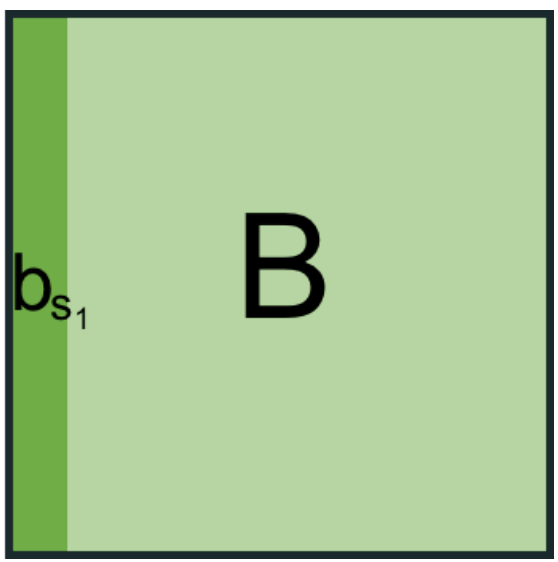
Adaptive Sampling

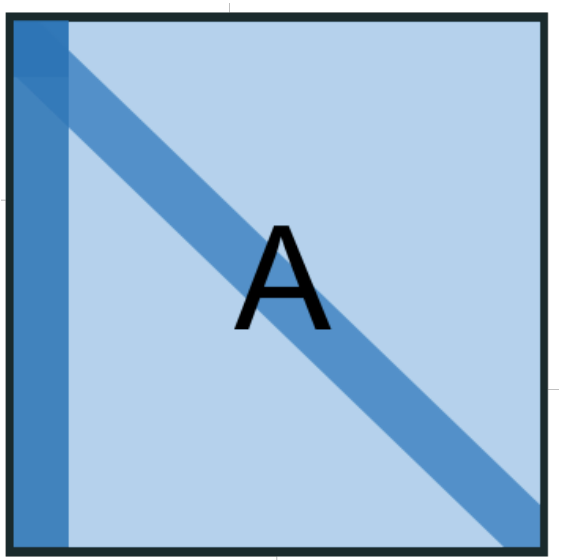
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n} \|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$.





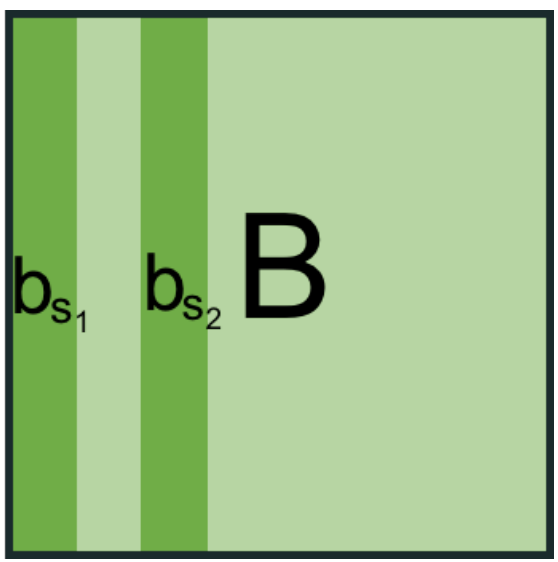
Adaptive Sampling

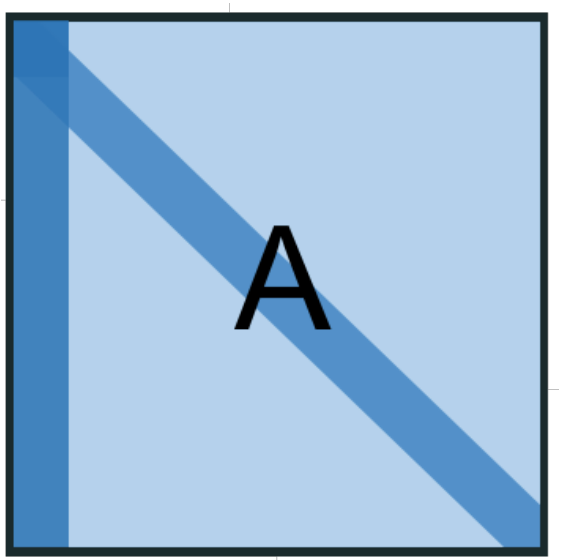
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n} \|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$.





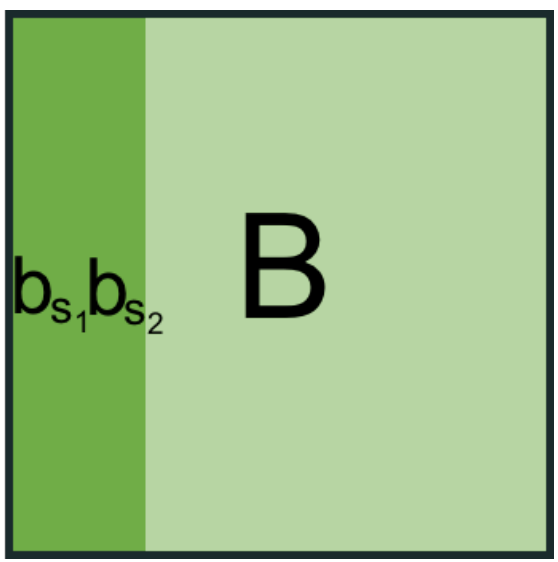
Adaptive Sampling

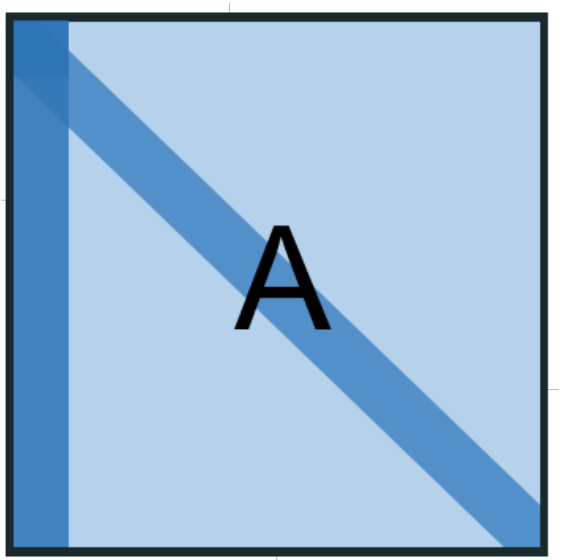
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n} \|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$.





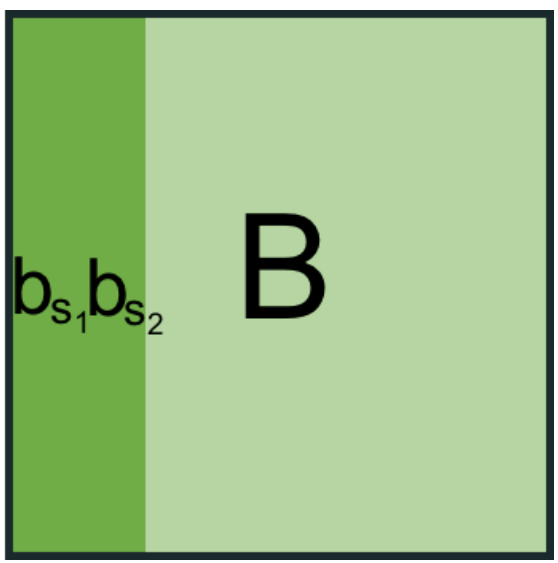
Adaptive Sampling

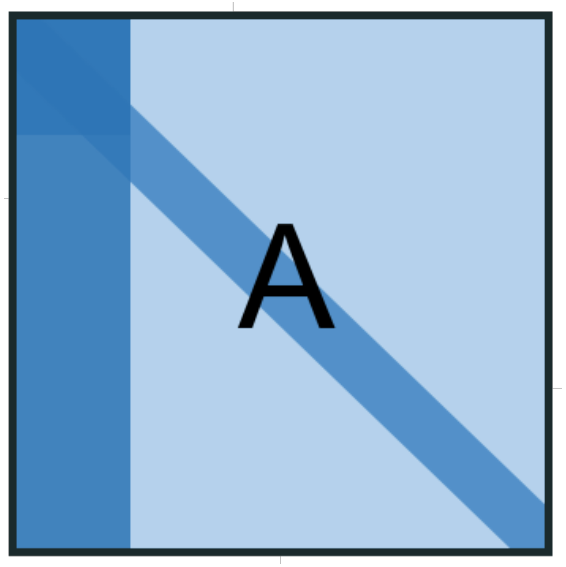
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n} \|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$.





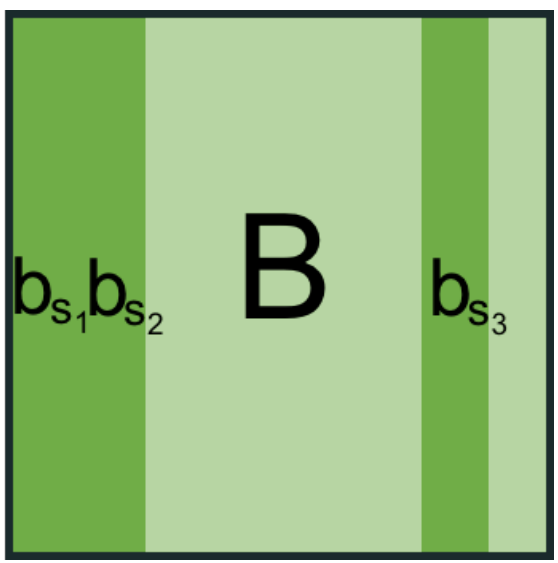
Adaptive Sampling

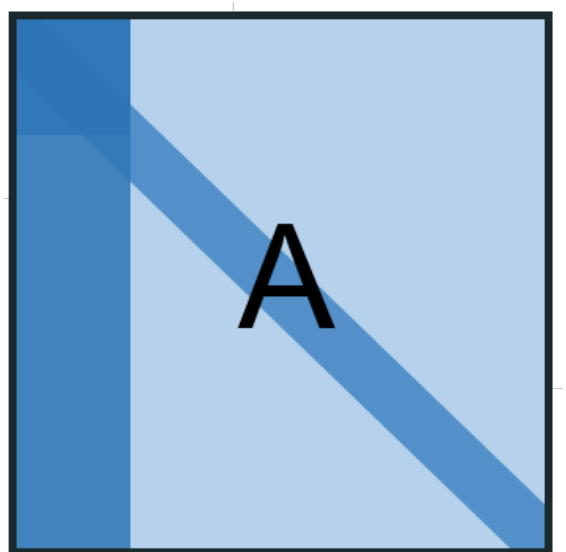
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n} \|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$.





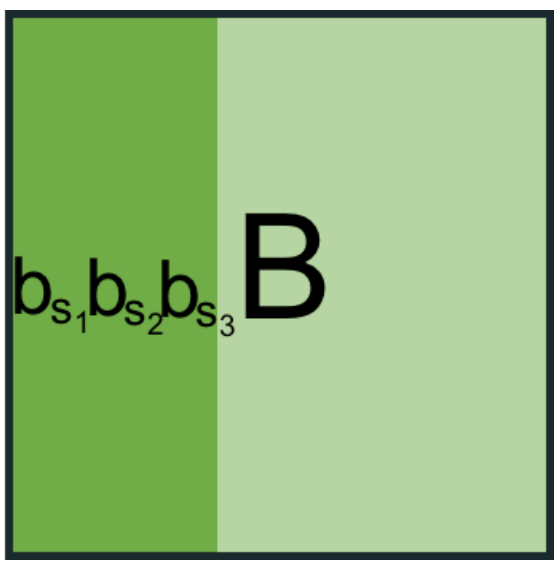
Adaptive Sampling

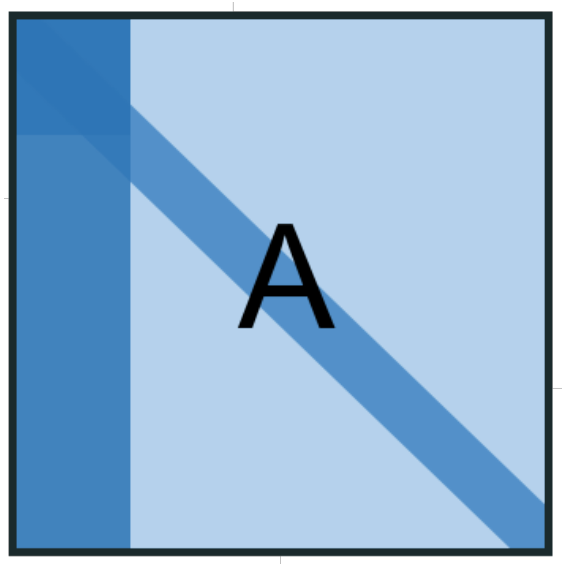
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n} \|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$.





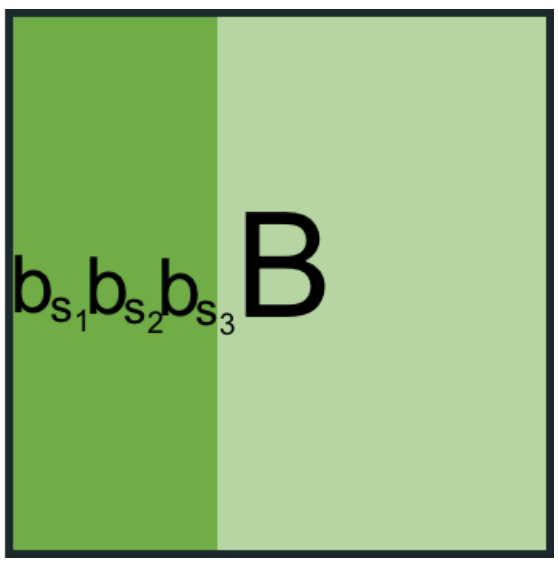
Adaptive Sampling

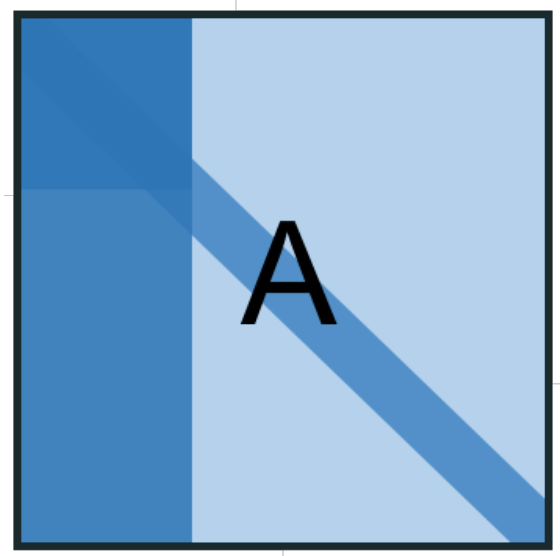
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

For $t = 1, \ldots, \tilde{O}(k^2/\epsilon)$

Let $\mathbf{P}_{\mathcal{S}}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^{n} \|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$.





Theorem (Factor Matrix Low-Rank Approximation)

There is an algorithm using $\tilde{O}(nk^2/\epsilon)$ accesses to $\mathbf{A} = \mathbf{B}^T\mathbf{B}$ that computes $\mathbf{Z} \in \mathbb{R}^{n\times k}$ satisfying with probability 99/100:

$$\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F \leq (1+\epsilon)\|\mathbf{B} - \mathbf{B}_k\|_F.$$

.

Theorem (Factor Matrix Low-Rank Approximation)

There is an algorithm using $\tilde{O}(nk^2/\epsilon)$ *accesses to* $\mathbf{A} = \mathbf{B}^T\mathbf{B}$ *that computes* $\mathbf{Z} \in \mathbb{R}^{n\times k}$ *satisfying with probability* 99/100*:*

$$\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F \leq (1+\epsilon)\|\mathbf{B} - \mathbf{B}_k\|_F.$$

- Can be improved using leverage-score-based Nyström approximation [Alaoui, Mahoney '15], [Musco, Musco '17].

Theorem (Factor Matrix Low-Rank Approximation)

There is an algorithm using $\tilde{O}(nk/\epsilon)$ accesses to $\mathbf{A}$ that computes $\mathbf{Z} \in \mathbb{R}^{n \times k}$ satisfying with high probability:

$$\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F \leq (1+\epsilon)\|\mathbf{B} - \mathbf{B}_k\|_F.$$

- Can be improved using leverage-score-based Nyström approximation [Alaoui, Mahoney '15], [Musco, Musco '17].

Theorem (Factor Matrix Low-Rank Approximation)

There is an algorithm using $\tilde{O}(nk/\epsilon)$ accesses to $\mathbf{A}$ that computes $\mathbf{Z} \in \mathbb{R}^{n \times k}$ satisfying with high probability:

$$\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F \leq (1+\epsilon)\|\mathbf{B} - \mathbf{B}_k\|_F.$$

- Can be improved using leverage-score-based Nyström approximation [Alaoui, Mahoney '15], [Musco, Musco '17].
- How does this translate to low-rank approximation of $\mathbf{A}$?

Theorem (Factor Matrix Low-Rank Approximation)

There is an algorithm using $\tilde{O}(nk/\epsilon)$ accesses to $\mathbf{A}$ that computes $\mathbf{N} \in \mathbb{R}^{n \times k}$ satisfying with high probability:

$$\|\mathbf{A} - \mathbf{N}\mathbf{N}^T\|_* \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_*.$$

- Can be improved using leverage-score-based Nyström approximation [Alaoui, Mahoney '15], [Musco, Musco '17].
- How does this translate to low-rank approximation of $\mathbf{A}$?

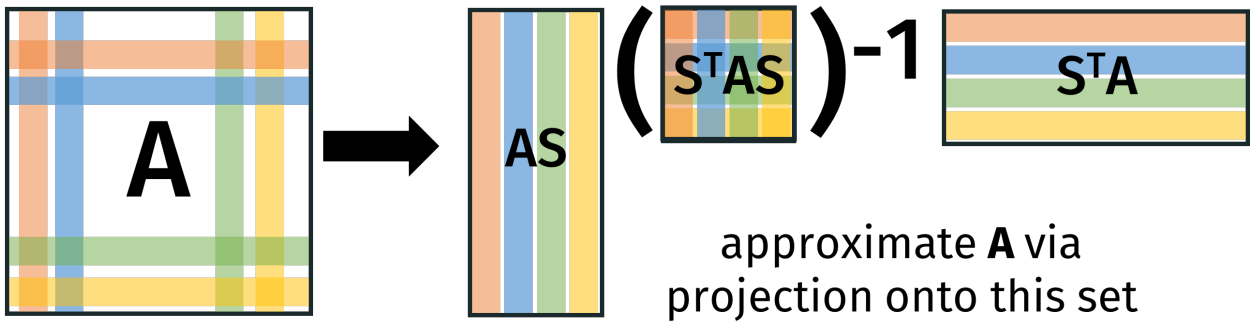
Theorem (Factor Matrix Low-Rank Approximation)

There is an algorithm using $\tilde{O}(nk/\epsilon)$ accesses to $\mathbf{A}$ that computes $\mathbf{N} \in \mathbb{R}^{n \times k}$ satisfying with high probability:

$$\|\mathbf{A} - \mathbf{N}\mathbf{N}^T\|_* \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_*.$$

- Can be improved using leverage-score-based Nyström approximation [Alaoui, Mahoney '15], [Musco, Musco '17].
- How does this translate to low-rank approximation of $\mathbf{A}$?
- Nyström algorithm computes $\mathbf{N}$ with $\mathbf{N}\mathbf{N}^T \preceq \mathbf{A} \preceq \mathbf{N}\mathbf{N}^T + \lambda\mathbf{I}$, with essentially optimal tradeoff between rank and λ.

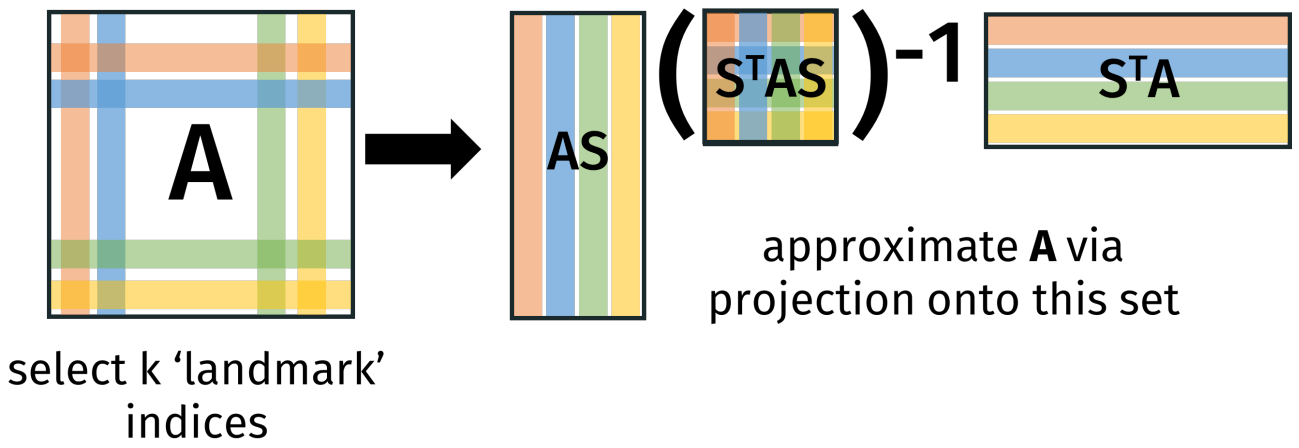
NYSTRÖM ALGORITHM



select k ‘landmark’ indices

approximate **A** via projection onto this set

NYSTRÖM ALGORITHM



- If we think of **A** as a kernel matrix, this corresponds to performing column subset selection in kernel space.

LIMITATIONS OF COLUMN SAMPLING

Stronger Guarantee: $\|\mathbf{A} - \mathbf{N}\mathbf{N}^T\|_F \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F$?

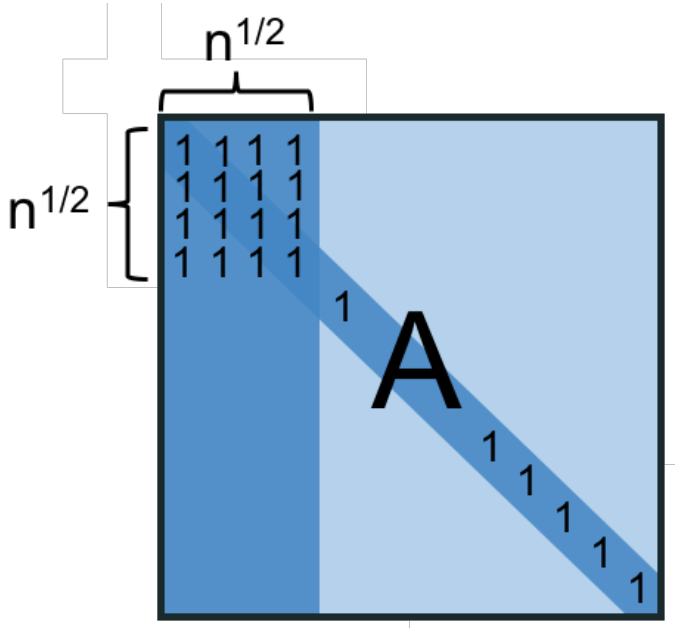
LIMITATIONS OF COLUMN SAMPLING

Stronger Guarantee: $\|\mathbf{A} - \mathbf{N}\mathbf{N}^T\|_F \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F$?

- Our Nyström algorithm accesses the diagonal of $\mathbf{A}$ along with some carefully chosen subset of its columns.

Stronger Guarantee: $\|\mathbf{A} - \mathbf{N}\mathbf{N}^T\|_F \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F$?

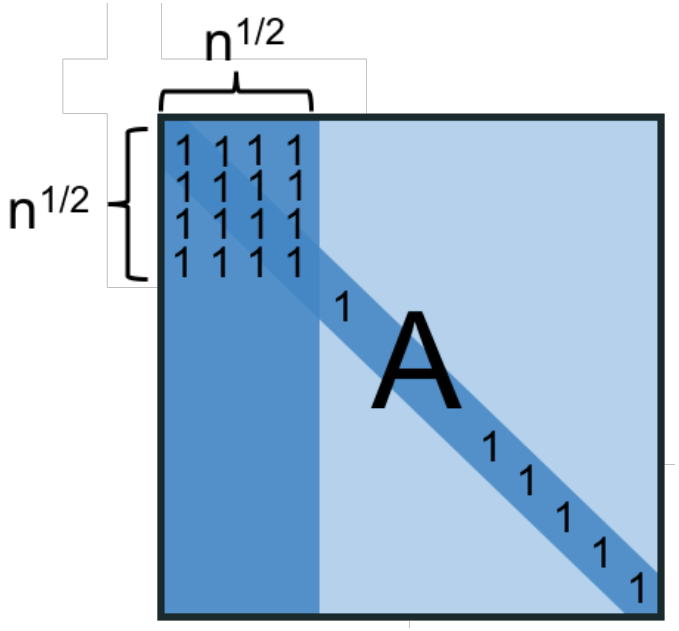
- Our Nyström algorithm accesses the diagonal of $\mathbf{A}$ along with some carefully chosen subset of its columns.



- If we take $\leq \sqrt{n}$ columns, we can miss a $\sqrt{n} \times \sqrt{n}$ block which contains a constant fraction of $\mathbf{A}$'s Frobenius norm.

Stronger Guarantee: $\|\mathbf{A} - \mathbf{N}\mathbf{N}^T\|_F \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F$?

- Our Nyström algorithm accesses the diagonal of $\mathbf{A}$ along with some carefully chosen subset of its columns.



- If we take $\leq \sqrt{n}$ columns, we can miss a $\sqrt{n} \times \sqrt{n}$ block which contains a constant fraction of $\mathbf{A}$'s Frobenius norm.
- Column sampling cannot give $o(n^{3/2})$ runtime.

Solution: Sample both rows and columns of $\mathbf{A}$, using leverage score approximations for $\mathbf{A}^{1/2}$.

Solution: Sample both rows and columns of $\mathbf{A}$, using leverage score approximations for $\mathbf{A}^{1/2}$.

- A randomly sampled $O(\sqrt{nk}) \times O(\sqrt{nk})$ submatrix contains enough information to identify a near optimal low-rank approximation of $\mathbf{A}$.

Solution: Sample both rows and columns of $\mathbf{A}$, using leverage score approximations for $\mathbf{A}^{1/2}$.

- A randomly sampled $O(\sqrt{nk}) \times O(\sqrt{nk})$ submatrix contains enough information to identify a near optimal low-rank approximation of $\mathbf{A}$.
- Sample $\mathbf{AS}$ is a projection-cost-preserving sketch for $\mathbf{A}$ [Cohen et al '15,'17]. For any rank-k projection $\mathbf{P}$

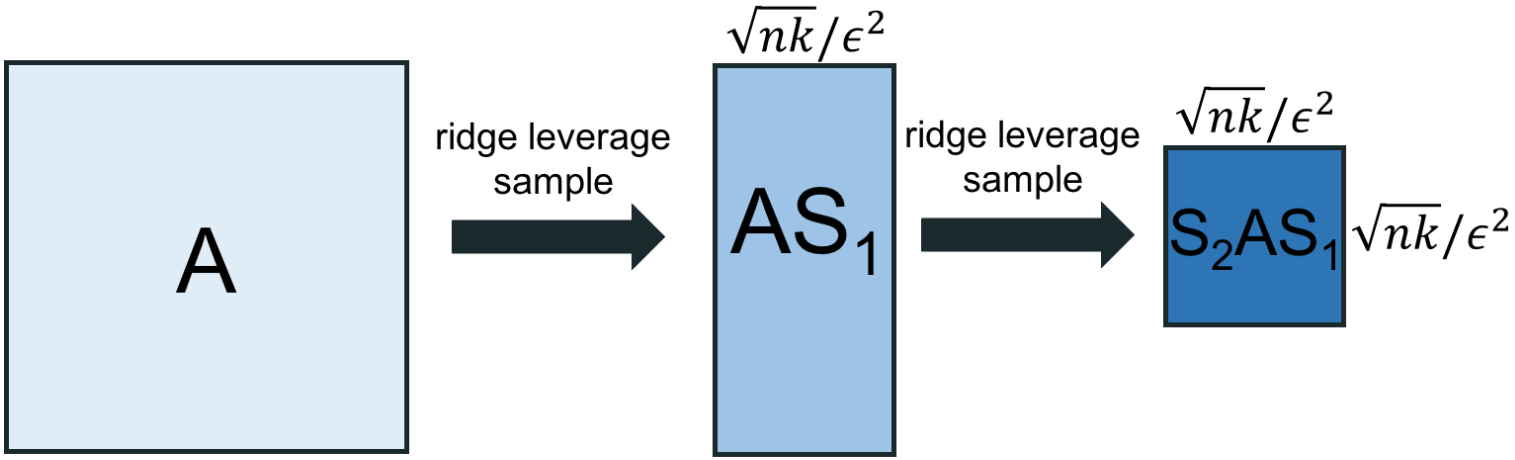
$$\|\mathbf{AS} - \mathbf{PAS}\|_F^2 = (1 \pm \epsilon)\|\mathbf{A} - \mathbf{PA}\|_F^2.$$

FINAL ALGORITHM

Recover low-rank approximation using two-sided sampling and projection-cost-preserving sketch property.

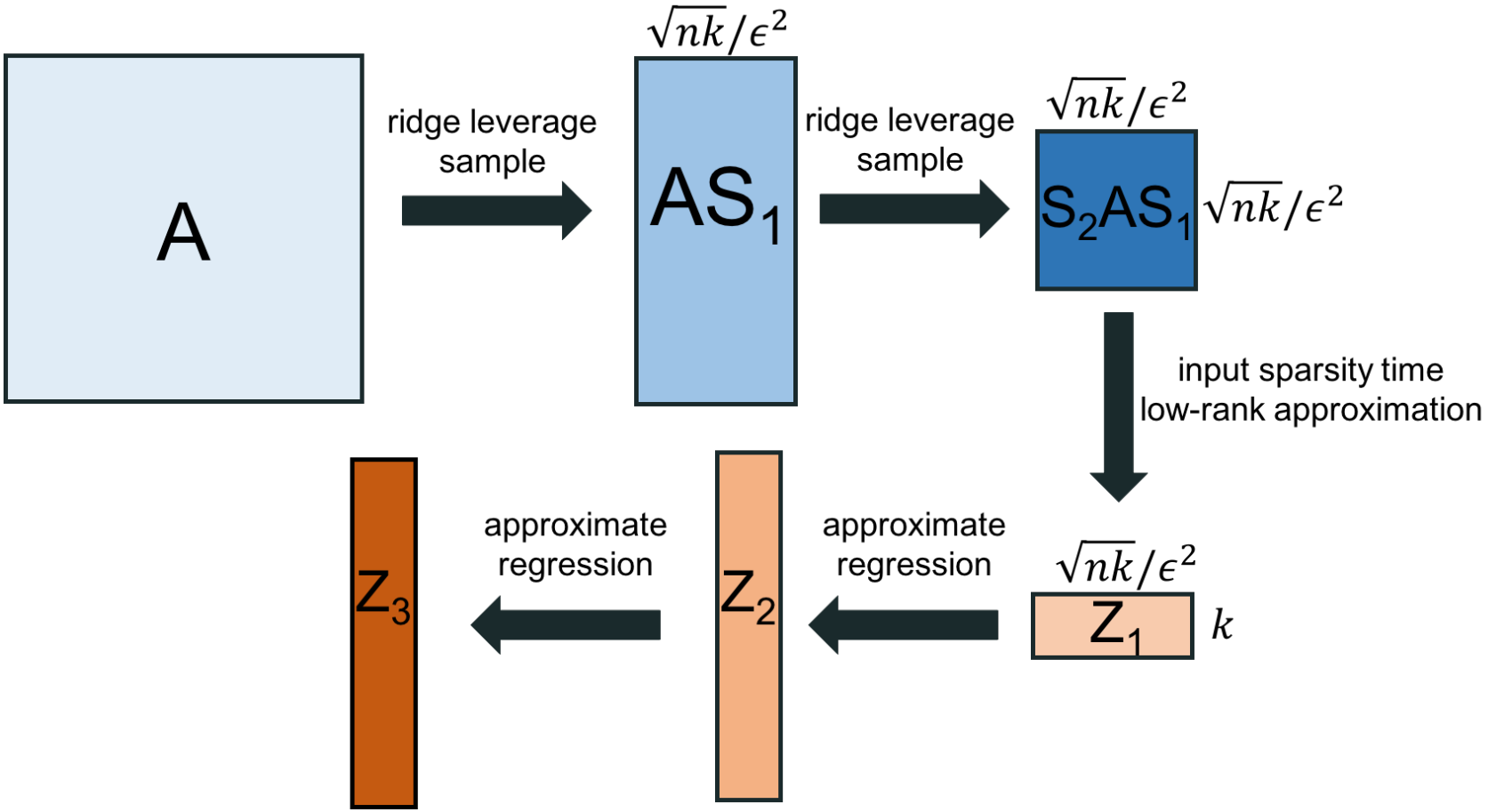
FINAL ALGORITHM

Recover low-rank approximation using two-sided sampling and projection-cost-preserving sketch property.



FINAL ALGORITHM

Recover low-rank approximation using two-sided sampling and projection-cost-preserving sketch property.



Theorem (Main Result – Musco, Woodruff '17)

There is an algorithm that, given PSD $\mathbf{A}$*, accesses* $\tilde{O}\left(\frac{nk}{\epsilon^{2.5}}\right)$ *entries of* $\mathbf{A}$ *and outputs* $\mathbf{N}, \mathbf{M} \in \mathbb{R}^{n \times k}$ *satisfying with probability* 99/100*:*

$$\|\mathbf{A} - \mathbf{N}\mathbf{M}^T\|_F \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F.$$

Theorem (Main Result – Musco, Woodruff '17)

There is an algorithm that, given PSD $\mathbf{A}$*, accesses* $\tilde{O}\left(\frac{nk}{\epsilon^{2.5}}\right)$ *entries of* $\mathbf{A}$ *and outputs* $\mathbf{N}, \mathbf{M} \in \mathbb{R}^{n \times k}$ *satisfying with probability* 99/100*:*

$$\|\mathbf{A} - \mathbf{N}\mathbf{M}^T\|_F \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F.$$

- Compare to [Clarkson Woodruff '13] which takes $O(\text{nnz}(\mathbf{A})) + n\,\text{poly}(k, 1/\epsilon)$ time.

Theorem (Main Result – Musco, Woodruff '17)

There is an algorithm that, given PSD $\mathbf{A}$*, accesses* $\tilde{O}\left(\frac{nk}{\epsilon^{2.5}}\right)$ *entries of* $\mathbf{A}$ *and outputs* $\mathbf{N}, \mathbf{M} \in \mathbb{R}^{n \times k}$ *satisfying with probability* 99/100*:*

$$\|\mathbf{A} - \mathbf{N}\mathbf{M}^T\|_F \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F.$$

- Compare to [Clarkson Woodruff '13] which takes $O(\text{nnz}(\mathbf{A})) + n\,\text{poly}(k, 1/\epsilon)$ time.

OPEN QUESTIONS

OPEN QUESTIONS

- What else can be done for PSD matrices? We give applications to ridge regression, but what other linear algebraic problems require a second look?

OPEN QUESTIONS

- What else can be done for PSD matrices? We give applications to ridge regression, but what other linear algebraic problems require a second look?
- Are there other natural classes of matrices that admit sublinear time low-rank approximation?

OPEN QUESTIONS

- What else can be done for PSD matrices? We give applications to ridge regression, but what other linear algebraic problems require a second look?
- Are there other natural classes of matrices that admit sublinear time low-rank approximation?
  - E.g. distance matrices [Bakshi, Woodruff '17]

OPEN QUESTIONS

- What else can be done for PSD matrices? We give applications to ridge regression, but what other linear algebraic problems require a second look?
- Are there other natural classes of matrices that admit sublinear time low-rank approximation?
  - E.g. distance matrices [Bakshi, Woodruff '17]
- Can we do even better for PSD matrices with additional structure? E.g. kernel matrices.

EXPLOITING ADDITIONAL STRUCTURE

When $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the kernel matrix corresponding to given $\mathbf{X} \in \mathbb{R}^{n \times d}$:

When $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the kernel matrix corresponding to given $\mathbf{X} \in \mathbb{R}^{n \times d}$:

- Our methods access $\tilde{O}(nk)$ entries of $\mathbf{A}$. Require $\tilde{\Omega}(\text{nnz}(\mathbf{X})k)$ runtime, since computing $\mathbf{A}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ requires reading $\mathbf{x}_i, \mathbf{x}_j$.

EXPLOITING ADDITIONAL STRUCTURE

When $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the kernel matrix corresponding to given $\mathbf{X} \in \mathbb{R}^{n \times d}$:

- Our methods access $\tilde{O}(nk)$ entries of $\mathbf{A}$. Require $\tilde{\Omega}(\text{nnz}(\mathbf{X})k)$ runtime, since computing $\mathbf{A}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ requires reading $\mathbf{x}_i, \mathbf{x}_j$.
- We show that $\Omega(\text{nnz}(\mathbf{X})k)$ is required to compute $\mathbf{N}$ satisfying $\|\mathbf{A} - \mathbf{N}\mathbf{N}^T\|_F \leq \Delta\|\mathbf{A} - \mathbf{A}_k\|_F^2$ for any Δ, unless the algorithm significantly advances the state of the art in fast rectangular matrix multiplication [Woodruff, Musco '17].

When $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the kernel matrix corresponding to given $\mathbf{X} \in \mathbb{R}^{n \times d}$:

- Our methods access $\tilde{O}(nk)$ entries of $\mathbf{A}$. Require $\tilde{\Omega}(\text{nnz}(\mathbf{X})k)$ runtime, since computing $\mathbf{A}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ requires reading $\mathbf{x}_i, \mathbf{x}_j$.
- We show that $\Omega(\text{nnz}(\mathbf{X})k)$ is required to compute $\mathbf{N}$ satisfying $\|\mathbf{A} - \mathbf{N}\mathbf{N}^T\|_F \leq \Delta\|\mathbf{A} - \mathbf{A}_k\|_F^2$ for any Δ, unless the algorithm significantly advances the state of the art in fast rectangular matrix multiplication [Woodruff, Musco '17].
- Obtaining $O(\text{nnz}(\mathbf{X}))$ time is open when approximation can be in terms of $\|\mathbf{A} - \mathbf{A}_k\|_*$ or for computing $\mathbf{N}\mathbf{N}^T$ satisfying

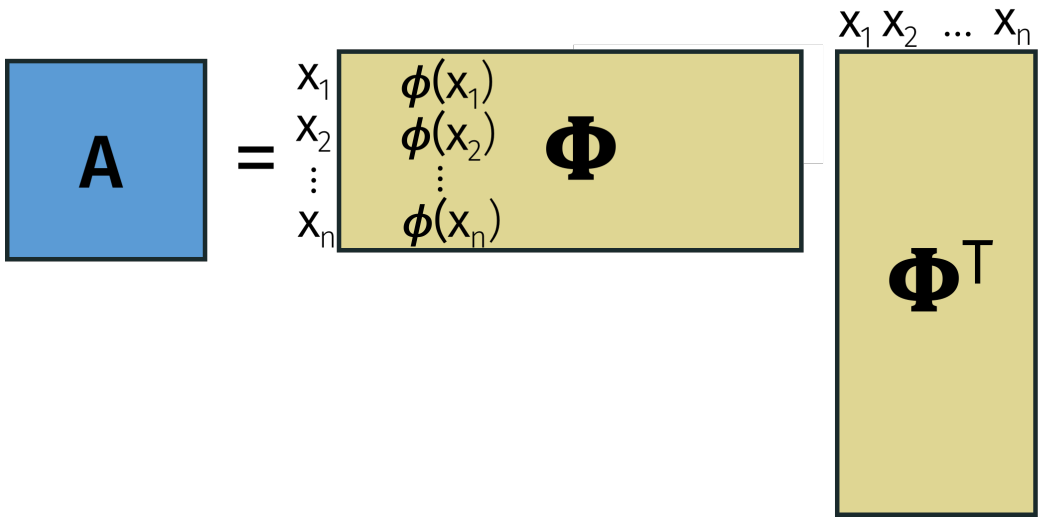
$$(1-\epsilon)\mathbf{N}\mathbf{N}^T \preceq \mathbf{A} \preceq (1+\epsilon)\mathbf{N}\mathbf{N}^T + \lambda\mathbf{I}.$$

EXPLOITING ADDITIONAL STRUCTURE

Finding input sparsity time algorithms might require developing ideas on oblivious kernel embeddings.

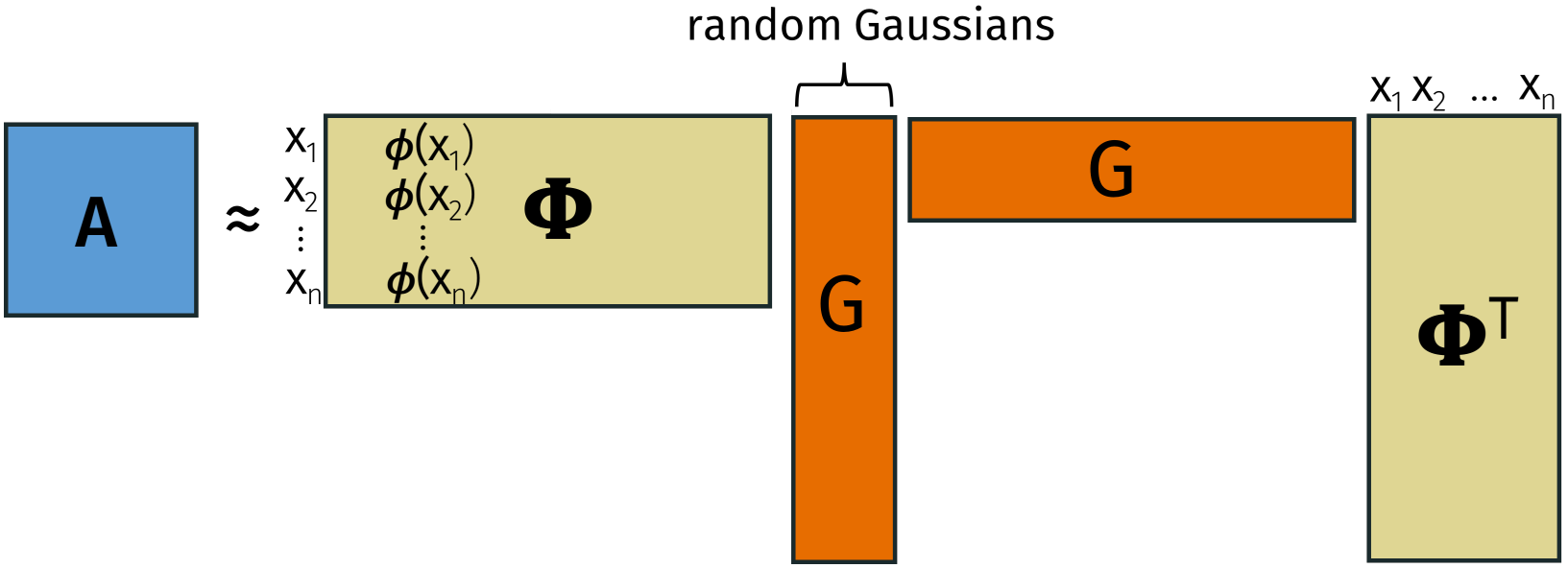
EXPLOITING ADDITIONAL STRUCTURE

Finding input sparsity time algorithms might require developing ideas on oblivious kernel embeddings.



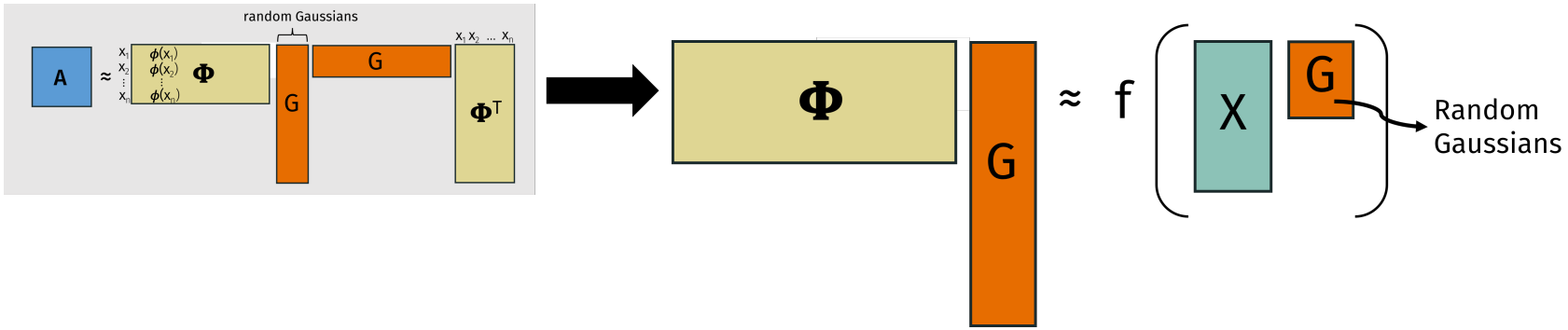
EXPLOITING ADDITIONAL STRUCTURE

Finding input sparsity time algorithms might require developing ideas on oblivious kernel embeddings.



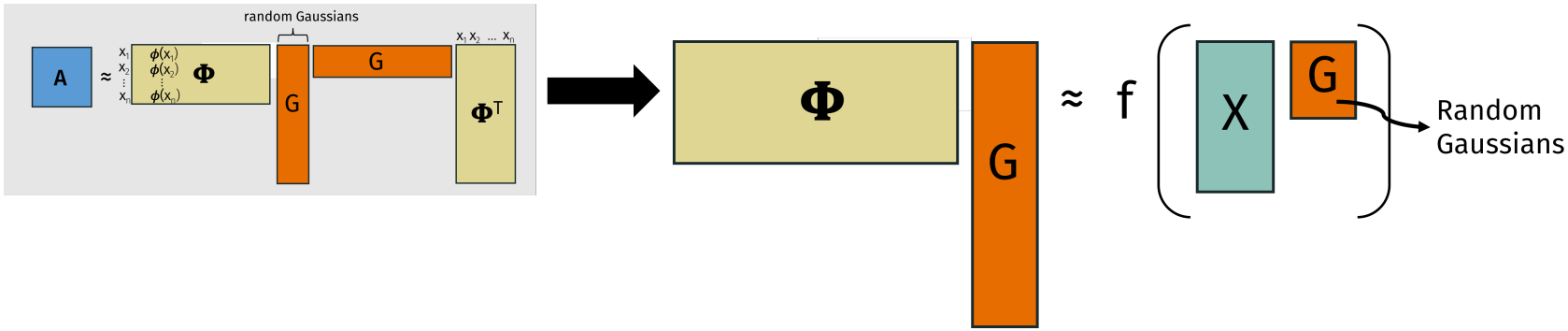
EXPLOITING ADDITIONAL STRUCTURE

Finding input sparsity time algorithms might require developing ideas on oblivious kernel embeddings.



EXPLOITING ADDITIONAL STRUCTURE

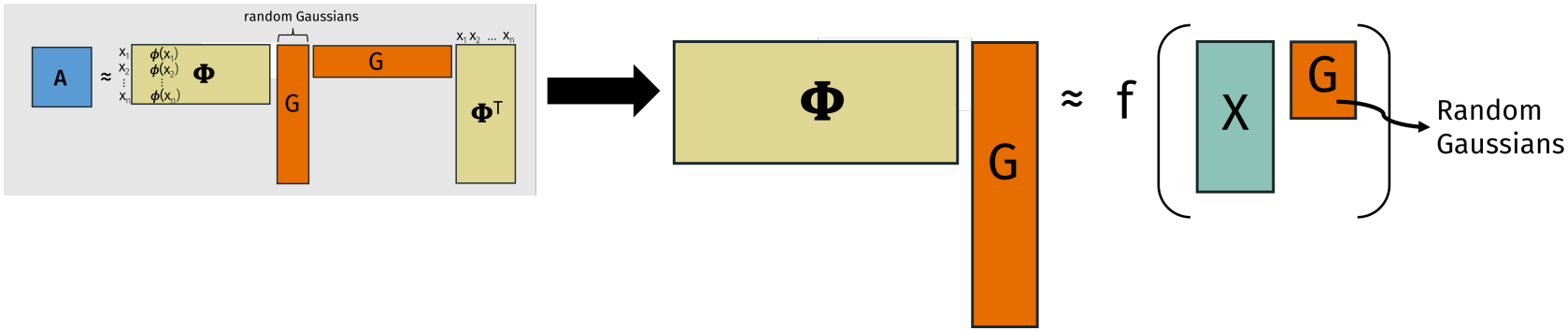
Finding input sparsity time algorithms might require developing ideas on oblivious kernel embeddings.



- Some initial progress in [Woodruff, Musco '17] combining leverage-score-based random Fourier feature sampling [Avron et al. '17] with fast approximate Gaussian matrix multiplication [Kapralov, Potluru, Woodruff '16].

EXPLOITING ADDITIONAL STRUCTURE

Finding input sparsity time algorithms might require developing ideas on oblivious kernel embeddings.



- Some initial progress in [Woodruff, Musco '17] combining leverage-score-based random Fourier feature sampling [Avron et al. '17] with fast approximate Gaussian matrix multiplication [Kapralov, Potluru, Woodruff '16].
- Further progress possible with better understanding of *fourier leverage scores*.

Thanks! Questions?

Remembering Michael Cohen (1992-2017)

Michael B. Cohen

"Michael was more than just inclusive: he actively engaged with everyone around him, on any topic, and generously shared his knowledge and observations with others. He loved teaching, and it was a joy to watch him do it."

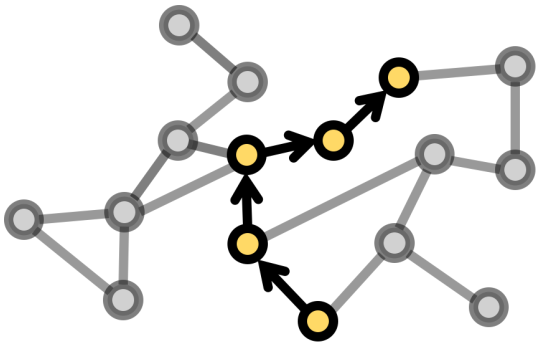
Michael's Research

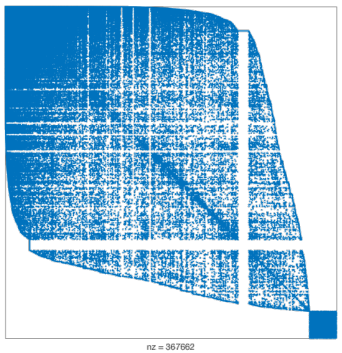
Michael's Research

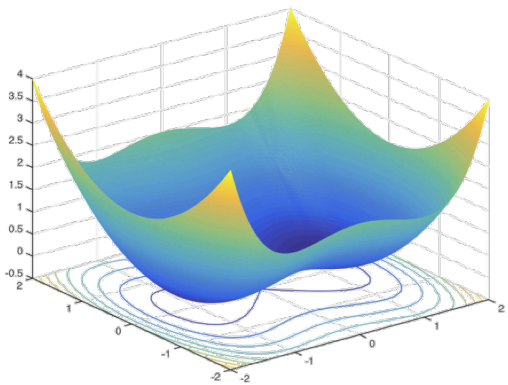
- 20+ papers, with 30+ coauthors.

Michael's Research

- 20+ papers, with 30+ coauthors.
- Faster algorithms in many fundamental areas – graph algorithms, optimization, randomized numerical linear algebra, online algorithms, computational geometry, ...







Contributions to RandNLA

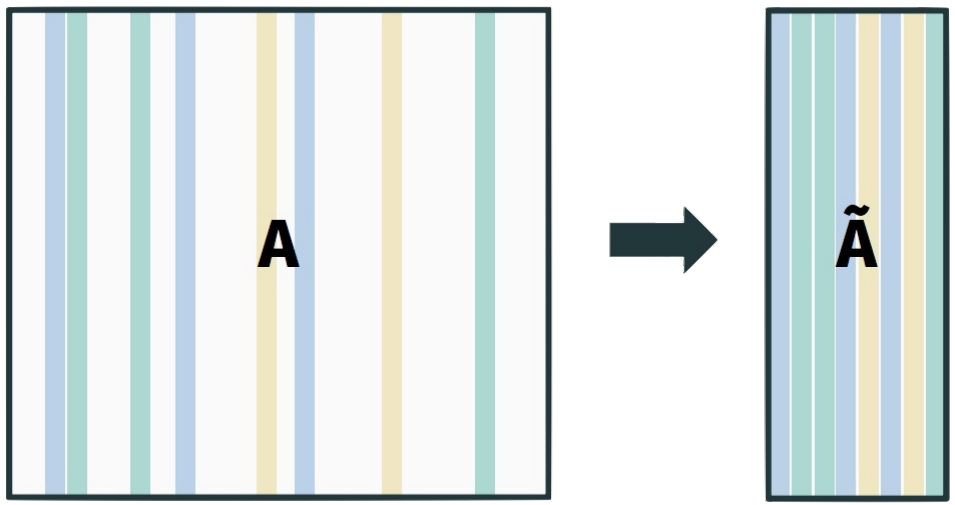
Contributions to RandNLA

Contributions to Two Main Lines of Work:

Contributions to RandNLA

Contributions to Two Main Lines of Work:

- Random row and column sampling.



Contributions to RandNLA

Contributions to Two Main Lines of Work:

- Random row and column sampling.
- *Input Sparsity Time Low-rank Approximation via Ridge Leverage Score Sampling* [Cohen, Musco, Musco, SODA 2017]
- *Online Row Sampling*
  [Cohen, Musco, Pachocki, APPROX 2016]
- *Uniform Sampling for Matrix Approximation*
  [Cohen, Lee, Musco, Musco, Peng, Sidford, ITCS 2015]
- *L_p Row Sampling by Lewis Weights*
  [Cohen, Peng, STOC 2015]

Contributions to RandNLA

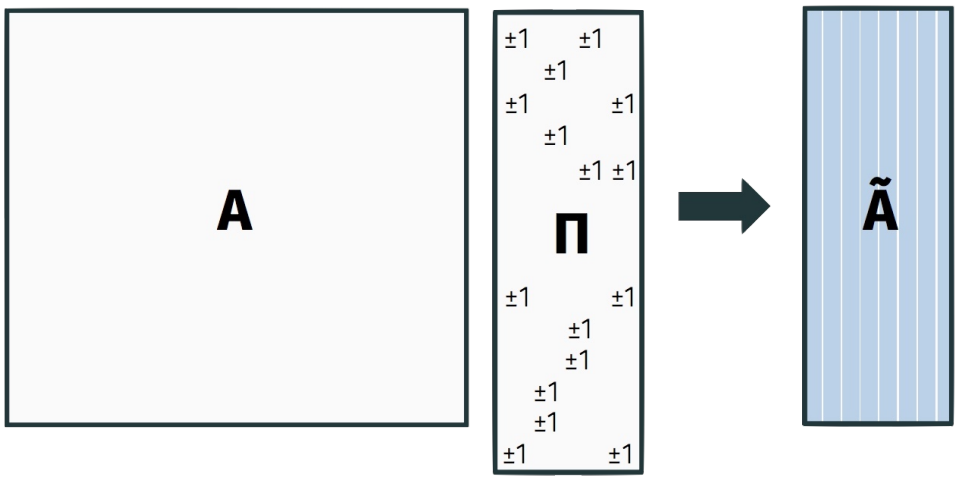
Contributions to Two Main Lines of Work:

- Random row and column sampling.
- *Input Sparsity Time Low-rank Approximation via Ridge Leverage Score Sampling* [Cohen, Musco, Musco, SODA 2017]
- *Online Row Sampling*
  [Cohen, Musco, Pachocki, APPROX 2016]
- *Uniform Sampling for Matrix Approximation*
  [Cohen, Lee, Musco, Musco, Peng, Sidford, ITCS 2015]
- *L_p Row Sampling by Lewis Weights*
  [Cohen, Peng, STOC 2015]

Contributions to RandNLA

Contributions to Two Main Lines of Work:

- Random linear sketching (Johnson-Lindenstrauss transforms).



Contributions to RandNLA

Contributions to Two Main Lines of Work:

- Random linear sketching (Johnson-Lindenstrauss transforms).
- *Simple Analyses of the Sparse Johnson-Lindenstrauss Transform*
  [Cohen, Jayram, Nelson, SOSA 2018]
- *Optimal Approximate Matrix Product in Terms of Stable Rank*
  [Cohen, Nelson, Woodruff, ICALP 2016]
- *Nearly Tight Oblivious Subspace Embeddings by Trace Inequalities*
  [Cohen, SODA 2016]
- *Dimensionality Reduction for k-Means Clustering and Low Rank Approximation*
  [Cohen, Elder, Musco, Musco, Persu, STOC 2015]

Contributions to RandNLA

Contributions to Two Main Lines of Work:

- Random linear sketching (Johnson-Lindenstrauss transforms).
- *Simple Analyses of the Sparse Johnson-Lindenstrauss Transform*
  [Cohen, Jayram, Nelson, SOSA 2018]
- *Optimal Approximate Matrix Product in Terms of Stable Rank*
  [Cohen, Nelson, Woodruff, ICALP 2016]
- *Nearly Tight Oblivious Subspace Embeddings by Trace Inequalities*
  [Cohen, SODA 2016]
- *Dimensionality Reduction for k-Means Clustering and Low Rank Approximation*
  [Cohen, Elder, Musco, Musco, Persu, STOC 2015]

Moment of Silence