SUBLINEAR TIME LOW-RANK APPROXIMATION OF POSITIVE SEMIDEFINITE MATRICES

Cameron Musco (MIT) and David P. Woodruff (CMU)

Our Contributions:

Our Contributions:

- A near optimal low-rank approximation for any positive semidefinite (PSD) matrix can be computed in sublinear time (i.e. without reading the full matrix).

Our Contributions:

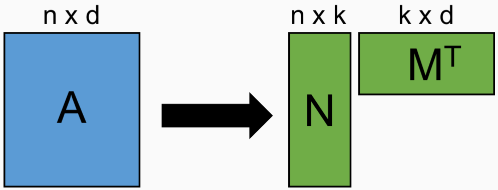
- A near optimal low-rank approximation for any positive semidefinite (PSD) matrix can be computed in sublinear time (i.e. without reading the full matrix).
- **Concrete:** Significantly improves on previous, roughly linear time approaches for general matrices, and bypasses a trivial linear time lower bound for general matrices.

Our Contributions:

- A near optimal low-rank approximation for any positive semidefinite (PSD) matrix can be computed in sublinear time (i.e. without reading the full matrix).
- **Concrete:** Significantly improves on previous, roughly linear time approaches for general matrices, and bypasses a trivial linear time lower bound for general matrices.
- **High Level:** Demonstrates that PSD structure can be exploited in a much stronger way than previously known for low-rank approximation. Opens the possibility of further advances in algorithms for PSD matrices.

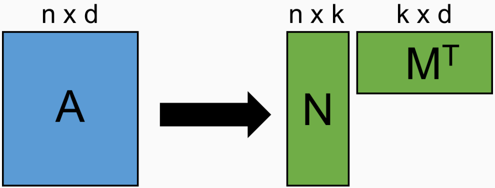
Low-rank approximation is one of the most widely used methods for general matrix and data compression.

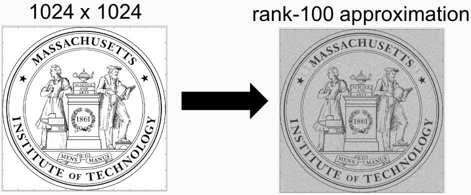
Low-rank approximation is one of the most widely used methods for general matrix and data compression.



LOW-RANK MATRIX APPROXIMATION

Low-rank approximation is one of the most widely used methods for general matrix and data compression.





- Closely related to principal component analysis, spectral embedding/clustering, and low-rank matrix completion.

- Closely related to principal component analysis, spectral embedding/clustering, and low-rank matrix completion.
- Used widely as a general pre-processing step for dimensionality reduction and data denoising.

- Closely related to principal component analysis, spectral embedding/clustering, and low-rank matrix completion.
- Used widely as a general pre-processing step for dimensionality reduction and data denoising.
- Applications to clustering, topic modeling and latent semantic analysis, recommendation systems, distribution learning, and countless other problems.

Many applications require low-rank approximation of positive semidefinite (PSD) matrices.

Many applications require low-rank approximation of positive semidefinite (PSD) matrices. $\mathbf{A} \in \mathbb{R}^{n\times n}$ with:

$$\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^n.$$

Many applications require low-rank approximation of positive semidefinite (PSD) matrices. $\mathbf{A} \in \mathbb{R}^{n \times n}$ with:

$$\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^n.$$

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

PSD LOW-RANK APPROXIMATION

Many applications require low-rank approximation of positive semidefinite (PSD) matrices. $\mathbf{A} \in \mathbb{R}^{n \times n}$ with:

$$\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^n.$$

- Includes graph Laplacians, Gram matrices and kernel matrices, covariance matrices, Hessians for convex functions.
- In the multi-dimensional scaling literature, PSD low-rank approximation is known as 'strain minimization'.

Many applications require low-rank approximation of positive semidefinite (PSD) matrices. $\mathbf{A} \in \mathbb{R}^{n \times n}$ with:

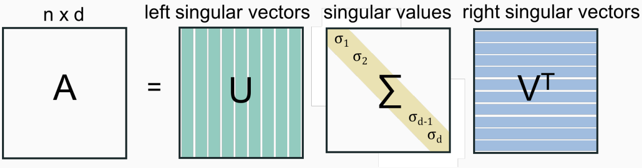
$$\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^n.$$

- Includes graph Laplacians, Gram matrices and kernel matrices, covariance matrices, Hessians for convex functions.
- In the multi-dimensional scaling literature, PSD low-rank approximation is known as ‘strain minimization’.
- Completion of (nearly) low-rank PSD matrices is applied in quantum state tomography and for global positioning using local distances (i.e. triangulation).

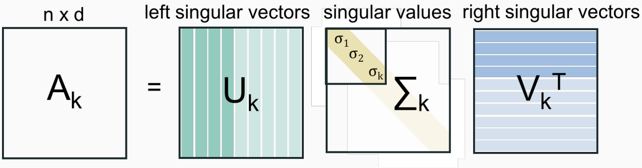
OPTIMAL LOW-RANK APPROXIMATION

An optimal low-rank approximation can be computed via the singular value decomposition (SVD).

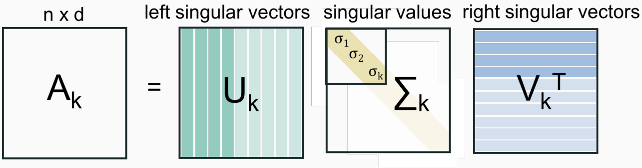
An optimal low-rank approximation can be computed via the singular value decomposition (SVD).



An optimal low-rank approximation can be computed via the singular value decomposition (SVD).

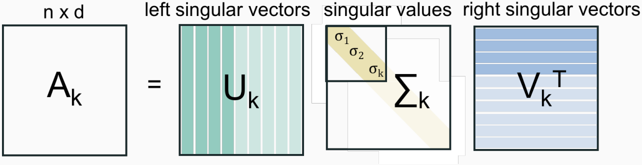


An optimal low-rank approximation can be computed via the singular value decomposition (SVD).



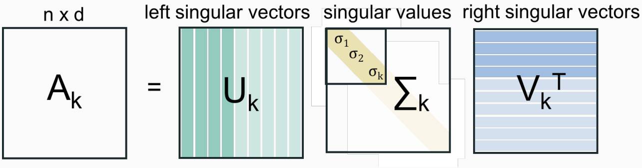
$$\mathbf{A}_k = \underset{\mathbf{B}:\text{rank}(\mathbf{B})=k}{\arg\min} \|\mathbf{A} - \mathbf{B}\|_F$$

An optimal low-rank approximation can be computed via the singular value decomposition (SVD).



$$\mathbf{A}_k = \underset{\mathbf{B}:\text{rank}(\mathbf{B})=k}{\arg\min} \|\mathbf{A}-\mathbf{B}\|_F = \sqrt{\sum_{i,j}(\mathbf{A}_{ij}-\mathbf{B}_{ij})^2}$$

An optimal low-rank approximation can be computed via the singular value decomposition (SVD).



$$\mathbf{A}_k = \operatorname*{arg\,min}_{\mathbf{B}:\mathrm{rank}(\mathbf{B})=k} \|\mathbf{A} - \mathbf{B}\|_F = \sqrt{\sum_{i,j} (\mathbf{A}_{ij} - \mathbf{B}_{ij})^2}$$

- Unfortunately, computing the SVD takes $O(nd^2)$ time.

- Traditionally, the power method of iterative Krylov subspace methods which compute just the top k singular vectors of $\mathbf{A}$ are used in lieu of a full SVD.

- Traditionally, the power method of iterative Krylov subspace methods which compute just the top k singular vectors of $\mathbf{A}$ are used in lieu of a full SVD.
- Recent work on matrix sketching gives state-of-the-art runtimes.

- Traditionally, the power method of iterative Krylov subspace methods which compute just the top k singular vectors of $\mathbf{A}$ are used in lieu of a full SVD.
- Recent work on matrix sketching gives state-of-the-art runtimes.

Theorem (Clarkson, Woodruff '13)

There is an algorithm which in $O(\mathrm{nnz}(\mathbf{A}) + n \cdot \mathrm{poly}(k, 1/\epsilon))$ *time outputs* $\mathbf{N} \in \mathbb{R}^{n \times k}, \mathbf{M} \in R^{d \times k}$ *satisfying with prob.* $99/100$*:*

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

- Traditionally, the power method of iterative Krylov subspace methods which compute just the top k singular vectors of $\mathbf{A}$ are used in lieu of a full SVD.
- Recent work on matrix sketching gives state-of-the-art runtimes.

Theorem (Clarkson, Woodruff '13)

There is an algorithm which in $O(\text{nnz}(\mathbf{A}) + n \cdot \text{poly}(k, 1/\epsilon))$ *time outputs* $\mathbf{N} \in \mathbb{R}^{n \times k}, \mathbf{M} \in R^{d \times k}$ *satisfying with prob.* 99/100*:*

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

- When $k, 1/\epsilon$ are not too large, runtime is linear in input size.

- Traditionally, the power method of iterative Krylov subspace methods which compute just the top k singular vectors of $\mathbf{A}$ are used in lieu of a full SVD.
- Recent work on matrix sketching gives state-of-the-art runtimes.

Theorem (Clarkson, Woodruff '13)

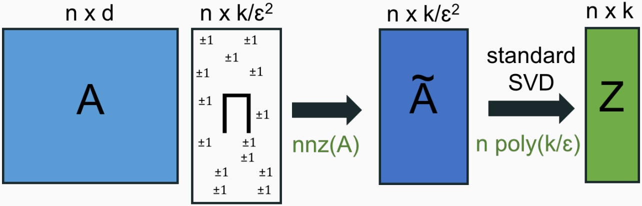
There is an algorithm which in $O(\text{nnz}(\mathbf{A}) + n \cdot \text{poly}(k, 1/\epsilon))$ *time outputs* $\mathbf{N} \in \mathbb{R}^{n \times k}, \mathbf{M} \in R^{d \times k}$ *satisfying with prob.* $99/100$*:*

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

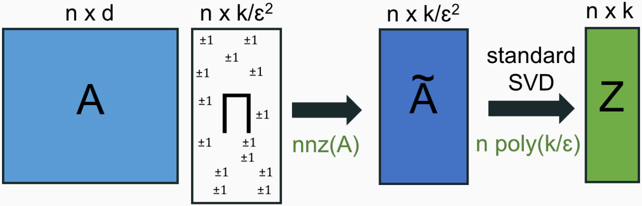
- When $k, 1/\epsilon$ are not too large, runtime is linear in input size.
- Best known runtime for both general and PSD matrices.

Clarkson and Woodruff work, along with most followup papers, is based on the ‘sketch-and-solve’ paradigm.

Clarkson and Woodruff work, along with most followup papers, is based on the 'sketch-and-solve' paradigm.



Clarkson and Woodruff work, along with most followup papers, is based on the 'sketch-and-solve' paradigm.



Similar runtimes possible via leverage score based sampling techniques.

Theorem (Main Result – Musco, Woodruff '17)

There is an algorithm running in $\tilde{O}\left(\frac{nk^2}{\epsilon^4}\right)$ *time which, given PSD* $\mathbf{A}$, *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 running in $\tilde{O}\left(\frac{nk^2}{\epsilon^4}\right)$ *time which, given PSD* $\mathbf{A}$, *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 CW'13 which takes $O(\text{nnz}(\mathbf{A})) + n\,\text{poly}(k, 1/\epsilon)$.

Theorem (Main Result – Musco, Woodruff '17)

There is an algorithm running in $\tilde{O}\left(\frac{nk^2}{\epsilon^4}\right)$ *time which, given PSD* $\mathbf{A}$*, 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 CW'13 which takes $O(\text{nnz}(\mathbf{A})) + n\,\text{poly}(k, 1/\epsilon)$.

Theorem (Main Result – Musco, Woodruff '17)

There is an algorithm running in $\tilde{O}\left(\frac{nk^2}{\epsilon^4}\right)$ *time which, given PSD* $\mathbf{A}$, *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 CW'13 which takes $O(\text{nnz}(\mathbf{A})) + n\,\text{poly}(k, 1/\epsilon)$.
- If $k, 1/\epsilon$ are not too large compared to $\text{nnz}(\mathbf{A})$, our runtime is significantly sublinear in the size of $\mathbf{A}$.

Theorem (Main Result – Musco, Woodruff '17)

There is an algorithm running in $\cancel{O\left(\frac{nk^2}{\epsilon^4}\right)}$ $O\left(n^{3/2} \cdot \text{poly}(k/\epsilon)\right)$ *time which, given PSD* $\mathbf{A}$, *outputs* $\mathbf{N}, \mathbf{M} \in \mathbb{R}^{n \times k}$ *satisfying:*

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

- Compare to CW'13 which takes $O(\text{nnz}(\mathbf{A})) + n\,\text{poly}(k, 1/\epsilon)$.
- If $k, 1/\epsilon$ are not too large compared to $\text{nnz}(\mathbf{A})$, our runtime is significantly sublinear in the size of $\mathbf{A}$.

For general matrices, $\Omega(\mathrm{nnz}(\mathbf{A}))$ time is required.

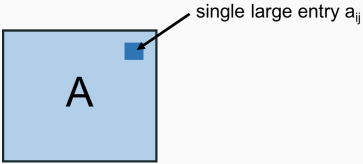
LOWER BOUND FOR GENERAL MATRICES

For general matrices, $\Omega(\text{nnz}(\mathbf{A}))$ time is required.

- Randomly place a single entry which dominates $\mathbf{A}$'s Frobenius norm.

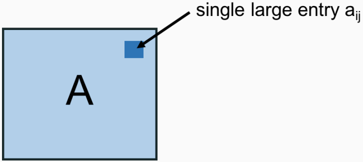
For general matrices, $\Omega(\text{nnz}(\mathbf{A}))$ time is required.

- Randomly place a single entry which dominates $\mathbf{A}$'s Frobenius norm.



For general matrices, $\Omega(\mathrm{nnz}(\mathbf{A}))$ time is required.

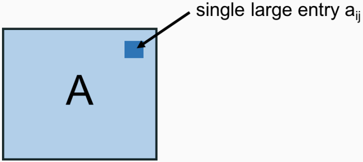
- Randomly place a single entry which dominates $\mathbf{A}$'s Frobenius norm.
- Finding it with constant probability requires reading at least a constant fraction of the non-zero entries in $\mathbf{A}$.



LOWER BOUND FOR GENERAL MATRICES

For general matrices, $\Omega(\text{nnz}(\mathbf{A}))$ time is required.

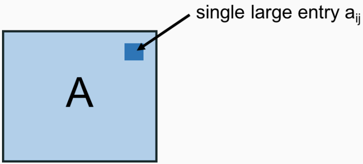
- Randomly place a single entry which dominates $\mathbf{A}$'s Frobenius norm.
- Finding it with constant probability requires reading at least a constant fraction of the non-zero entries in $\mathbf{A}$.



- Lower bound holds for any approximation factor and even rules out $o(\text{nnz}(\mathbf{A}))$ time for weaker guarantees.

For general matrices, $\Omega(\text{nnz}(\mathbf{A}))$ time is required.

- Randomly place a single entry which dominates $\mathbf{A}$'s Frobenius norm.
- Finding it with constant probability requires reading at least a constant fraction of the non-zero entries in $\mathbf{A}$.

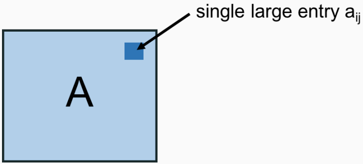


- Lower bound holds for any approximation factor and even rules out $o(\text{nnz}(\mathbf{A}))$ time for weaker guarantees.

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

For general matrices, $\Omega(\text{nnz}(\mathbf{A}))$ time is required.

- Randomly place a single entry which dominates $\mathbf{A}$'s Frobenius norm.
- Finding it with constant probability requires reading at least a constant fraction of the non-zero entries in $\mathbf{A}$.



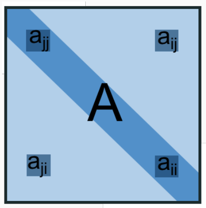
- Lower bound holds for any approximation factor and even rules out $o(\text{nnz}(\mathbf{A}))$ time for weaker guarantees.

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

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 $(\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{A}(\mathbf{e}_i - \mathbf{e}_j) < 0$, contradicting the positive semidefinite requirement.



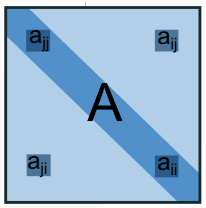
WHAT ABOUT FOR PSD MATRICES?

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 $(\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{A}(\mathbf{e}_i - \mathbf{e}_j) < 0$, contradicting the positive semidefinite requirement.

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



WHAT ABOUT FOR PSD MATRICES?

This 'heavy diagonal' fact is enough to break our lower bound for general matrices.

what about for psd matrices?

This 'heavy diagonal' fact is enough to break our lower bound for general matrices.

Question: How can we exploit additional structure arising from positive semidefiniteness to achieve sublinear runtime?

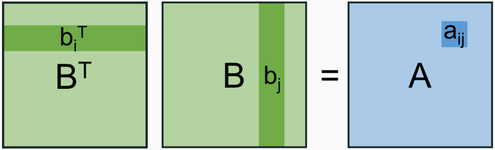
Very Simple Fact: Every PSD matrix $\mathbf{A} \in R^{n \times n}$ can be written as $\mathbf{B}^T\mathbf{B}$ for some $\mathbf{B} \in \mathbb{R}^{n \times n}$.

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 $\mathbf{V\Sigma V}^T$ be the SVD of $\mathbf{A}$, we can set $\mathbf{B} = \mathbf{\Sigma}^{1/2}\mathbf{V}^T$.

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 SVD 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 $\mathbf{a}_{ij} = \mathbf{b}_i^T\mathbf{b}_j$.



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:

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

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:

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

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 $\mathbf{a}_{ij}$.

Question: Can we compute a low-rank approximation of $\mathbf{B}$ using $o(n^2)$ column dot products? I.e. $o(n^2)$ accesses to $\mathbf{A}$?

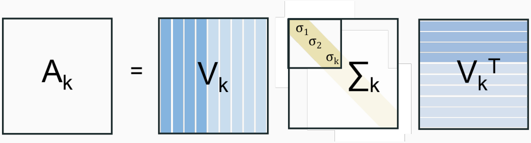
Question: Can we compute a low-rank approximation of $\mathbf{B}$ using $o(n^2)$ column dot products? I.e. $o(n^2)$ accesses to $\mathbf{A}$?

Why? $\mathbf{B}$ has the same (right) singular vectors as $\mathbf{A}$, and its singular values are closely related: $\sigma_i(\mathbf{B}) = \sqrt{\sigma_i(\mathbf{A})}$.

Question: Can we compute a low-rank approximation of $\mathbf{B}$ using $o(n^2)$ column dot products? I.e. $o(n^2)$ accesses to $\mathbf{A}$?

Why? $\mathbf{B}$ has the same (right) singular vectors as $\mathbf{A}$, and its singular values are closely related: $\sigma_i(\mathbf{B}) = \sqrt{\sigma_i(\mathbf{A})}$.

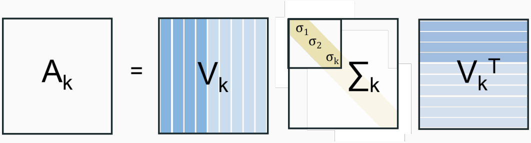
- So the top k singular vectors are the same for the two matrices. An optimal low-rank approximation for $\mathbf{B}$ thus gives an optimal low-rank approximation for $\mathbf{A}$.



Question: Can we compute a low-rank approximation of $\mathbf{B}$ using $o(n^2)$ column dot products? I.e. $o(n^2)$ accesses to $\mathbf{A}$?

Why? $\mathbf{B}$ has the same (right) singular vectors as $\mathbf{A}$, and its singular values are closely related: $\sigma_i(\mathbf{B}) = \sqrt{\sigma_i(\mathbf{A})}$.

- So the top k singular vectors are the same for the two matrices. An optimal low-rank approximation for $\mathbf{B}$ thus gives an optimal low-rank approximation for $\mathbf{A}$.
- Things will be messier once we introduce approximation.



FACTOR MATRIX LOW-RANK APPROXIMATION

FACTOR MATRIX LOW-RANK APPROXIMATION

More concretely, we want to compute some orthogonal span $\mathbf{Z} \in \mathbb{R}^{n \times k}$ (i.e. with $\mathbf{Z}^T\mathbf{Z} = \mathbf{I}$) satisfying:

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

using a sublinear number of column dot products (i.e. accesses to $\mathbf{A} = \mathbf{B}^T\mathbf{B}$.)

More concretely, we want to compute some orthogonal span $\mathbf{Z} \in \mathbb{R}^{n \times k}$ (i.e. with $\mathbf{Z}^T\mathbf{Z} = \mathbf{I}$) satisfying:

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

using a sublinear number of column dot products (i.e. accesses to $\mathbf{A} = \mathbf{B}^T\mathbf{B}$.)

Aside: Computing a low-rank approximation of $\mathbf{B}$ is interesting in its own right. When $\mathbf{A}$ is a kernel matrix, this is essentially the problem of kernel PCA.

More concretely, we want to compute some orthogonal span $\mathbf{Z} \in \mathbb{R}^{n \times k}$ (i.e. with $\mathbf{Z}^T\mathbf{Z} = \mathbf{I}$) satisfying:

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

using a sublinear number of column dot products (i.e. accesses to $\mathbf{A} = \mathbf{B}^T\mathbf{B}$.)

Aside: Computing a low-rank approximation of $\mathbf{B}$ is interesting in its own right. When $\mathbf{A}$ is a kernel matrix, this is essentially the problem of kernel PCA.

- Can also be used to accelerate kernel ridge regression, k-means clustering, and CCA (Musco, Musco '17).

Theorem (Deshpande, Vempala '06)

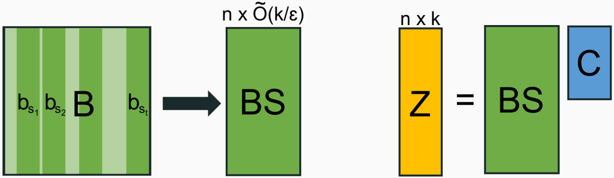
For any $\mathbf{B} \in \mathbb{R}^{n\times n}$, *there exists a subset of* $4k/\epsilon + 2k\log(k+1)$ *columns whose span contains* $\mathbf{Z} \in \mathbb{R}^{n\times k}$ *satisfying:*

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

Theorem (Deshpande, Vempala '06)

For any $\mathbf{B} \in \mathbb{R}^{n \times n}$, *there exists a subset of* $4k/\epsilon + 2k\log(k+1)$ *columns whose span contains* $\mathbf{Z} \in \mathbb{R}^{n \times k}$ *satisfying:*

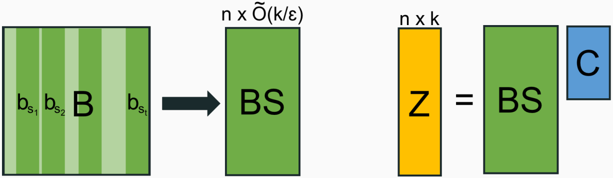
$$\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F \leq (1+\epsilon)\|\mathbf{B} - \mathbf{B}_k\|_F$$



Theorem (Deshpande, Vempala '06)

For any $\mathbf{B} \in \mathbb{R}^{n \times n}$*, there exists a subset of* $4k/\epsilon + 2k\log(k+1)$ *columns whose span contains* $\mathbf{Z} \in \mathbb{R}^{n \times k}$ *satisfying:*

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



Observation: Given column subset, $\mathbf{C}$ can be computed using just $\tilde{O}(n \cdot k/\epsilon)$ column dot products (i.e. must compute $(\mathbf{BS})^T\mathbf{B}$).

Additionally, a $\tilde{O}(k^2/\epsilon)$ sized column subset can be found using an intuitive adaptive sampling strategy.

IDENTIFYING THE COLUMN SUBSET

Additionally, a $\tilde{O}(k^2/\epsilon)$ sized column subset can be found using an intuitive adaptive sampling strategy.

- A number of alternatives using leverage scores, DPPs, or deterministic potential function methods exist, but adaptive sampling is the simplest.

Additionally, a $\tilde{O}(k^2/\epsilon)$ sized column subset can be found using an intuitive adaptive sampling strategy.

- A number of alternatives using leverage scores, DPPs, or deterministic potential function methods exist, but adaptive sampling is the simplest.

Adaptive Sampling Column Subset Selection

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 $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.

Adaptive Sampling Column Subset Selection

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

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.

Adaptive Sampling Column Subset Selection

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

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.

$$\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2} = \frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^n \|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$$

Adaptive Sampling Column Subset Selection

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

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.

$$\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2} = \frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^n \|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$$

Adaptive Sampling Column Subset Selection

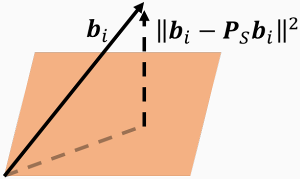
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.

$$\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2} = \frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\sum_{i=1}^n \|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}$$



Adaptive Sampling Column Subset Selection

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

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.

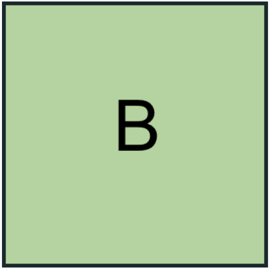
Adaptive Sampling Column Subset Selection

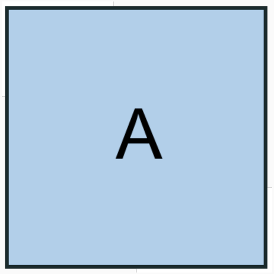
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





IDENTIFYING THE COLUMN SUBSET

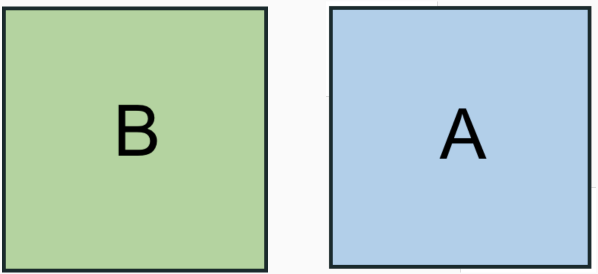
Adaptive Sampling Column Subset Selection

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

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

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

  Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.



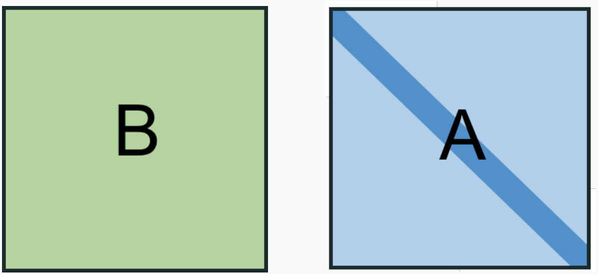
Adaptive Sampling Column Subset Selection

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

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

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

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



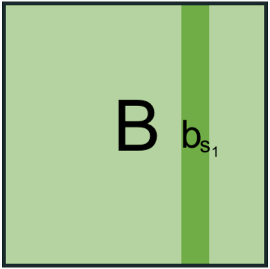
Adaptive Sampling Column Subset Selection

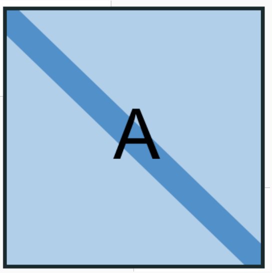
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

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

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





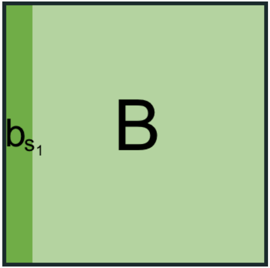
Adaptive Sampling Column Subset Selection

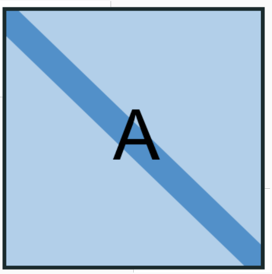
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

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

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





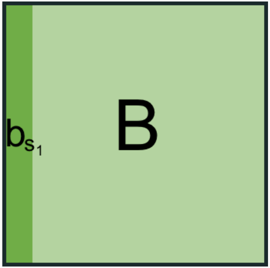
Adaptive Sampling Column Subset Selection

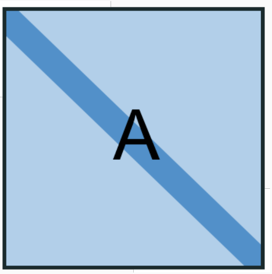
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





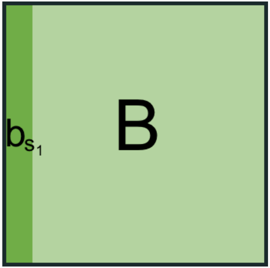
Adaptive Sampling Column Subset Selection

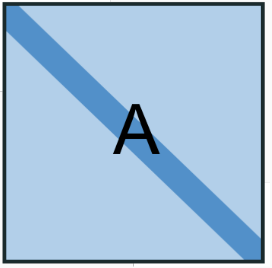
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

Let $\mathbf{P}_{\mathcal{S}} = \frac{\mathbf{b}_{s_1}\mathbf{b}_{s_1}^T}{\|\mathbf{b}_{s_1}\|^2}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





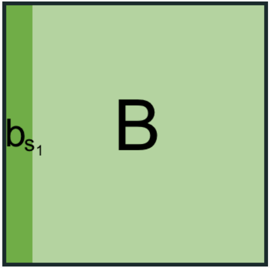
Adaptive Sampling Column Subset Selection

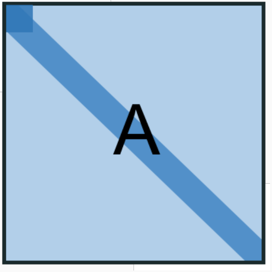
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

Let $\mathbf{P}_{\mathcal{S}} = \frac{\mathbf{b}_{s_1}\mathbf{b}_{s_1}^T}{\|\mathbf{b}_{s_1}\|^2}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





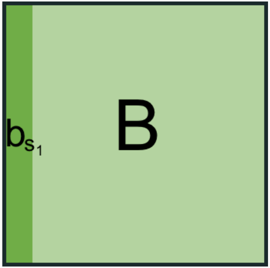
Adaptive Sampling Column Subset Selection

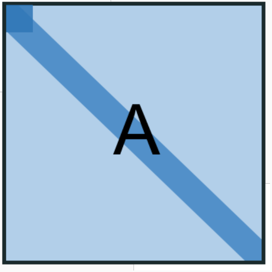
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

Let $\mathbf{P}_{\mathcal{S}} = \frac{\mathbf{b}_{s_1}\mathbf{b}_{s_1}^T}{\|\mathbf{b}_{s_1}\|^2}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





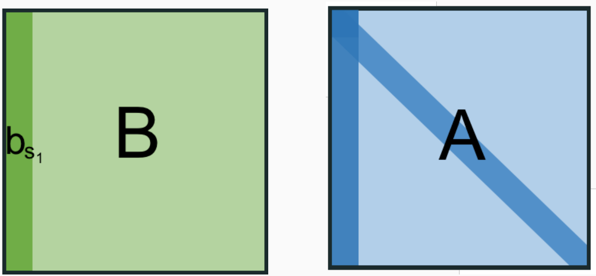
Adaptive Sampling Column Subset Selection

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

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

Let $\mathbf{P}_{\mathcal{S}} = \frac{\mathbf{b}_{s_1}\mathbf{b}_{s_1}^T}{\|\mathbf{b}_{s_1}\|^2}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.



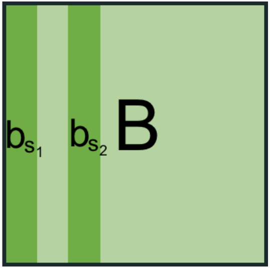
Adaptive Sampling Column Subset Selection

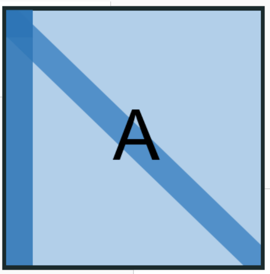
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

Let $\mathbf{P}_{\mathcal{S}} = \frac{\mathbf{b}_{s_1}\mathbf{b}_{s_1}^T}{\|\mathbf{b}_{s_1}\|^2}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





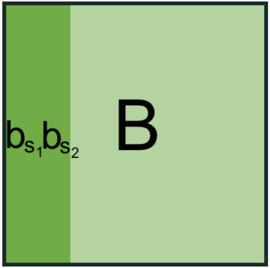
Adaptive Sampling Column Subset Selection

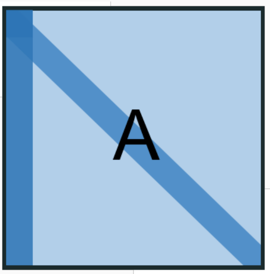
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

Let $\mathbf{P}_{\mathcal{S}} = \frac{\mathbf{b}_{s_1}\mathbf{b}_{s_1}^T}{\|\mathbf{b}_{s_1}\|^2}$ be the projection onto the columns in $\mathcal{S}$.

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





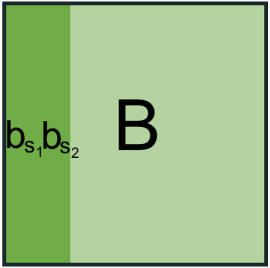
Adaptive Sampling Column Subset Selection

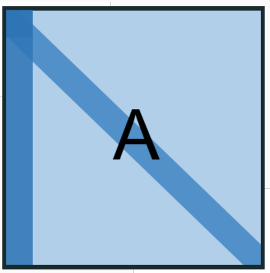
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





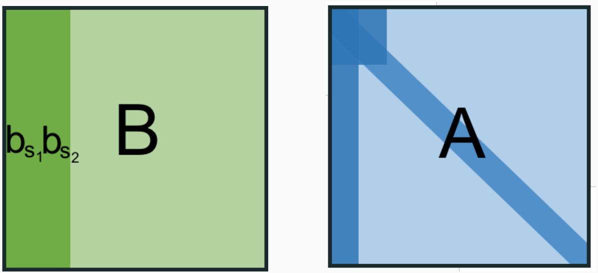
Adaptive Sampling Column Subset Selection

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

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.



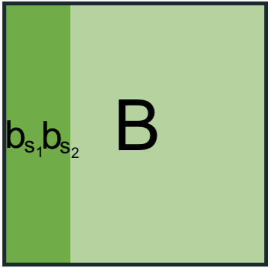
Adaptive Sampling Column Subset Selection

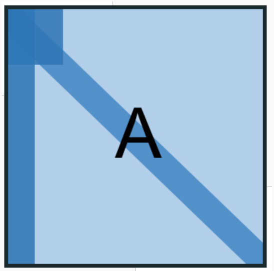
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





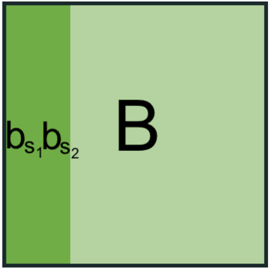
Adaptive Sampling Column Subset Selection

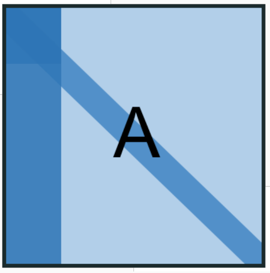
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





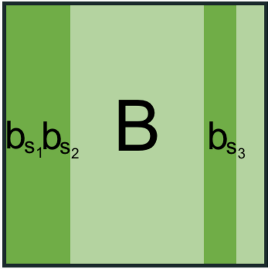
Adaptive Sampling Column Subset Selection

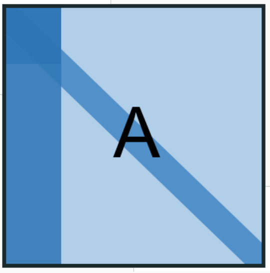
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





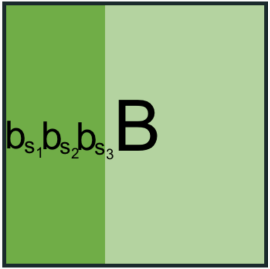
Adaptive Sampling Column Subset Selection

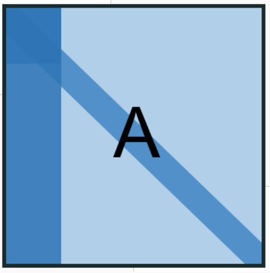
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





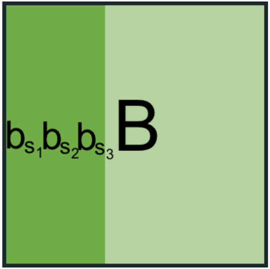
Adaptive Sampling Column Subset Selection

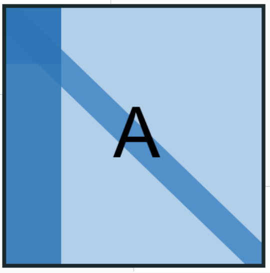
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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 $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





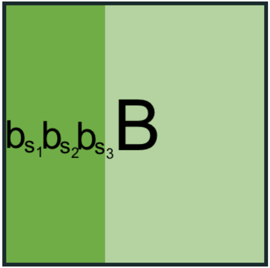
Adaptive Sampling Column Subset Selection

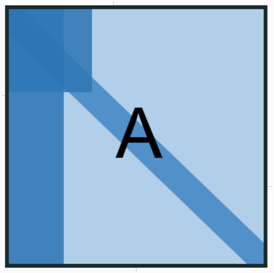
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





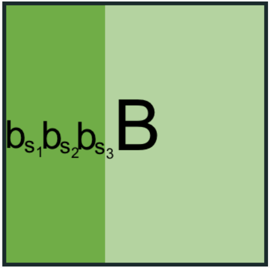
Adaptive Sampling Column Subset Selection

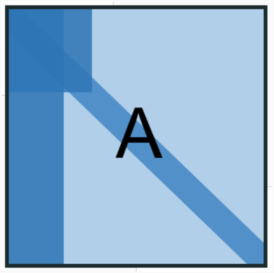
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





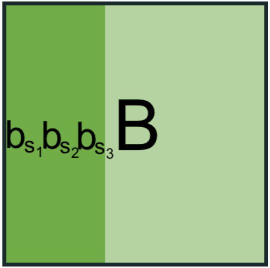
Adaptive Sampling Column Subset Selection

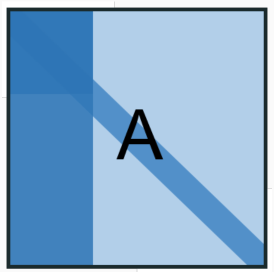
Initially, start with an empty column subset $\mathcal{S} := \{\}$.

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

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

Add $\mathbf{b}_i$ to $\mathcal{S}$ with probability $\frac{\|\mathbf{b}_i - \mathbf{P}_{\mathcal{S}}\mathbf{b}_i\|^2}{\|\mathbf{B} - \mathbf{P}_{\mathcal{S}}\mathbf{B}\|_F^2}$.





Theorem

There is an algorithm using $\tilde{O}(n \cdot k^2/\epsilon)$ *column dot products (i.e. accesses to* $\mathbf{A} = \mathbf{B}^T\mathbf{B}$*) which computes sampling matrix* $\mathbf{S} \in \mathbb{R}^{n \times \tilde{O}(k^2/\epsilon)}$ *and* $\mathbf{C} \in \mathbb{R}^{\tilde{O}(k^2/\epsilon) \times k}$ *such that* $\mathbf{Z} = \mathbf{BSC}$ *satisfies with probability* $99/100$*:*

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

Theorem

There is an algorithm using $\tilde{O}(n \cdot k^2/\epsilon)$ column dot products (i.e. accesses to $\mathbf{A} = \mathbf{B}^T\mathbf{B}$) which computes sampling matrix $\mathbf{S} \in \mathbb{R}^{n \times \tilde{O}(k^2/\epsilon)}$ and $\mathbf{C} \in \mathbb{R}^{\tilde{O}(k^2/\epsilon) \times k}$ such that $\mathbf{Z} = \mathbf{BSC}$ satisfies with probability $99/100$:

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

- I.e., a near optimal low-rank approximation can be found using much less information about **B**'s column span than a full SVD.

Theorem

There is an algorithm using $\tilde{O}(n \cdot k^2/\epsilon)$ column dot products (i.e. accesses to $\mathbf{A} = \mathbf{B}^T\mathbf{B}$) which computes sampling matrix $\mathbf{S} \in \mathbb{R}^{n \times \tilde{O}(k^2/\epsilon)}$ and $\mathbf{C} \in \mathbb{R}^{\tilde{O}(k^2/\epsilon) \times k}$ such that $\mathbf{Z} = \mathbf{BSC}$ satisfies with probability $99/100$:

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

- I.e., a near optimal low-rank approximation can be found using much less information about **B**'s column span than a full SVD.
- But what can we do with this result?

As mentioned, if $\mathbf{Z}$ gave an optimal low-rank approximation $\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F = \|\mathbf{B} - \mathbf{B}_k\|_F$ then it would immediately give an optimal approximation for $\mathbf{A} = \mathbf{B}^T\mathbf{B}$.

As mentioned, if $\mathbf{Z}$ gave an optimal low-rank approximation $\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F = \|\mathbf{B} - \mathbf{B}_k\|_F$ then it would immediately give an optimal approximation for $\mathbf{A} = \mathbf{B}^T\mathbf{B}$.

$$\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}$$

As mentioned, if $\mathbf{Z}$ gave an optimal low-rank approximation $\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F = \|\mathbf{B} - \mathbf{B}_k\|_F$ then it would immediately give an optimal approximation for $\mathbf{A} = \mathbf{B}^T\mathbf{B}$.

$$\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B} = (\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T)(\mathbf{Z}\mathbf{Z}^T\mathbf{B})$$

As mentioned, if $\mathbf{Z}$ gave an optimal low-rank approximation $\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F = \|\mathbf{B} - \mathbf{B}_k\|_F$ then it would immediately give an optimal approximation for $\mathbf{A} = \mathbf{B}^T\mathbf{B}$.

$$\begin{aligned}\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B} &= (\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T)(\mathbf{Z}\mathbf{Z}^T\mathbf{B}) \\ &= \mathbf{B}_k^T\mathbf{B}_k\end{aligned}$$

As mentioned, if $\mathbf{Z}$ gave an optimal low-rank approximation $\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F = \|\mathbf{B} - \mathbf{B}_k\|_F$ then it would immediately give an optimal approximation for $\mathbf{A} = \mathbf{B}^T\mathbf{B}$.

$$\begin{aligned}
\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B} &= (\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T)(\mathbf{Z}\mathbf{Z}^T\mathbf{B}) \\
&= \mathbf{B}_k^T\mathbf{B}_k \\
&= \mathbf{V}\mathbf{\Sigma}_k^{1/2}\mathbf{\Sigma}_k^{1/2}\mathbf{V}^T
\end{aligned}$$

As mentioned, if $\mathbf{Z}$ gave an optimal low-rank approximation $\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F = \|\mathbf{B} - \mathbf{B}_k\|_F$ then it would immediately give an optimal approximation for $\mathbf{A} = \mathbf{B}^T\mathbf{B}$.

$$\begin{aligned}
\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B} &= (\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T)(\mathbf{Z}\mathbf{Z}^T\mathbf{B}) \\
&= \mathbf{B}_k^T\mathbf{B}_k \\
&= \mathbf{V}\boldsymbol{\Sigma}_k^{1/2}\boldsymbol{\Sigma}_k^{1/2}\mathbf{V}^T \\
&= \mathbf{A}_k.
\end{aligned}$$

BOOSTING TO A PSD MATRIX APPROXIMATION

As mentioned, if $\mathbf{Z}$ gave an optimal low-rank approximation $\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F = \|\mathbf{B} - \mathbf{B}_k\|_F$ then it would immediately give an optimal approximation for $\mathbf{A} = \mathbf{B}^T\mathbf{B}$.

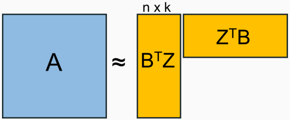
$$\begin{aligned}
\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B} &= (\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T)(\mathbf{Z}\mathbf{Z}^T\mathbf{B}) \\
&= \mathbf{B}_k^T\mathbf{B}_k \\
&= \mathbf{V}\boldsymbol{\Sigma}_k^{1/2}\boldsymbol{\Sigma}_k^{1/2}\mathbf{V}^T \\
&= \mathbf{A}_k.
\end{aligned}$$

- Gives $n \cdot \text{poly}(k)$ time low-rank PSD matrix completion (i.e. when $\|\mathbf{A} - \mathbf{A}_k\|_F = 0$).

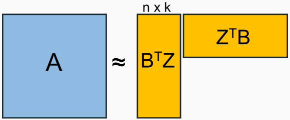
Given $\mathbf{Z}\mathbf{Z}^T\mathbf{B}$ we approximate $\mathbf{A}$ with $\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}$.

BOOSTING TO A PSD MATRIX APPROXIMATION

Given $\mathbf{Z}\mathbf{Z}^T\mathbf{B}$ we approximate $\mathbf{A}$ with $\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}$.

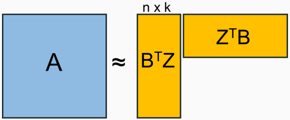


Given $\mathbf{Z}\mathbf{Z}^T\mathbf{B}$ we approximate $\mathbf{A}$ with $\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}$.



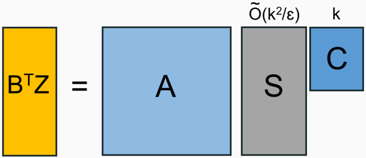
$\mathbf{B}^T\mathbf{Z}$ can be computed efficiently without explicitly forming $\mathbf{B}$.

Given $\mathbf{Z}\mathbf{Z}^T\mathbf{B}$ we approximate $\mathbf{A}$ with $\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}$.

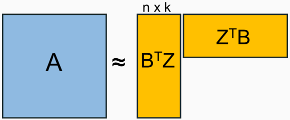


$\mathbf{B}^T\mathbf{Z}$ can be computed efficiently without explicitly forming $\mathbf{B}$.

- $\mathbf{B}^T\mathbf{Z} = \mathbf{B}^T(\mathbf{BSC}) = \mathbf{ASC}$.

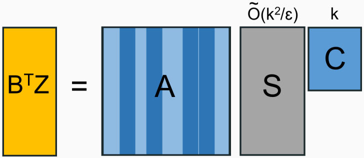


Given $\mathbf{Z}\mathbf{Z}^T\mathbf{B}$ we approximate $\mathbf{A}$ with $\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}$.

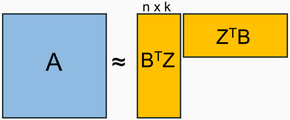


$\mathbf{B}^T\mathbf{Z}$ can be computed efficiently without explicitly forming $\mathbf{B}$.

- $\mathbf{B}^T\mathbf{Z} = \mathbf{B}^T(\mathbf{BSC}) = \mathbf{ASC}$.

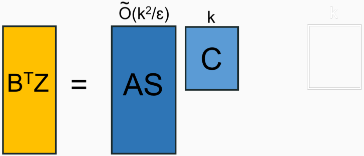


Given $\mathbf{Z}\mathbf{Z}^T\mathbf{B}$ we approximate $\mathbf{A}$ with $\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}$.

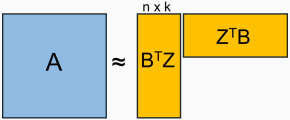


$\mathbf{B}^T\mathbf{Z}$ can be computed efficiently without explicitly forming $\mathbf{B}$.

- $\mathbf{B}^T\mathbf{Z} = \mathbf{B}^T(\mathbf{BSC}) = \mathbf{ASC}$.

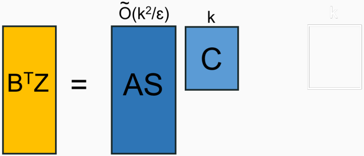


Given $\mathbf{ZZ}^T\mathbf{B}$ we approximate $\mathbf{A}$ with $\mathbf{B}^T\mathbf{ZZ}^T\mathbf{B}$.



$\mathbf{B}^T\mathbf{Z}$ can be computed efficiently without explicitly forming $\mathbf{B}$.

- $\mathbf{B}^T\mathbf{Z} = \mathbf{B}^T(\mathbf{BSC}) = \mathbf{ASC}$.



- $n \cdot \text{poly}(k, 1/\epsilon)$ accesses to $\mathbf{A}$ and run time.

What about when $\mathbf{Z}$ just gives a near-optimal approximation?

What about when $\mathbf{Z}$ just gives a near-optimal approximation?

Lemma

If $\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \left(1 + \frac{\epsilon^{3/2}}{\sqrt{n}}\right)\|\mathbf{B} - \mathbf{B}_k\|_F^2$ *where* $\mathbf{Z} = \mathbf{BSC}$, *then for* $\mathbf{A} = \mathbf{B}^T\mathbf{B}$*:*

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

What about when $\mathbf{Z}$ just gives a near-optimal approximation?

Lemma

If $\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \left(1 + \frac{\epsilon^{3/2}}{\sqrt{n}}\right) \|\mathbf{B} - \mathbf{B}_k\|_F^2$ *where* $\mathbf{Z} = \mathbf{BSC}$, *then for* $\mathbf{A} = \mathbf{B}^T\mathbf{B}$*:*

$$\|\mathbf{A} - (\mathbf{ASC})(\mathbf{ASC})^T\|_F^2 \leq (1 + \epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

What about when $\mathbf{Z}$ just gives a near-optimal approximation?

Lemma

If $\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \left(1 + \frac{\epsilon^{3/2}}{\sqrt{n}}\right)\|\mathbf{B} - \mathbf{B}_k\|_F^2$ *where* $\mathbf{Z} = \mathbf{BSC}$*, then for* $\mathbf{A} = \mathbf{B}^T\mathbf{B}$*:*

$$\|\mathbf{A} - (\mathbf{ASC})(\mathbf{ASC})^T\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

What about when $\mathbf{Z}$ just gives a near-optimal approximation?

Lemma

If $\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \left(1 + \frac{\epsilon^{3/2}}{\sqrt{n}}\right)\|\mathbf{B} - \mathbf{B}_k\|_F^2$ *where* $\mathbf{Z} = \mathbf{BSC}$, *then for* $\mathbf{A} = \mathbf{B}^T\mathbf{B}$:

$$\|\mathbf{A} - (\mathbf{ASC})(\mathbf{ASC})^T\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

This will give an low-rank approximation algorithm which accesses just $\tilde{O}\left(\frac{nk^2}{\epsilon^{3/2}/\sqrt{n}}\right) = n^{3/2} \cdot \text{poly}(k, 1/\epsilon)$ entries of $\mathbf{A}$.

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 = \|\mathbf{B}^T(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{B}\|_F^2$$

$$
\begin{aligned}
\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 &= \|\mathbf{B}^T(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{B}\|_F^2 \\
&= \sum_{i=1}^{n-k} \sigma_i^2(\mathbf{B}^T(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{B})
\end{aligned}
$$

$$
\begin{aligned}
\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 &= \|\mathbf{B}^T(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{B}\|_F^2 \\
&= \sum_{i=1}^{n-k} \sigma_i^2(\mathbf{B}^T(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{B}) \\
&= \sum_{i=1}^{n-k} \sigma_i^4((\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{B}).
\end{aligned}
$$

$$\begin{aligned}\|\mathbf{A}-\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 &= \|\mathbf{B}^T(\mathbf{I}-\mathbf{Z}\mathbf{Z}^T)\mathbf{B}\|_F^2\\ &= \sum_{i=1}^{n-k}\sigma_i^2(\mathbf{B}^T(\mathbf{I}-\mathbf{Z}\mathbf{Z}^T)\mathbf{B})\\ &= \sum_{i=1}^{n-k}\sigma_i^4((\mathbf{I}-\mathbf{Z}\mathbf{Z}^T)\mathbf{B}).\end{aligned}$$

- Write $(\mathbf{I}-\mathbf{Z}\mathbf{Z}^T)\mathbf{B}=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ using the SVD and note that $\mathbf{B}^T(\mathbf{I}-\mathbf{Z}\mathbf{Z}^T)\mathbf{B}=\mathbf{B}^T(\mathbf{I}-\mathbf{Z}\mathbf{Z}^T)(\mathbf{I}-\mathbf{Z}\mathbf{Z}^T)\mathbf{B}=\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T=\mathbf{V}^T\boldsymbol{\Sigma}^2\mathbf{V}$.

$$\begin{aligned}
\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 &= \|\mathbf{B}^T(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{B}\|_F^2 \\
&= \sum_{i=1}^{n-k} \sigma_i^2(\mathbf{B}^T(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{B}) \\
&= \sum_{i=1}^{n-k} \sigma_i^4((\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{B}).
\end{aligned}$$

- Write $(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{B} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ using the SVD and note that $\mathbf{B}^T(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{B} = \mathbf{B}^T(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{B} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{V}^T\boldsymbol{\Sigma}^2\mathbf{V}$.
- So the error on $\mathbf{A}$ is just a higher moment of the error on $\mathbf{B}$:

$$\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 = \sum_{i=1}^{n-k} \sigma_i^2(\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}).$$

PROOF OF BOOSTING LEMMA

$$\|\mathbf{A} - \mathbf{B}\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 = \sum_{i=1}^{n-k} \sigma_i^4(\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B})$$

$$\|\mathbf{A} - \mathbf{B}\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 = \sum_{i=1}^{n-k} \sigma_i^4(\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B})$$

Have: $\|\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 - \|\mathbf{B} - \mathbf{B}_k\|_F^2 \leq \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2$

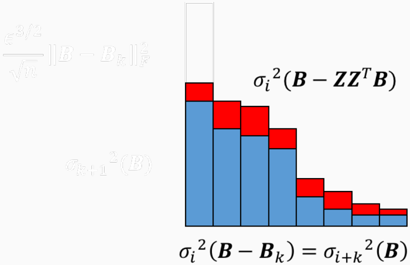
$$\|\mathbf{A} - \mathbf{B}\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 = \sum_{i=1}^{n-k} \sigma_i^4(\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B})$$

Have: $\left[\sum \sigma_i^2(\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B})\right] - \left[\sum \sigma_i^2(\mathbf{B} - \mathbf{B}_k)\right] \leq \frac{\epsilon^{3/2}}{\sqrt{n}} \|\mathbf{B} - \mathbf{B}_k\|_F^2$

PROOF OF BOOSTING LEMMA

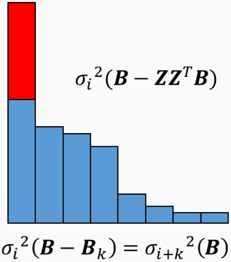
$$\|\mathbf{A} - \mathbf{B}\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 = \sum_{i=1}^{n-k} \sigma_i^4(\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B})$$

Have: $\left[\sum \sigma_i^2(\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B})\right] - \left[\sum \sigma_i^2(\mathbf{B} - \mathbf{B}_k)\right] \leq \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2$



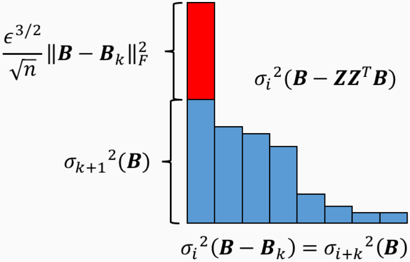
$$\|\mathbf{A} - \mathbf{B}\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 = \sum_{i=1}^{n-k} \sigma_i^4(\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B})$$

Have: $\left[\sum \sigma_i^2(\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B})\right] - \left[\sum \sigma_i^2(\mathbf{B} - \mathbf{B}_k)\right] \leq \frac{\epsilon^{3/2}}{\sqrt{n}} \|\mathbf{B} - \mathbf{B}_k\|_F^2$



$$\|\mathbf{A} - \mathbf{BZZ}^T\mathbf{B}\|_F^2 = \sum_{i=1}^{n-k} \sigma_i^4(\mathbf{B} - \mathbf{ZZ}^T\mathbf{B})$$

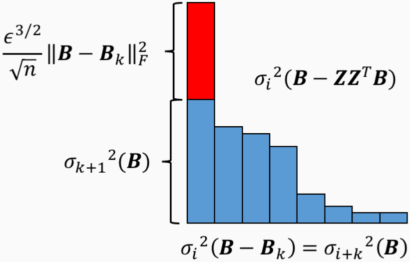
Have: $\left[\sum \sigma_i^2(\mathbf{B} - \mathbf{ZZ}^T\mathbf{B})\right] - \left[\sum \sigma_i^2(\mathbf{B} - \mathbf{B}_k)\right] \leq \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2$



$$\|\mathbf{A} - \mathbf{B}\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 = \sum_{i=1}^{n-k} \sigma_i^4(\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B})$$

$$\leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left[\sigma_{k+1}^2(\mathbf{B}) + \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2\right]^2$$

Have: $\left[\sum \sigma_i^2(\mathbf{B} - \mathbf{Z}\mathbf{Z}^T\mathbf{B})\right] - \left[\sum \sigma_i^2(\mathbf{B} - \mathbf{B}_k)\right] \leq \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2$



PROOF OF BOOSTING LEMMA

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left[\sigma_{k+1}^2(\mathbf{B}) + \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2\right]^2$$

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left[\sigma_{k+1}^2(\mathbf{B}) + \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2\right]^2$$

If $\sigma_{k+1}^2(\mathbf{B}) \geq \sqrt{\frac{\epsilon}{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2$ then can bound as:

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left[\sigma_{k+1}^2(\mathbf{B}) + \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2\right]^2$$

If $\sigma_{k+1}^2(\mathbf{B}) \geq \sqrt{\frac{\epsilon}{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2$ then can bound as:

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + (1+\epsilon)^2\sigma_{k+1}^4(\mathbf{B})$$

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left[\sigma_{k+1}^2(\mathbf{B}) + \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2\right]^2$$

If $\sigma_{k+1}^2(\mathbf{B}) \geq \sqrt{\frac{\epsilon}{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2$ then can bound as:

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + (1+\epsilon)^2\sigma_{k+1}^4(\mathbf{B})$$

$$\leq (1+\epsilon)^2 \sum_{i=k+1}^{n} \sigma_i^4(\mathbf{B})$$

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left[\sigma_{k+1}^2(\mathbf{B}) + \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2\right]^2$$

If $\sigma_{k+1}^2(\mathbf{B}) \geq \sqrt{\frac{\epsilon}{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2$ then can bound as:

$$\begin{aligned}\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 &\leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + (1+\epsilon)^2\sigma_{k+1}^4(\mathbf{B}) \\ &\leq (1+\epsilon)^2 \sum_{i=k+1}^{n} \sigma_i^4(\mathbf{B}) \\ &= (1+\epsilon)^2 \sum_{i=k+1}^{n} \sigma_i^2(\mathbf{A})\end{aligned}$$

$$\|\mathbf{A}-\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left[\sigma_{k+1}^2(\mathbf{B}) + \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B}-\mathbf{B}_k\|_F^2\right]^2$$

If $\sigma_{k+1}^2(\mathbf{B}) \geq \sqrt{\frac{\epsilon}{n}}\|\mathbf{B}-\mathbf{B}_k\|_F^2$ then can bound as:

$$\begin{aligned}
\|\mathbf{A}-\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 &\leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + (1+\epsilon)^2\sigma_{k+1}^4(\mathbf{B}) \\
&\leq (1+\epsilon)^2 \sum_{i=k+1}^{n} \sigma_i^4(\mathbf{B}) \\
&= (1+\epsilon)^2 \sum_{i=k+1}^{n} \sigma_i^2(\mathbf{A}) \\
&= (1+3\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_F^2.
\end{aligned}$$

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left[\sigma_{k+1}^2(\mathbf{B}) + \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2\right]^2$$

If $\sigma_{k+1}^2(\mathbf{B}) \leq \sqrt{\frac{\epsilon}{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2$ then can bound as:

PROOF OF BOOSTING LEMMA

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left[\sigma_{k+1}^2(\mathbf{B}) + \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2\right]^2$$

If $\sigma_{k+1}^2(\mathbf{B}) \leq \sqrt{\frac{\epsilon}{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2$ then can bound as:

$$\|\mathbf{A} - (\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left(\sqrt{\frac{\epsilon}{n}} + \frac{\epsilon^{3/2}}{\sqrt{n}}\right)^2 \|\mathbf{B} - \mathbf{B}_k\|_F^4$$

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left[\sigma_{k+1}^2(\mathbf{B}) + \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2\right]^2$$

If $\sigma_{k+1}^2(\mathbf{B}) \leq \sqrt{\frac{\epsilon}{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2$ then can bound as:

$$\begin{aligned}\|\mathbf{A} - (\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 &\leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left(\sqrt{\frac{\epsilon}{n}} + \frac{\epsilon^{3/2}}{\sqrt{n}}\right)^2 \|\mathbf{B} - \mathbf{B}_k\|_F^4 \\ &\leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \frac{4\epsilon}{n} \cdot \left(\sum_{i=k+1}^{n} \sigma_i^2(\mathbf{B})\right)^2\end{aligned}$$

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left[\sigma_{k+1}^2(\mathbf{B}) + \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2\right]^2$$

If $\sigma_{k+1}^2(\mathbf{B}) \leq \sqrt{\frac{\epsilon}{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2$ then can bound as:

$$\begin{aligned}
\|\mathbf{A} - (\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 &\leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left(\sqrt{\frac{\epsilon}{n}} + \frac{\epsilon^{3/2}}{\sqrt{n}}\right)^2 \|\mathbf{B} - \mathbf{B}_k\|_F^4 \\
&\leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \frac{4\epsilon}{n} \cdot \left(\sum_{i=k+1}^{n} \sigma_i^2(\mathbf{B})\right)^2 \\
&\leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + 4\epsilon \cdot \sum_{i=k+1}^{n} \sigma_i^4(\mathbf{B})
\end{aligned}$$

$$\|\mathbf{A} - \mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left[\sigma_{k+1}^2(\mathbf{B}) + \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2\right]^2$$

If $\sigma_{k+1}^2(\mathbf{B}) \leq \sqrt{\frac{\epsilon}{n}}\|\mathbf{B} - \mathbf{B}_k\|_F^2$ then can bound as:

$$\begin{aligned}
\|\mathbf{A} - (\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 &\leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \left(\sqrt{\frac{\epsilon}{n}} + \frac{\epsilon^{3/2}}{\sqrt{n}}\right)^2 \|\mathbf{B} - \mathbf{B}_k\|_F^4 \\
&\leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + \frac{4\epsilon}{n} \cdot \left(\sum_{i=k+1}^{n} \sigma_i^2(\mathbf{B})\right)^2 \\
&\leq \sum_{i=k+2}^{n} \sigma_i^4(\mathbf{B}) + 4\epsilon \cdot \sum_{i=k+1}^{n} \sigma_i^4(\mathbf{B}) \\
&\leq (1 + 4\epsilon) \sum_{i=k+1}^{n} \sigma_i^2(\mathbf{A})
\end{aligned}$$

$$\|\mathbf{A}-\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 \leq \sum_{i=k+2}^{n}\sigma_i^4(\mathbf{B}) + \left[\sigma_{k+1}^2(\mathbf{B}) + \frac{\epsilon^{3/2}}{\sqrt{n}}\|\mathbf{B}-\mathbf{B}_k\|_F^2\right]^2$$

If $\sigma_{k+1}^2(\mathbf{B}) \leq \sqrt{\frac{\epsilon}{n}}\|\mathbf{B}-\mathbf{B}_k\|_F^2$ then can bound as:

$$\begin{aligned}
\|\mathbf{A}-(\mathbf{B}^T\mathbf{Z}\mathbf{Z}^T\mathbf{B}\|_F^2 &\leq \sum_{i=k+2}^{n}\sigma_i^4(\mathbf{B}) + \left(\sqrt{\frac{\epsilon}{n}}+\frac{\epsilon^{3/2}}{\sqrt{n}}\right)^2\|\mathbf{B}-\mathbf{B}_k\|_F^4 \\
&\leq \sum_{i=k+2}^{n}\sigma_i^4(\mathbf{B}) + \frac{4\epsilon}{n}\cdot\left(\sum_{i=k+1}^{n}\sigma_i^2(\mathbf{B})\right)^2 \\
&\leq \sum_{i=k+2}^{n}\sigma_i^4(\mathbf{B}) + 4\epsilon\cdot\sum_{i=k+1}^{n}\sigma_i^4(\mathbf{B}) \\
&\leq (1+4\epsilon)\sum_{i=k+1}^{n}\sigma_i^2(\mathbf{A}) = (1+4\epsilon)\|\mathbf{A}-\mathbf{A}_k\|_F^2.
\end{aligned}$$

Theorem ('Slow' Sublinear Time Low-Rank Approximation)

There is an algorithm which given PSD $\mathbf{A} \in \mathbb{R}^{n \times n}$ *accesses* $O\left(n^{3/2} \cdot \text{poly}(k, 1/\epsilon)\right)$ *entries of the matrix and outputs* $\mathbf{N}, \mathbf{M} \in \mathbb{R}^{n \times k}$ *which satisfy with probability* $99/100$*:*

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

Theorem ('Slow' Sublinear Time Low-Rank Approximation)

There is an algorithm which given PSD $\mathbf{A} \in \mathbb{R}^{n \times n}$ *accesses* $O\left(n^{3/2} \cdot \text{poly}(k, 1/\epsilon)\right)$ *entries of the matrix and outputs* $\mathbf{N}, \mathbf{M} \in \mathbb{R}^{n \times k}$ *which satisfy with probability* $99/100$*:*

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

- The algorithm can be shown to run in $n^{1.69} \cdot \text{poly}(k, 1/\epsilon)$ time using fast matrix multiplication.

Theorem ('Slow' Sublinear Time Low-Rank Approximation)

There is an algorithm which given PSD $\mathbf{A} \in \mathbb{R}^{n \times n}$ *accesses* $O\left(n^{3/2} \cdot \text{poly}(k, 1/\epsilon)\right)$ *entries of the matrix and outputs* $\mathbf{N}, \mathbf{M} \in \mathbb{R}^{n \times k}$ *which satisfy with probability* $99/100$*:*

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

- The algorithm can be shown to run in $n^{1.69} \cdot \text{poly}(k, 1/\epsilon)$ time using fast matrix multiplication.
- Our best algorithm accesses just $\tilde{O}\left(\frac{nk}{\epsilon^{2.5}}\right)$ entries of $\mathbf{A}$ and runs in $\tilde{O}\left(\frac{nk^2}{\epsilon^4}\right)$ time.

Theorem ('Slow' Sublinear Time Low-Rank Approximation)

There is an algorithm which given PSD $\mathbf{A} \in \mathbb{R}^{n \times n}$ *accesses* $O\left(n^{3/2} \cdot \mathrm{poly}(k, 1/\epsilon)\right)$ *entries of the matrix and outputs* $\mathbf{N}, \mathbf{M} \in \mathbb{R}^{n \times k}$ *which satisfy with probability* $99/100$*:*

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

- The algorithm can be shown to run in $n^{1.69} \cdot \mathrm{poly}(k, 1/\epsilon)$ time using fast matrix multiplication.
- Our best algorithm accesses just $\tilde{O}\left(\frac{nk}{\epsilon^{2.5}}\right)$ entries of $\mathbf{A}$ and runs in $\tilde{O}\left(\frac{nk^2}{\epsilon^4}\right)$ time. Query complexity is optimal up to a $\frac{1}{\epsilon^{1.5}}$ factor.

Theorem ('Slow' Sublinear Time Low-Rank Approximation)

There is an algorithm which given PSD $\mathbf{A} \in \mathbb{R}^{n \times n}$ *accesses* $O\left(n^{3/2} \cdot \text{poly}(k, 1/\epsilon)\right)$ *entries of the matrix and outputs* $\mathbf{N}, \mathbf{M} \in \mathbb{R}^{n \times k}$ *which satisfy with probability* $99/100$*:*

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

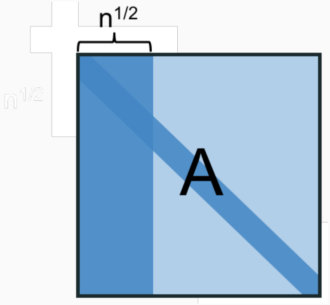
- The algorithm can be shown to run in $n^{1.69} \cdot \text{poly}(k, 1/\epsilon)$ time using fast matrix multiplication.
- Our best algorithm accesses just $\tilde{O}\left(\frac{nk}{\epsilon^{2.5}}\right)$ entries of $\mathbf{A}$ and runs in $\tilde{O}\left(\frac{nk^2}{\epsilon^4}\right)$ time. Query complexity is optimal up to a $\frac{1}{\epsilon^{1.5}}$ factor. How can we achieve this?

Recall that our algorithm is based off adaptive sampling - we iteratively select $\tilde{O}(k^2/\epsilon)$ columns of $\mathbf{B}$ and project to them.

Recall that our algorithm is based off adaptive sampling - we iteratively select $\tilde{O}(k^2 \cdot \sqrt{n})$ columns of $\mathbf{B}$ and project to them.

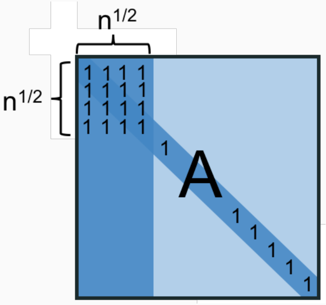
Recall that our algorithm is based off adaptive sampling - we iteratively select $\tilde{O}(k^2 \cdot \sqrt{n})$ columns of $\mathbf{B}$ and project to them.

- Requires accessing the diagonal and $\tilde{O}(\sqrt{n}k^2)$ columns of $\mathbf{A}$.



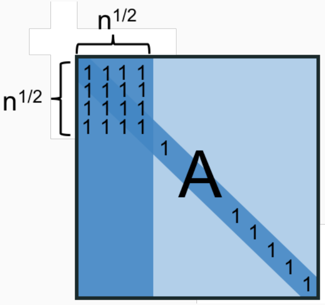
Recall that our algorithm is based off adaptive sampling - we iteratively select $\tilde{O}(k^2 \cdot \sqrt{n})$ columns of **B** and project to them.

- Requires accessing the diagonal and $\tilde{O}(\sqrt{n}k^2)$ columns of **A**.

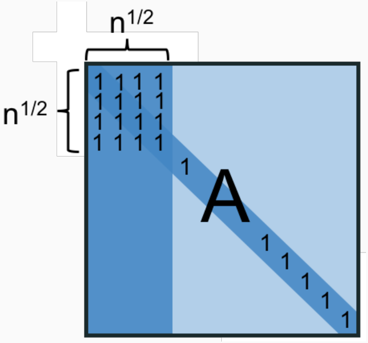


Recall that our algorithm is based off adaptive sampling - we iteratively select $\tilde{O}(k^2 \cdot \sqrt{n})$ columns of **B** and project to them.

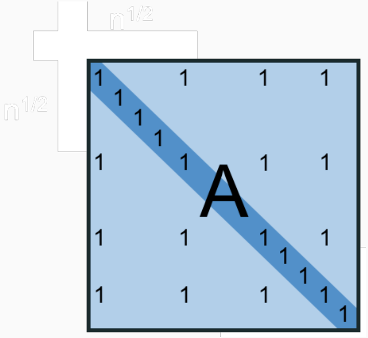
- Requires accessing the diagonal and $\tilde{O}(\sqrt{n}k^2)$ columns of **A**.



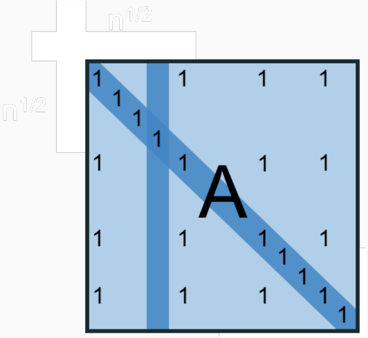
- If we take fewer columns, we can miss a $\sqrt{n} \times \sqrt{n}$ block which contains a constant fraction of **A**'s Frobenius norm.


n1/2
n1/2
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1
A
1
1
1
1
1

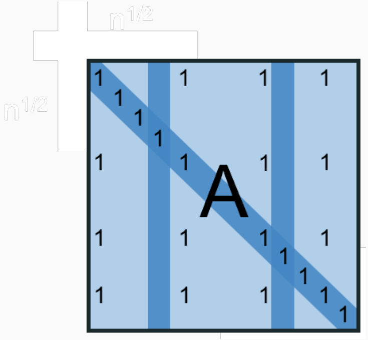
LIMITATIONS OF COLUMN SAMPLING



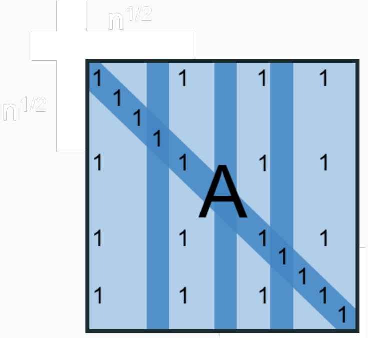
- Probability that a column sample hits a single off diagonal entry is $O(1/\sqrt{n})$.



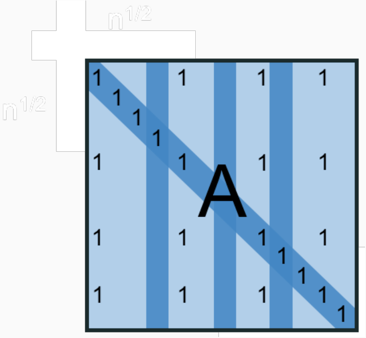
- Probability that a column sample hits a single off diagonal entry is $O(1/\sqrt{n})$.



- Probability that a column sample hits a single off diagonal entry is $O(1/\sqrt{n})$.

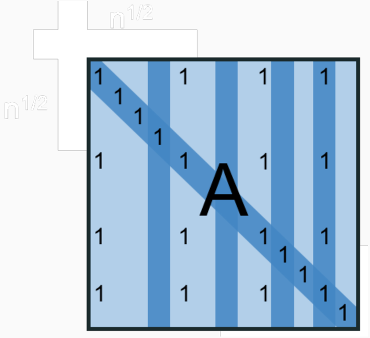


- Probability that a column sample hits a single off diagonal entry is $O(1/\sqrt{n})$.



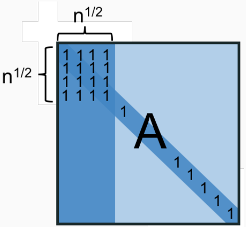
- Probability that a column sample hits a single off diagonal entry is $O(1/\sqrt{n})$.
- So $\Omega(\sqrt{n})$ samples are required to find the block.

LIMITATIONS OF COLUMN SAMPLING

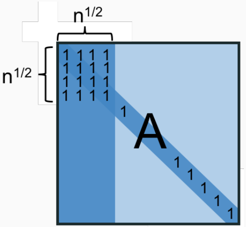


- Probability that a column sample hits a single off diagonal entry is $O(1/\sqrt{n})$.
- So $\Omega(\sqrt{n})$ samples are required to find the block.

LIMITATIONS OF COLUMN SAMPLING



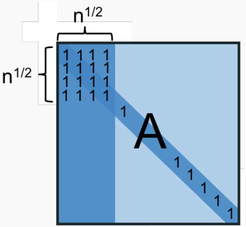
Highlights the difference between low-rank approximation of **A** and its square root **B**.



Highlights the difference between low-rank approximation of **A** and its square root **B**.

- $\sigma_1^2(\mathbf{A}) = n$ and $\|\mathbf{A} - \mathbf{A}_1\|_F^2 \approx n$. Even obtaining a 2-approximation to the best rank-1 approximation requires finding the block.

LIMITATIONS OF COLUMN SAMPLING



Highlights the difference between low-rank approximation of **A** and its square root **B**.

- $\sigma_1^2(\mathbf{A}) = n$ and $\|\mathbf{A} - \mathbf{A}_1\|_F^2 \approx n$. Even obtaining a 2-approximation to the best rank-1 approximation requires finding the block.
- $\sigma_1^2(\mathbf{B}) = \sqrt{n}$ and $\|\mathbf{B} - \mathbf{B}_1\|_F^2 \approx n$, so the block does not need to be recovered to obtain a $\left(1 + \frac{1}{\sqrt{n}}\right)$-optimal approximation.

Solution: Sample both rows and columns of $\mathbf{A}$.

Solution: Sample both rows and columns of $\mathbf{A}$.

- Instead of adaptive sampling we use ridge leverage scores, which can also be computed using an iterative sampling scheme making $\tilde{O}(nk)$ accesses to $\mathbf{A}$ (Musco, Musco '17).

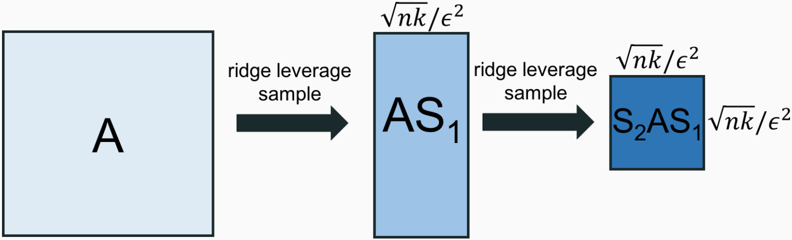
Solution: Sample both rows and columns of $\mathbf{A}$.

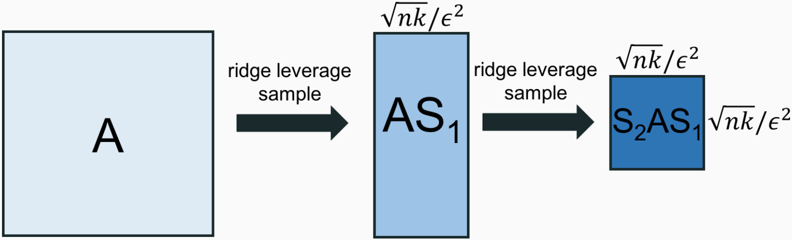
- Instead of adaptive sampling we use ridge leverage scores, which can also be computed using an iterative sampling scheme making $\tilde{O}(nk)$ accesses to $\mathbf{A}$ (Musco, Musco '17).
- Same intuition – select a diverse set of columns which span a near-optimal low-rank approximation of the matrix. However come with much stronger guarantees.

Solution: Sample both rows and columns of $\mathbf{A}$.

- Instead of adaptive sampling we use ridge leverage scores, which can also be computed using an iterative sampling scheme making $\tilde{O}(nk)$ accesses to $\mathbf{A}$ (Musco, Musco '17).
- Same intuition – select a diverse set of columns which span a near-optimal low-rank approximation of the matrix. However come with much stronger guarantees.
- Sample $\mathbf{AS}$ is a projection-cost-preserving sketch for $\mathbf{A}$. For any rank-k projection $\mathbf{P}$,

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


A
ridge leverage sample
$\sqrt{nk}/\epsilon^2$
AS_1
ridge leverage sample
$\sqrt{nk}/\epsilon^2$
S_2AS_1
$\sqrt{nk}/\epsilon^2$

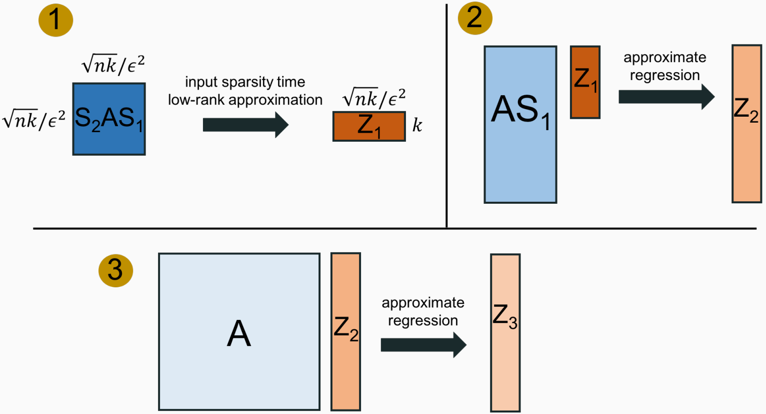


Technical Challenge: Proving that $\mathbf{S}_2\mathbf{A}\mathbf{S}_1$ is a projection-cost preserving sketch of $\mathbf{A}\mathbf{S}_1$.

FINAL ALGORITHM

Recover low-rank approximation using projection-cost preserving sketches.

Recover low-rank approximation using projection-cost preserving sketches.



- View each entry of $\mathbf{A}$ as encoding a large amount of information about its square root $\mathbf{B}$. In particular $\mathbf{a}_{ij} = \mathbf{b}_i^T \mathbf{b}_j$.

SUMMARY OF MAIN IDEAS

- View each entry of $\mathbf{A}$ as encoding a large amount of information about its square root $\mathbf{B}$. In particular $\mathbf{a}_{ij} = \mathbf{b}_i^T \mathbf{b}_j$.
- Use this view to find a low-rank approximation to $\mathbf{B}$ using sublinear accesses to $\mathbf{A}$.

- View each entry of $\mathbf{A}$ as encoding a large amount of information about its square root $\mathbf{B}$. In particular $\mathbf{a}_{ij} = \mathbf{b}_i^T \mathbf{b}_j$.
- Use this view to find a low-rank approximation to $\mathbf{B}$ using sublinear accesses to $\mathbf{A}$.
- Since $\mathbf{B}$ has the same singular vectors as $\mathbf{A}$ and $\sigma_i(\mathbf{B}) = \sqrt{\sigma_i(\mathbf{A})}$, a low-rank approximation of $\mathbf{B}$ can used to find one for $\mathbf{A}$, albiet with a $\sqrt{n}$ factor loss in quality.

- View each entry of $\mathbf{A}$ as encoding a large amount of information about its square root $\mathbf{B}$. In particular $\mathbf{a}_{ij} = \mathbf{b}_i^T \mathbf{b}_j$.
- Use this view to find a low-rank approximation to $\mathbf{B}$ using sublinear accesses to $\mathbf{A}$.
- Since $\mathbf{B}$ has the same singular vectors as $\mathbf{A}$ and $\sigma_i(\mathbf{B}) = \sqrt{\sigma_i(\mathbf{A})}$, a low-rank approximation of $\mathbf{B}$ can used to find one for $\mathbf{A}$, albiet with a $\sqrt{n}$ factor loss in quality.
- Obtain near-optimal complexity using ridge leverage scores to sample both rows and columns of $\mathbf{A}$.

OPEN QUESTIONS

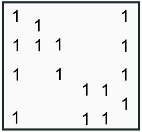
OPEN QUESTIONS

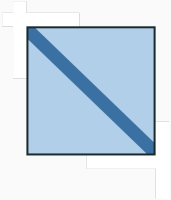
- What else can be done for PSD matrices? We give applications to ridge regression, but what other linear algebraic problems require a second look?

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?

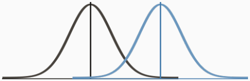
- What else can be done for PSD matrices? We give applications to ridge regression, but what other linear algebraic problems require a second look?
- Are there other natural classes of matrices that admit sublinear time low-rank approximation?
  - Starting points are matrices that break the $\Omega(\text{nnz}(\mathbf{A}))$ time lower bound: e.g. binary matrices, diagonally dominant matrices.





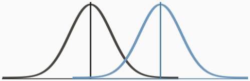
- What can we do when we have PSD matrices with additional structure?

- What can we do when we have PSD matrices with additional structure? E.g. kernel matrices.



$$\langle \mathbf{x}, \mathbf{y} \rangle = e^{-\|\mathbf{x}-\mathbf{y}\|_2^2}$$

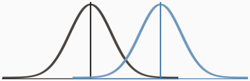
- What can we do when we have PSD matrices with additional structure? E.g. kernel matrices.



$$\langle \mathbf{x}, \mathbf{y} \rangle = e^{-\|\mathbf{x}-\mathbf{y}\|_2^2}$$

- Can apply our algorithm – accessing an entry of $\mathbf{A}$ is equivalent to computing a single kernel dot product. But in some cases you may be able to something smarter.

- What can we do when we have PSD matrices with additional structure? E.g. kernel matrices.



$$\langle \mathbf{x}, \mathbf{y} \rangle = e^{-\|\mathbf{x}-\mathbf{y}\|_2^2}$$

- Can apply our algorithm – accessing an entry of $\mathbf{A}$ is equivalent to computing a single kernel dot product. But in some cases you may be able to something smarter.
- Low-rank approximation of the square root kernel matrix (the 'kernelized dataset') is also interesting here.

Thanks! Questions?