Uniform Sampling for Matrix Approximation

Michael Cohen, Yin Tat Lee, **Cameron Musco**, Christopher Musco,
Richard Peng, Aaron Sidford

Massachusetts Institute of Technology

November 20, 2014

Overview

Goal

- Reduce large matrix $\mathbf{A}$ to some smaller matrix $\tilde{\mathbf{A}}$. Use $\tilde{\mathbf{A}}$ to approximate solution to some problem - e.g. regression.

Main Result

- Simple and efficient *iterative sampling* algorithms for matrix approximation.
- Alternatives to Johnson-Lindenstrauss (random projection) type approaches

Main technique

- Understanding what information is preserved when we sample rows of matrix *uniformly at random*

Overview

Goal

- Reduce large matrix $\mathbf{A}$ to some smaller matrix $\tilde{\mathbf{A}}$. Use $\tilde{\mathbf{A}}$ to approximate solution to some problem - e.g. regression.

Main Result

- Simple and efficient *iterative sampling* algorithms for matrix approximation.
- Alternatives to Johnson-Lindenstrauss (random projection) type approaches

Main technique

- Understanding what information is preserved when we sample rows of matrix *uniformly at random*

Overview

Goal

- Reduce large matrix $\mathbf{A}$ to some smaller matrix $\tilde{\mathbf{A}}$. Use $\tilde{\mathbf{A}}$ to approximate solution to some problem - e.g. regression.

Main Result

- Simple and efficient *iterative sampling* algorithms for matrix approximation.
- Alternatives to Johnson-Lindenstrauss (random projection) type approaches

Main technique

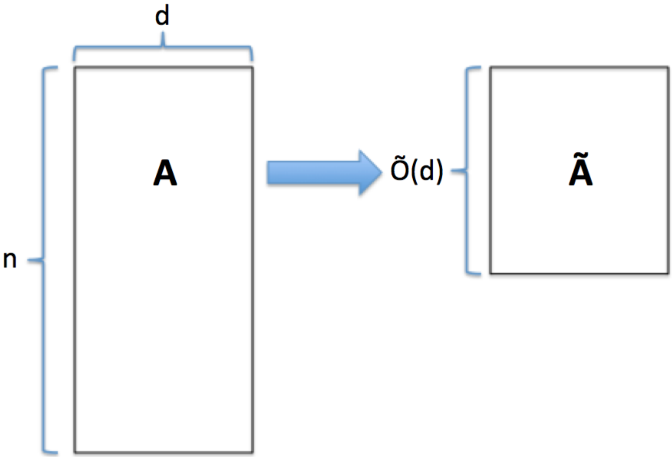
- Understanding what information is preserved when we sample rows of matrix *uniformly at random*

Outline

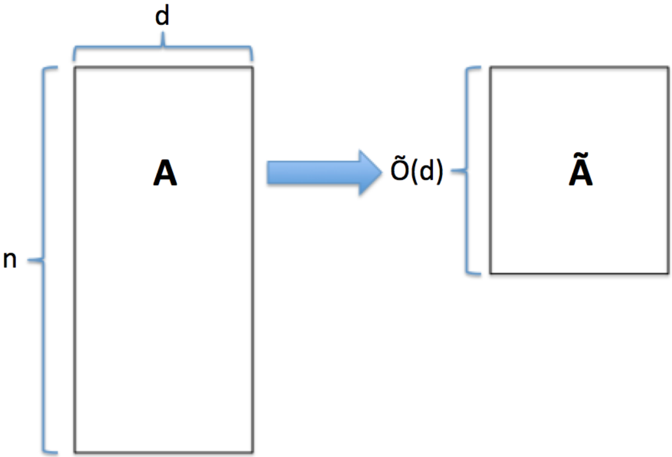
1. Spectral Matrix Approximation
2. Leverage Score Sampling
3. Iterative Leverage Score Computation
4. Coherence Reducing Reweighting

Overview

Spectral Matrix Approximation



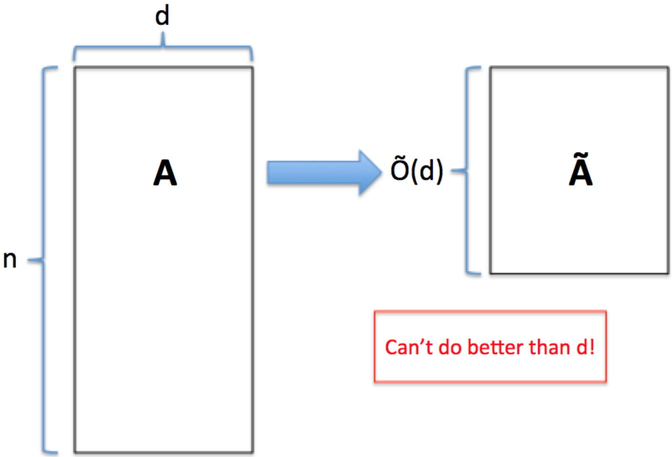
Spectral Matrix Approximation



Goal

$$(1-\epsilon)\|\mathbf{A}\mathbf{x}\|_2^2 \leq \|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 \leq (1+\epsilon)\|\mathbf{A}\mathbf{x}\|_2^2$$

Spectral Matrix Approximation

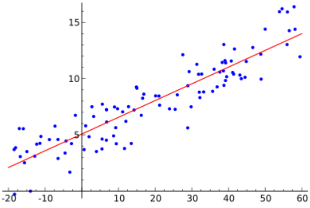


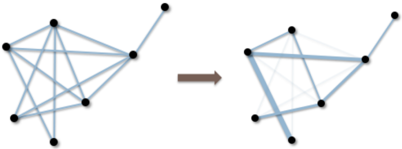
Goal

$$(1-\epsilon)\|\mathbf{Ax}\|_2^2 \leq \|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 \leq (1+\epsilon)\|\mathbf{Ax}\|_2^2$$

Applications

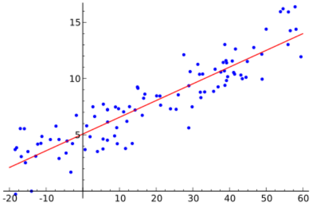
- Approximate linear algebra - e.g. regression
- Preconditioning
- Spectral Graph Sparsification
- Etc...

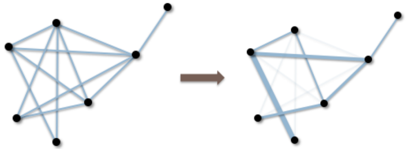




Applications

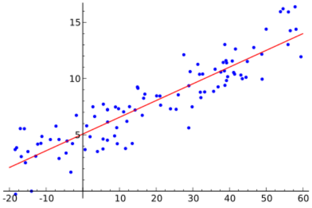
- Approximate linear algebra - e.g. regression
- Preconditioning
- Spectral Graph Sparsification
- Etc...

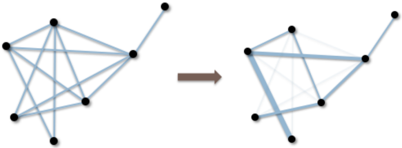




Applications

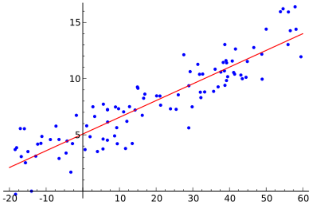
- Approximate linear algebra - e.g. regression
- Preconditioning
- Spectral Graph Sparsification
- Etc...

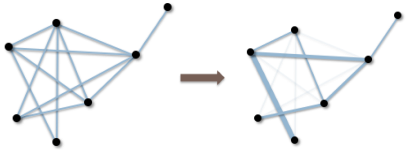




Applications

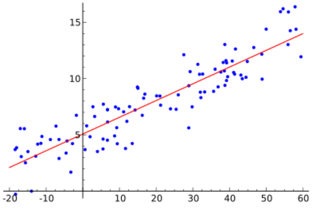
- Approximate linear algebra - e.g. regression
- Preconditioning
- Spectral Graph Sparsification
- Etc...

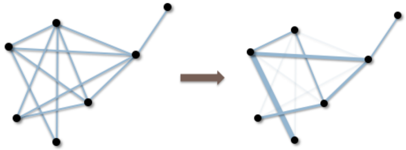




Applications

- Approximate linear algebra - e.g. regression
- Preconditioning
- Spectral Graph Sparsification
- Etc...





Regression

Goal:

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

Solve:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \Longrightarrow \mathbf{A}^\top(\mathbf{A}\mathbf{x}) = \mathbf{A}^\top\mathbf{b}$$

Set x to:

$$(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$$

Problem:

$\mathbf{A}$ is very tall. Too slow to compute $\mathbf{A}^\top\mathbf{A}$.

Regression

Goal:

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

Solve:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \implies \mathbf{A}^\top(\mathbf{A}\mathbf{x}) = \mathbf{A}^\top\mathbf{b}$$

Set x to:

$$(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$$

Problem:

$\mathbf{A}$ is very tall. Too slow to compute $\mathbf{A}^\top\mathbf{A}$.

Regression

Goal:

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

Solve:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \Longrightarrow \mathbf{A}^\top(\mathbf{A}\mathbf{x}) = \mathbf{A}^\top\mathbf{b}$$

Set x to:

$$(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$$

Problem:

$\mathbf{A}$ is very tall. Too slow to compute $\mathbf{A}^\top\mathbf{A}$.

Regression

Goal:

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

Solve:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \Longrightarrow \mathbf{A}^\top(\mathbf{A}\mathbf{x}) = \mathbf{A}^\top\mathbf{b}$$

Set x to:

$$(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$$

Problem:

$\mathbf{A}$ is very tall. Too slow to compute $\mathbf{A}^\top\mathbf{A}$.

Regression

Goal:

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

Solve:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \implies \mathbf{A}^\top(\mathbf{A}\mathbf{x}) = \mathbf{A}^\top\mathbf{b}$$

Set x to:

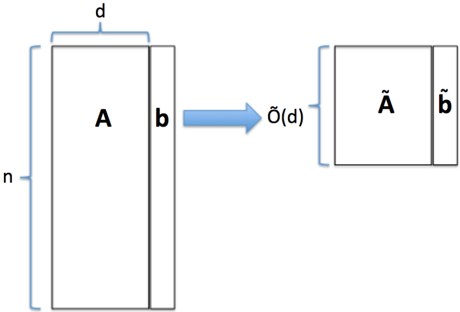
$$(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$$

Problem:

$\mathbf{A}$ is very tall. Too slow to compute $\mathbf{A}^\top\mathbf{A}$.

Regression

Solution:

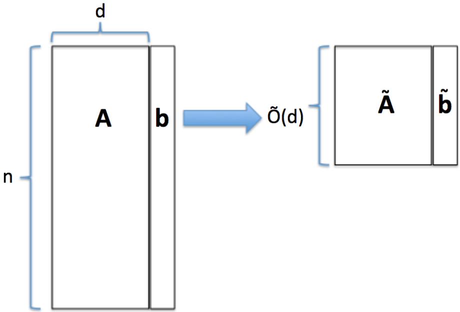


$$\|\mathbf{Ax}-\mathbf{b}\|_2^2 = \left\| \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} \right\|_2^2 \approx_\epsilon \left\| \begin{bmatrix} \tilde{\mathbf{A}} & \tilde{\mathbf{b}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} \right\|_2^2$$

$$\tilde{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{x}} \|\tilde{\mathbf{A}}\mathbf{x} - \tilde{\mathbf{b}}\|_2^2$$

Regression

Solution:

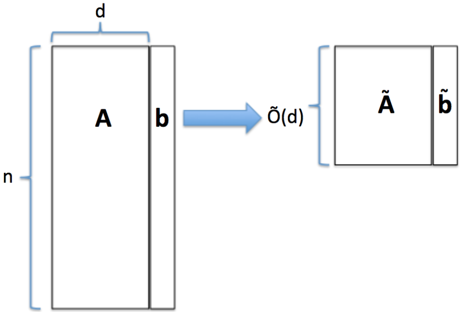


$$\|\mathbf{Ax}-\mathbf{b}\|_2^2 = \left\| \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} \right\|_2^2 \approx_\epsilon \left\| \begin{bmatrix} \tilde{\mathbf{A}} & \tilde{\mathbf{b}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} \right\|_2^2$$

$$\tilde{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{x}} \|\tilde{\mathbf{A}}\mathbf{x} - \tilde{\mathbf{b}}\|_2^2$$

Regression

Solution:

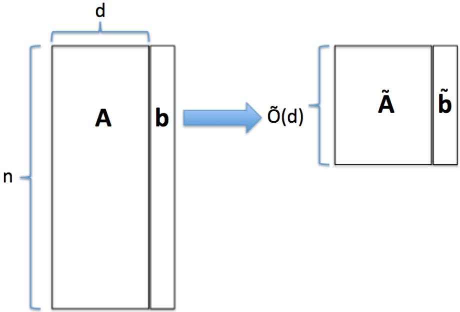


$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \left\| \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} \right\|_2^2 \approx_\epsilon \left\| \begin{bmatrix} \tilde{\mathbf{A}} & \tilde{\mathbf{b}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} \right\|_2^2$$

$$\tilde{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{x}} \|\tilde{\mathbf{A}}\mathbf{x} - \tilde{\mathbf{b}}\|_2^2$$

Regression

Solution:



Or, use preconditioned iterative method:

$$\kappa\left((\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1}(\mathbf{A}^\top\mathbf{A})\right) = O(1)$$

.

Spectral Matrix Approximation

All equivalent:

- Norm:

$$\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\|_2^2$$

- Quadratic Form:

$$\mathbf{x}^\top(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})\mathbf{x} = (1 \pm \epsilon)\mathbf{x}^\top(\mathbf{A}^\top\mathbf{A})\mathbf{x}$$

- Loewner Ordering:

$$(1-\epsilon)\mathbf{A}^\top\mathbf{A} \preceq \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} \preceq (1+\epsilon)\mathbf{A}^\top\mathbf{A}$$

- Inverse:

$$\frac{1}{(1+\epsilon)}(\mathbf{A}^\top\mathbf{A})^{-1} \preceq (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1} \preceq \frac{1}{(1-\epsilon)}(\mathbf{A}^\top\mathbf{A})^{-1}$$

Spectral Matrix Approximation

All equivalent:

- Norm:

$$\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\|_2^2$$

- Quadratic Form:

$$\mathbf{x}^\top(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})\mathbf{x} = (1 \pm \epsilon)\mathbf{x}^\top(\mathbf{A}^\top\mathbf{A})\mathbf{x}$$

- Loewner Ordering:

$$(1-\epsilon)\mathbf{A}^\top\mathbf{A} \preceq \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} \preceq (1+\epsilon)\mathbf{A}^\top\mathbf{A}$$

- Inverse:

$$\frac{1}{(1+\epsilon)}(\mathbf{A}^\top\mathbf{A})^{-1} \preceq (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1} \preceq \frac{1}{(1-\epsilon)}(\mathbf{A}^\top\mathbf{A})^{-1}$$

Spectral Matrix Approximation

All equivalent:

- Norm:

$$\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\|_2^2$$

- Quadratic Form:

$$\mathbf{x}^\top(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})\mathbf{x} = (1 \pm \epsilon)\mathbf{x}^\top(\mathbf{A}^\top\mathbf{A})\mathbf{x}$$

- Loewner Ordering:

$$(1-\epsilon)\mathbf{A}^\top\mathbf{A} \preceq \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} \preceq (1+\epsilon)\mathbf{A}^\top\mathbf{A}$$

- Inverse:

$$\frac{1}{(1+\epsilon)}(\mathbf{A}^\top\mathbf{A})^{-1} \preceq (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1} \preceq \frac{1}{(1-\epsilon)}(\mathbf{A}^\top\mathbf{A})^{-1}$$

Spectral Matrix Approximation

All equivalent:

- Norm:

$$\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\|_2^2$$

- Quadratic Form:

$$\mathbf{x}^\top(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})\mathbf{x} = (1 \pm \epsilon)\mathbf{x}^\top(\mathbf{A}^\top\mathbf{A})\mathbf{x}$$

- Loewner Ordering:

$$(1-\epsilon)\mathbf{A}^\top\mathbf{A} \preceq \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} \preceq (1+\epsilon)\mathbf{A}^\top\mathbf{A}$$

- Inverse:

$$\frac{1}{(1+\epsilon)}(\mathbf{A}^\top\mathbf{A})^{-1} \preceq (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1} \preceq \frac{1}{(1-\epsilon)}(\mathbf{A}^\top\mathbf{A})^{-1}$$

Spectral Matrix Approximation

All equivalent:

- Norm:

$$\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\|_2^2$$

- Quadratic Form:

$$\mathbf{x}^\top(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})\mathbf{x} = (1 \pm \epsilon)\mathbf{x}^\top(\mathbf{A}^\top\mathbf{A})\mathbf{x}$$

- Loewner Ordering:

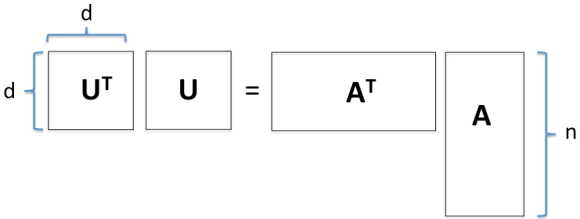
$$(1-\epsilon)\mathbf{A}^\top\mathbf{A} \preceq \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} \preceq (1+\epsilon)\mathbf{A}^\top\mathbf{A}$$

- Inverse:

$$\frac{1}{(1+\epsilon)}(\mathbf{A}^\top\mathbf{A})^{-1} \preceq (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1} \preceq \frac{1}{(1-\epsilon)}(\mathbf{A}^\top\mathbf{A})^{-1}$$

How to Find a Spectral Approximation?

Just take a matrix square root of $\mathbf{A}^\top\mathbf{A}$!

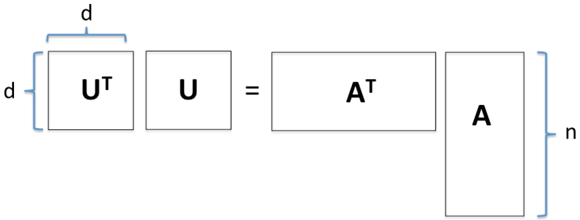


$$\mathbf{U}^\top\mathbf{U} = \mathbf{A}^\top\mathbf{A} \implies \|\mathbf{U}\mathbf{x}\|_2^2 = \|\mathbf{A}\mathbf{x}\|_2^2$$

- Cholesky decomposition, SVD, etc. give $\mathbf{U} \in \mathbb{R}^{d\times d}$
- Runs in something like $O(nd^2)$ time.

How to Find a Spectral Approximation?

Just take a matrix square root of $\mathbf{A}^\top\mathbf{A}$!

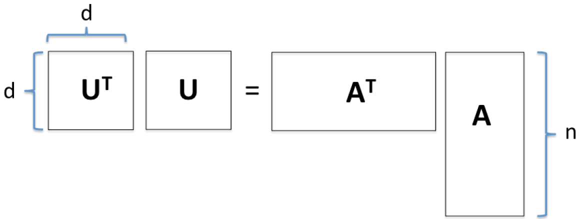


$$\mathbf{U}^\top\mathbf{U} = \mathbf{A}^\top\mathbf{A} \implies \|\mathbf{U}\mathbf{x}\|_2^2 = \|\mathbf{A}\mathbf{x}\|_2^2$$

- Cholesky decomposition, SVD, etc. give $\mathbf{U} \in \mathbb{R}^{d\times d}$
- Runs in something like $O(nd^2)$ time.

How to Find a Spectral Approximation?

Just take a matrix square root of $\mathbf{A}^\top\mathbf{A}$!

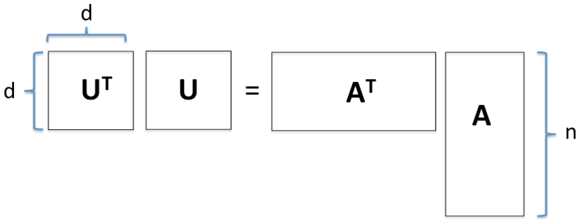


$$\mathbf{U}^\top\mathbf{U} = \mathbf{A}^\top\mathbf{A} \implies \|\mathbf{U}\mathbf{x}\|_2^2 = \|\mathbf{A}\mathbf{x}\|_2^2$$

- Cholesky decomposition, SVD, etc. give $\mathbf{U} \in \mathbb{R}^{d\times d}$
- Runs in something like $O(nd^2)$ time.

How to Find a Spectral Approximation?

Just take a matrix square root of $\mathbf{A}^\top\mathbf{A}$!



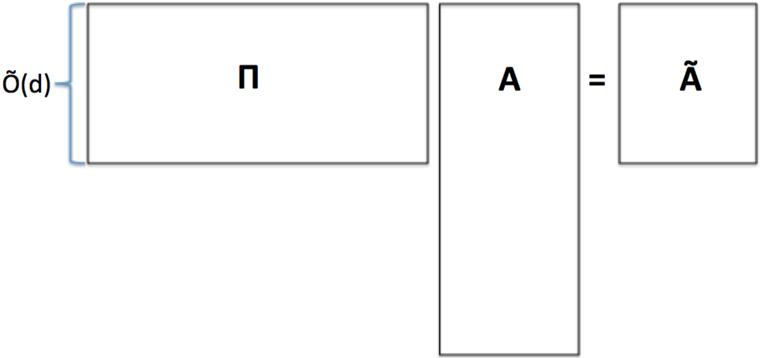
$$\mathbf{U}^\top\mathbf{U} = \mathbf{A}^\top\mathbf{A} \implies \|\mathbf{U}\mathbf{x}\|_2^2 = \|\mathbf{A}\mathbf{x}\|_2^2$$

- Cholesky decomposition, SVD, etc. give $\mathbf{U} \in \mathbb{R}^{d\times d}$
- Runs in something like $O(nd^2)$ time.

How to Find one Faster?

Use Subspace Embeddings [Sarlos '06], [Nelson, Nguyen '13], [Clarkson, Woodruff '13], [Mahoney, Meng '13]

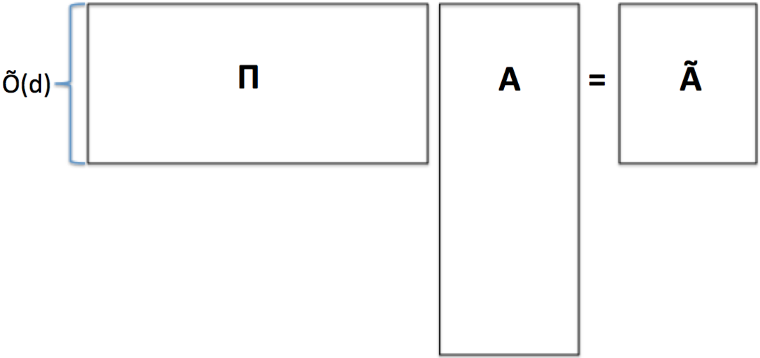
- Left multiply by sparse 'Johnson-Lindenstrauss matrix'.
- Can apply in $O(nnz(\mathbf{A}))$ time.
- Reduce to $O(d/\epsilon^2)$ dimensions.



How to Find one Faster?

Use Subspace Embeddings [Sarlos '06], [Nelson, Nguyen '13], [Clarkson, Woodruff '13], [Mahoney, Meng '13]

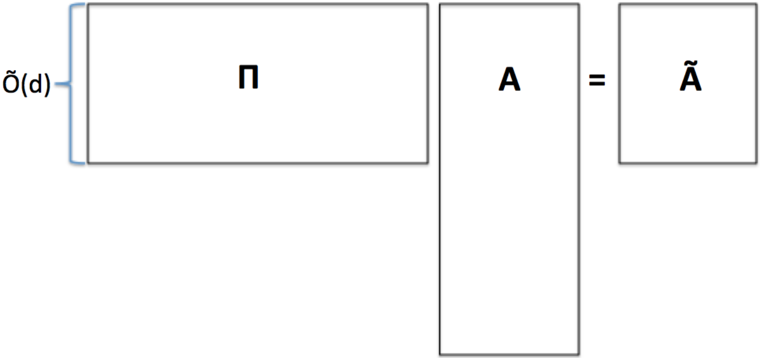
- Left multiply by sparse 'Johnson-Lindenstrauss matrix'.
- Can apply in $O(nnz(\mathbf{A}))$ time.
- Reduce to $O(d/\epsilon^2)$ dimensions.



How to Find one Faster?

Use Subspace Embeddings [Sarlos '06], [Nelson, Nguyen '13], [Clarkson, Woodruff '13], [Mahoney, Meng '13]

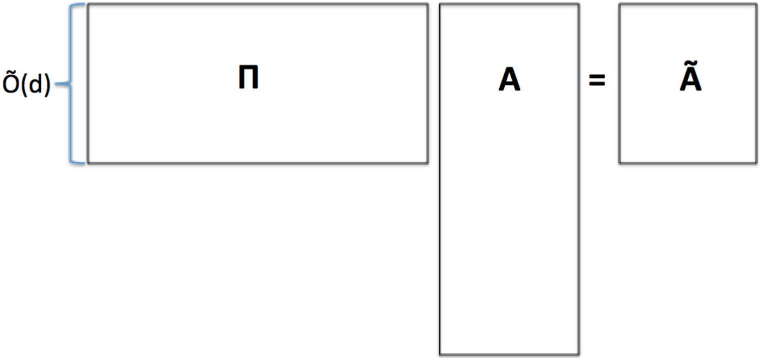
- Left multiply by sparse 'Johnson-Lindenstrauss matrix'.
- Can apply in $O(nnz(\mathbf{A}))$ time.
- Reduce to $O(d/\epsilon^2)$ dimensions.



How to Find one Faster?

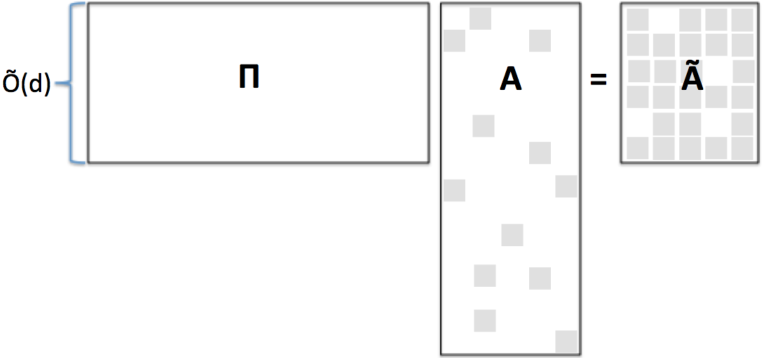
Use Subspace Embeddings [Sarlos '06], [Nelson, Nguyen '13], [Clarkson, Woodruff '13], [Mahoney, Meng '13]

- Left multiply by sparse 'Johnson-Lindenstrauss matrix'.
- Can apply in $O(nnz(\mathbf{A}))$ time.
- Reduce to $O(d/\epsilon^2)$ dimensions.



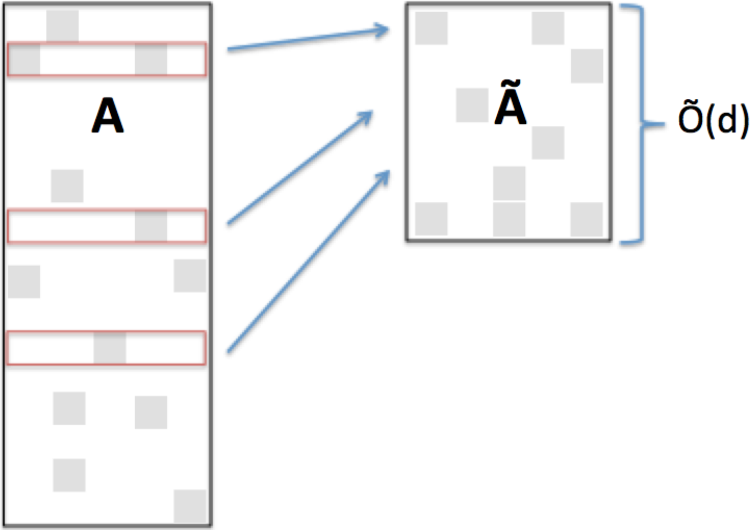
How to Find one Faster?

- ‘Squishes’ together rows



What if we want to preserve structure/sparsity?

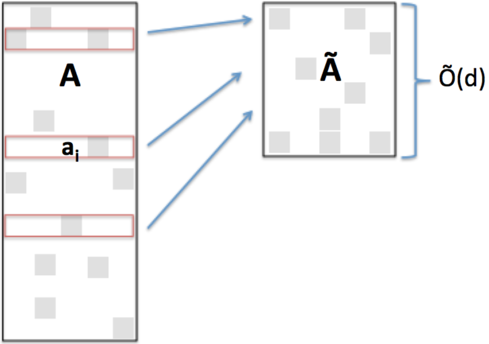
Use Row Sampling



Overview

Leverage Score Sampling

- Sample rows with probability proportional to *leverage scores*.

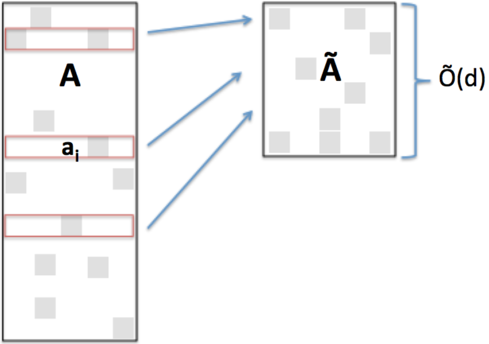


Leverage Score Sampling

- Sample rows with probability proportional to *leverage scores*.

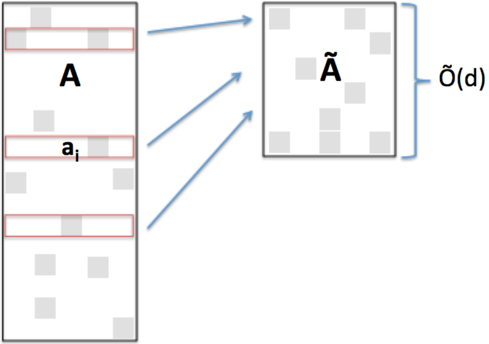
Leverage Score

$\tau_i(\mathbf{A}) = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{a}_i$



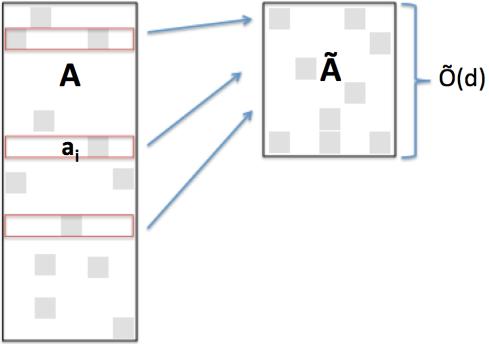
Leverage Score Sampling

- Sample each row independently with $p_i = O(\tau_i(\mathbf{A}) \log d/\epsilon^2)$.
- $\sum_i \tau_i(\mathbf{A}) = d$ giving reduction to $O(d \log d/\epsilon^2)$ rows.
- Straight-forward analysis with matrix Chernoff bounds.



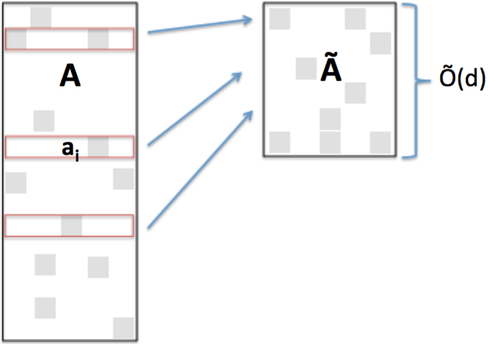
Leverage Score Sampling

- Sample each row independently with $p_i = O(\tau_i(\mathbf{A}) \log d/\epsilon^2)$.
- $\sum_i \tau_i(\mathbf{A}) = d$ giving reduction to $O(d \log d/\epsilon^2)$ rows.
- Straight-forward analysis with matrix Chernoff bounds.



Leverage Score Sampling

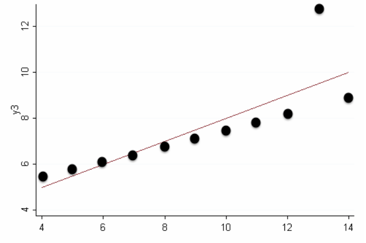
- Sample each row independently with $p_i = O(\tau_i(\mathbf{A}) \log d/\epsilon^2)$.
- $\sum_i \tau_i(\mathbf{A}) = d$ giving reduction to $O(d \log d/\epsilon^2)$ rows.
- Straight-forward analysis with matrix Chernoff bounds.



What is the leverage score?

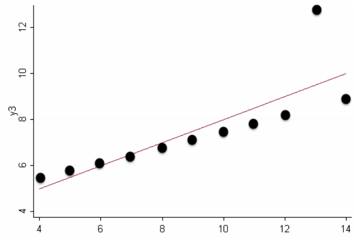
What is the leverage score?

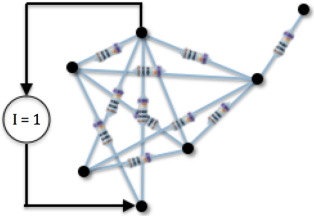
- Statistics: Outlier detection



What is the leverage score?

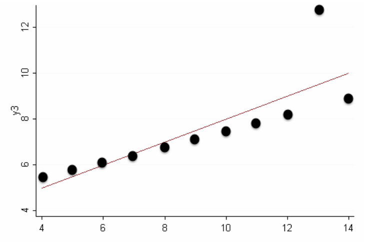
- Statistics: Outlier detection
- Spectral Graph Theory: Effective resistance, commute time

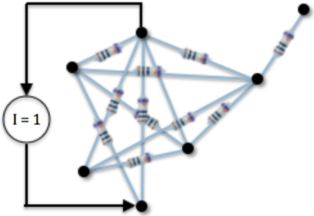




What is the leverage score?

- Statistics: Outlier detection
- Spectral Graph Theory: Effective resistance, commute time
- Matrix Approximation: Row's importance in composing the quadratic form of $\mathbf{A}^\top \mathbf{A}$.





What is the leverage score?

$$\tau_i(\mathbf{A}) = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{a}_i$$

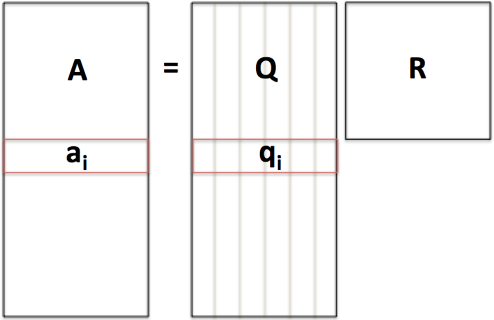
How to Understand this equation?

What is the leverage score?

$$\tau_i(\mathbf{A}) = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{a}_i$$

How to Understand this equation?

- Row norms of orthonormal basis for **A**'s columns

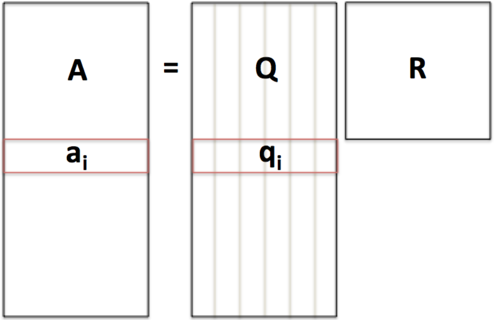


What is the leverage score?

$$\tau_i(\mathbf{A}) = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{a}_i$$

How to Understand this equation?

- Row norms of orthonormal basis for **A**'s columns



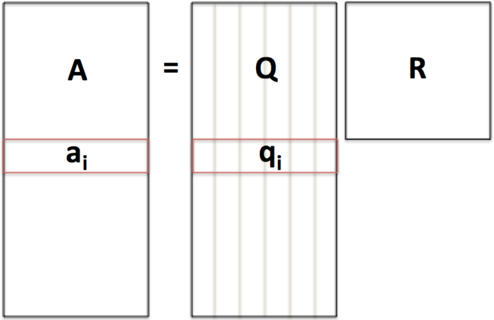
- Correct 'scaling' to make matrix Chernoff bound work

What is the leverage score?

$$\tau_i(\mathbf{A}) = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{a}_i$$

How to Understand this equation?

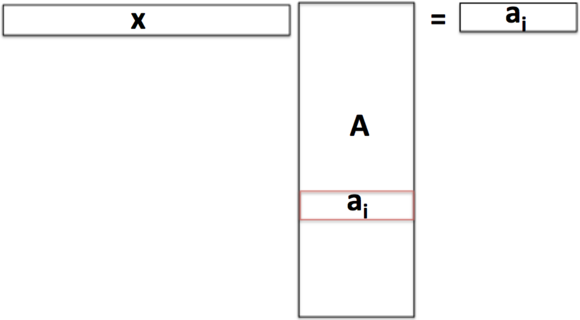
- Row norms of orthonormal basis for **A**'s columns



- Correct 'scaling' to make matrix Chernoff bound work
- **How easily a row can be reconstructed from other rows.**

What is the leverage score?

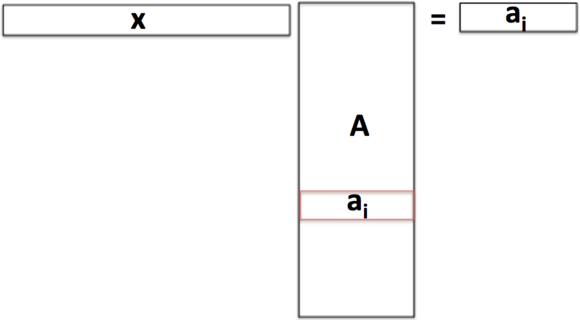
How easily a row can be reconstructed from other rows



- $\min \|\mathbf{x}\|_2^2 = \tau_i(\mathbf{A})$.
- $\mathbf{a}_i$ has component orthogonal to all other rows: $\mathbf{x} = \mathbf{e}_i$, $\|\mathbf{x}\|_2 = 1$.
- There are many rows pointing 'in the direction of' $\mathbf{a}_i$: $\mathbf{x}$ is well spread and has small norm.

What is the leverage score?

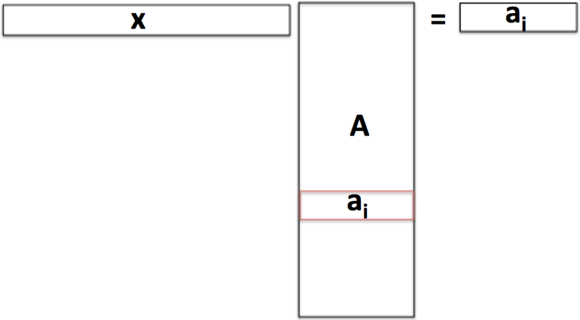
How easily a row can be reconstructed from other rows



- $\min \|\mathbf{x}\|_2^2 = \tau_i(\mathbf{A})$.
- $\mathbf{a}_i$ has component orthogonal to all other rows: $\mathbf{x} = \mathbf{e}_i$, $\|\mathbf{x}\|_2 = 1$.
- There are many rows pointing 'in the direction of' $\mathbf{a}_i$: $\mathbf{x}$ is well spread and has small norm.

What is the leverage score?

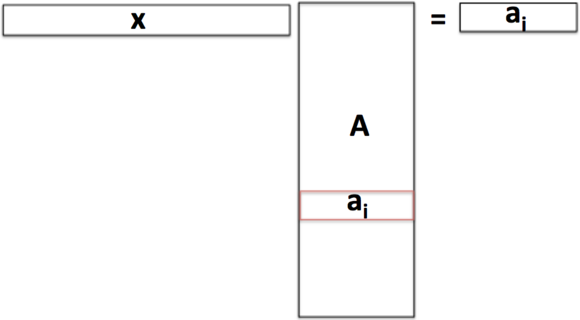
How easily a row can be reconstructed from other rows



- $\min \|\mathbf{x}\|_2^2 = \tau_i(\mathbf{A})$.
- $\mathbf{a}_i$ has component orthogonal to all other rows: $\mathbf{x} = \mathbf{e}_i$, $\|\mathbf{x}\|_2 = 1$.
- There are many rows pointing 'in the direction of' $\mathbf{a}_i$: $\mathbf{x}$ is well spread and has small norm.

What is the leverage score?

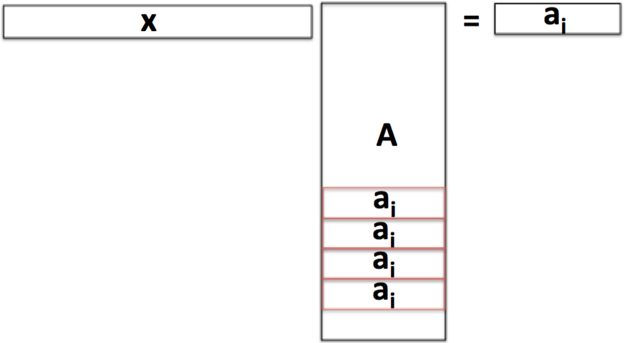
How easily a row can be reconstructed from other rows



- $\min \|\mathbf{x}\|_2^2 = \tau_i(\mathbf{A})$.
- $\mathbf{a}_i$ has component orthogonal to all other rows: $\mathbf{x} = \mathbf{e}_i$, $\|\mathbf{x}\|_2 = 1$.
- There are many rows pointing 'in the direction of' $\mathbf{a}_i$: $\mathbf{x}$ is well spread and has small norm.

What is the leverage score?

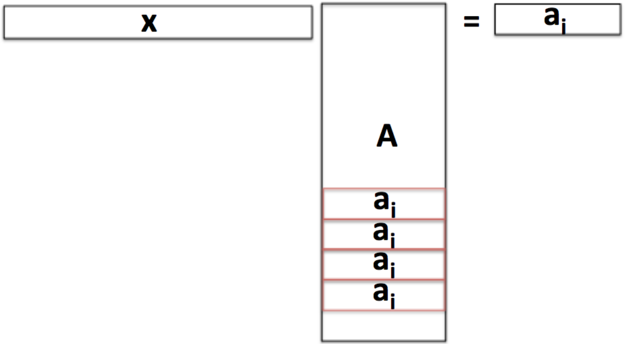
How easily a row can be reconstructed from other rows



- $\bar{\mathbf{x}} = [0, 0, \ldots, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, 0, \ldots]$ works.
- $\tau_i(\mathbf{A}) \leq \|\bar{\mathbf{x}}\|_2^2 = \frac{1}{4}$.

What is the leverage score?

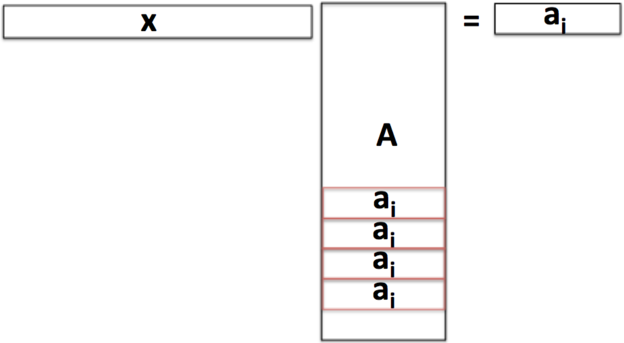
How easily a row can be reconstructed from other rows



- $\bar{\mathbf{x}} = [0, 0, \dots, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, 0, \dots]$ works.
- $\tau_i(\mathbf{A}) \leq \|\bar{\mathbf{x}}\|_2^2 = \frac{1}{4}$.

What is the leverage score?

How easily a row can be reconstructed from other rows



- $\bar{\mathbf{x}} = [0, 0, \ldots, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, 0, \ldots]$ works.
- $\tau_i(\mathbf{A}) \leq \|\bar{\mathbf{x}}\|_2^2 = \frac{1}{4}$.

What is the leverage score?

Mathematically:

$$\min_{\{\mathbf{x} | \mathbf{x}^\top \mathbf{A} = \mathbf{a}_i^\top\}} \|\mathbf{x}\|_2^2$$

To minimize, set

$$\mathbf{x}^\top = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$$

So:

$$\|\mathbf{x}\|_2^2 = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{a}_i = \tau_i(\mathbf{A})$$

What is the leverage score?

Mathematically:

$$\min_{\{\mathbf{x} | \mathbf{x}^\top \mathbf{A} = \mathbf{a}_i^\top\}} \|\mathbf{x}\|_2^2$$

To minimize, set

$$\mathbf{x}^\top = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$$

So:

$$\|\mathbf{x}\|_2^2 = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{a}_i = \tau_i(\mathbf{A})$$

What is the leverage score?

Mathematically:

$$\min_{\{\mathbf{x} | \mathbf{x}^\top \mathbf{A} = \mathbf{a}_i^\top\}} \|\mathbf{x}\|_2^2$$

To minimize, set

$$\mathbf{x}^\top = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$$

So:

$$\|\mathbf{x}\|_2^2 = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{a}_i = \tau_i(\mathbf{A})$$

What is the leverage score?

Mathematically:

$$\min_{\{\mathbf{x} | \mathbf{x}^\top \mathbf{A} = \mathbf{a}_i^\top\}} \|\mathbf{x}\|_2^2$$

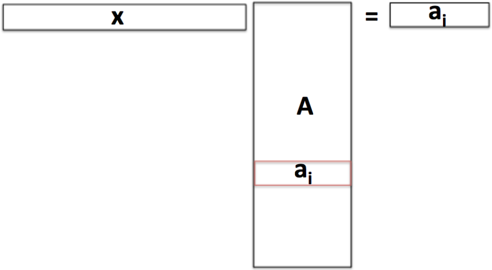
To minimize, set

$$\mathbf{x}^\top = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$$

So:

$$\|\mathbf{x}\|_2^2 = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{a}_i = \tau_i(\mathbf{A})$$

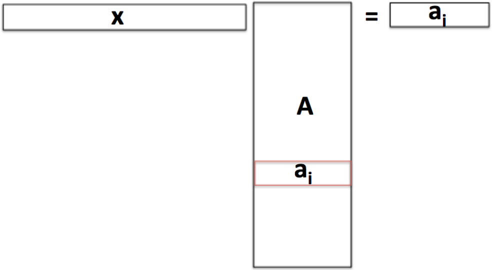
What is the leverage score?



Intepretation tells us:

- $\tau_i(\mathbf{A}) \leq 1$
- Adding rows to $\mathbf{A}$ can only decrease leverage scores of existing rows
- Removing rows can only increase leverage scores
- In fact always have: $\sum_{i=1}^{n} \tau_i(\mathbf{A}) = d$ (Foster's Theorem)

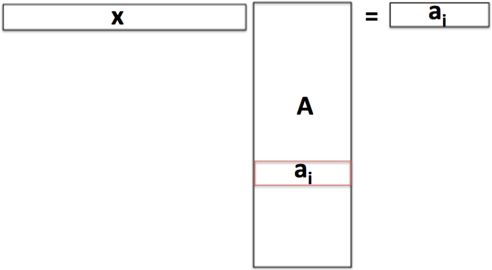
What is the leverage score?



Intepretation tells us:

- $\tau_i(\mathbf{A}) \leq 1$
- Adding rows to $\mathbf{A}$ can only decrease leverage scores of existing rows
- Removing rows can only increase leverage scores
- In fact always have: $\sum_{i=1}^{n} \tau_i(\mathbf{A}) = d$ (Foster's Theorem)

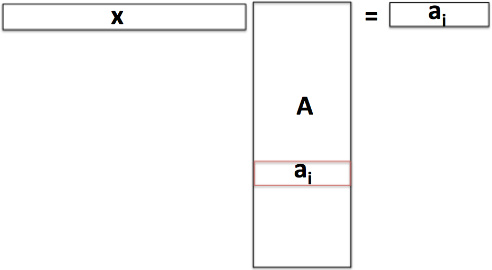
What is the leverage score?



Intepretation tells us:

- $\tau_i(\mathbf{A}) \leq 1$
- Adding rows to **A** can only decrease leverage scores of existing rows
- Removing rows can only increase leverage scores
- In fact always have: $\sum_{i=1}^{n} \tau_i(\mathbf{A}) = d$ (Foster's Theorem)

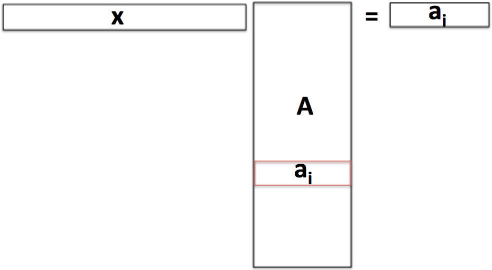
What is the leverage score?



Intepretation tells us:

- $\tau_i(\mathbf{A}) \leq 1$
- Adding rows to **A** can only decrease leverage scores of existing rows
- Removing rows can only increase leverage scores
- In fact always have: $\sum_{i=1}^{n} \tau_i(\mathbf{A}) = d$ (Foster's Theorem)

What is the leverage score?

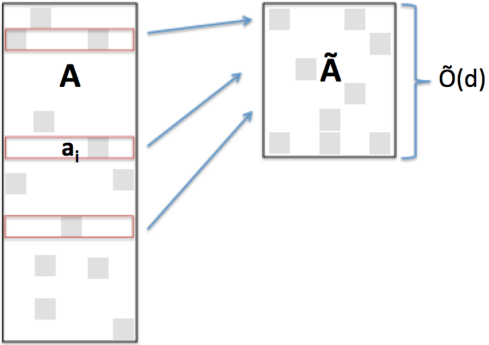


Intepretation tells us:

- $\tau_i(\mathbf{A}) \leq 1$
- Adding rows to **A** can only decrease leverage scores of existing rows
- Removing rows can only increase leverage scores
- In fact always have: $\sum_{i=1}^{n} \tau_i(\mathbf{A}) = d$ (Foster's Theorem)

Computing Leverage Scores

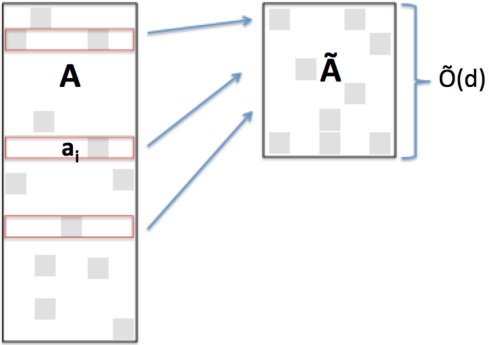
Great! Nicely interpretable scores that allow us to sample down $\mathbf{A}$.



Problem: Computing leverage scores naively requires computing $(\mathbf{A}^\top\mathbf{A})^{-1}$.

Computing Leverage Scores

Great! Nicely interpretable scores that allow us to sample down **A**.



Problem: Computing leverage scores naively requires computing $(\mathbf{A}^\top \mathbf{A})^{-1}$.

Computing Leverage Scores

Traditional Solution

- *Overestimates* are good enough. Just increases number of rows taken.
- Given a constant factor spectral approximation $\tilde{\mathbf{A}}$ we have:

$$\mathbf{a}_i^\top (\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}})^{-1} \mathbf{a}_i \approx_c \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{a}_i$$

- $\sum \tilde{\tau}_i(\mathbf{A}) \leq \sum c \cdot \tau_i(\mathbf{A}) = c \sum \tau_i(\mathbf{A}) = c \cdot d.$
- So can still sample $O(d \log d/\epsilon^2)$ rows using $\tilde{\tau}_i(\mathbf{A})$.

But how to obtain a constant factor spectral approximation?

- Unfortunately, to compute even a constant factor spectral approximation still requires either leverage score estimates, a subspace embedding or a matrix square root.

Computing Leverage Scores

Traditional Solution

- *Overestimates* are good enough. Just increases number of rows taken.
- Given a constant factor spectral approximation $\tilde{\mathbf{A}}$ we have:

$$\mathbf{a}_i^\top (\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}})^{-1} \mathbf{a}_i \approx_c \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{a}_i$$

- $\sum \tilde{\tau}_i(\mathbf{A}) \leq \sum c \cdot \tau_i(\mathbf{A}) = c \sum \tau_i(\mathbf{A}) = c \cdot d$.
- So can still sample $O(d \log d/\epsilon^2)$ rows using $\tilde{\tau}_i(\mathbf{A})$.

But how to obtain a constant factor spectral approximation?

- Unfortunately, to compute even a constant factor spectral approximation still requires either leverage score estimates, a subspace embedding or a matrix square root.

Computing Leverage Scores

Traditional Solution

- *Overestimates* are good enough. Just increases number of rows taken.
- Given a constant factor spectral approximation $\tilde{\mathbf{A}}$ we have:

$$\mathbf{a}_i^\top(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1}\mathbf{a}_i \approx_c \mathbf{a}_i^\top(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{a}_i$$

- $\sum \tilde{\tau}_i(\mathbf{A}) \leq \sum c \cdot \tau_i(\mathbf{A}) = c \sum \tau_i(\mathbf{A}) = c \cdot d.$
- So can still sample $O(d \log d/\epsilon^2)$ rows using $\tilde{\tau}_i(\mathbf{A})$.

But how to obtain a constant factor spectral approximation?

- Unfortunately, to compute even a constant factor spectral approximation still requires either leverage score estimates, a subspace embedding or a matrix square root.

Computing Leverage Scores

Traditional Solution

- *Overestimates* are good enough. Just increases number of rows taken.
- Given a constant factor spectral approximation $\tilde{\mathbf{A}}$ we have:

$$\mathbf{a}_i^\top(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1}\mathbf{a}_i \approx_c \mathbf{a}_i^\top(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{a}_i$$

- $\sum \tilde{\tau}_i(\mathbf{A}) \leq \sum c \cdot \tau_i(\mathbf{A}) = c \sum \tau_i(\mathbf{A}) = c \cdot d.$
- So can still sample $O(d \log d/\epsilon^2)$ rows using $\tilde{\tau}_i(\mathbf{A})$.

But how to obtain a constant factor spectral approximation?

- Unfortunately, to compute even a constant factor spectral approximation still requires either leverage score estimates, a subspace embedding or a matrix square root.

Computing Leverage Scores

Traditional Solution

- *Overestimates* are good enough. Just increases number of rows taken.
- Given a constant factor spectral approximation $\tilde{\mathbf{A}}$ we have:

$$\mathbf{a}_i^\top(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1}\mathbf{a}_i \approx_c \mathbf{a}_i^\top(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{a}_i$$

- $\sum \tilde{\tau}_i(\mathbf{A}) \leq \sum c \cdot \tau_i(\mathbf{A}) = c\sum \tau_i(\mathbf{A}) = c \cdot d$.
- So can still sample $O(d \log d/\epsilon^2)$ rows using $\tilde{\tau}_i(\mathbf{A})$.

But how to obtain a constant factor spectral approximation?

- Unfortunately, to compute even a constant factor spectral approximation still requires either leverage score estimates, a subspace embedding or a matrix square root.

Computing Leverage Scores

Traditional Solution

- *Overestimates* are good enough. Just increases number of rows taken.
- Given a constant factor spectral approximation $\tilde{\mathbf{A}}$ we have:

$$\mathbf{a}_i^\top (\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}})^{-1} \mathbf{a}_i \approx_c \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{a}_i$$

- $\sum \tilde{\tau}_i(\mathbf{A}) \leq \sum c \cdot \tau_i(\mathbf{A}) = c \sum \tau_i(\mathbf{A}) = c \cdot d$.
- So can still sample $O(d \log d/\epsilon^2)$ rows using $\tilde{\tau}_i(\mathbf{A})$.

But how to obtain a constant factor spectral approximation?

- Unfortunately, to compute even a constant factor spectral approximation still requires either leverage score estimates, a subspace embedding or a matrix square root.

Computing Leverage Scores

Traditional Solution

- *Overestimates* are good enough. Just increases number of rows taken.
- Given a constant factor spectral approximation $\tilde{\mathbf{A}}$ we have:

$$\mathbf{a}_i^\top (\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}})^{-1} \mathbf{a}_i \approx_c \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{a}_i$$

- $\sum \tilde{\tau}_i(\mathbf{A}) \leq \sum c \cdot \tau_i(\mathbf{A}) = c \sum \tau_i(\mathbf{A}) = c \cdot d$.
- So can still sample $O(d \log d/\epsilon^2)$ rows using $\tilde{\tau}_i(\mathbf{A})$.

But how to obtain a constant factor spectral approximation?

- Unfortunately, to compute even a constant factor spectral approximation still requires either leverage score estimates, a subspace embedding or a matrix square root.

Computing Leverage Scores

- A spectral approximation $\tilde{\mathbf{A}}$ gives the *for each* guarantee:

$$\tilde{\tau}_i(\mathbf{A}) \leq c \cdot \tau_i(\mathbf{A})$$

- This is stronger than we need though! We just need a bound on the sum:

$$\sum_i \tilde{\tau}_i(\mathbf{A}) \leq c \cdot d$$

Computing Leverage Scores

- A spectral approximation $\tilde{\mathbf{A}}$ gives the *for each* guarantee:

$$\tilde{\tau}_i(\mathbf{A}) \leq c \cdot \tau_i(\mathbf{A})$$

- This is stronger than we need though! We just need a bound on the sum:

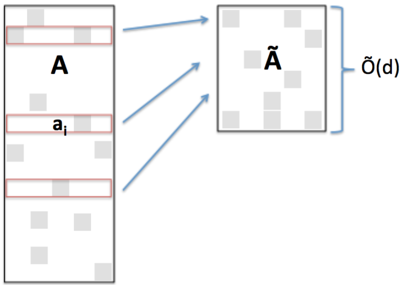
$$\sum_i \tilde{\tau}_i(\mathbf{A}) \leq c \cdot d$$

Overview

Iterative Leverage Score Computation

Review of what our goal is:

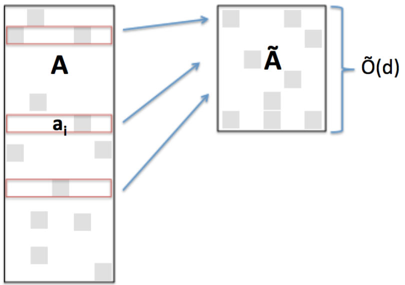
- Want to find $\tilde{\mathbf{A}}$ such that $\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 \approx_\epsilon \|\mathbf{A}\mathbf{x}\|_2^2$.
- Want to avoid JL Projection - just use random row sampling
- Need a way to efficiently compute approximations $\tilde{\tau}_i(\mathbf{A})$ such that $\sum \tilde{\tau}_i(\mathbf{A}) = O(d)$
- Efficient: $\tilde{O}(nnz(\mathbf{A}) + R(d,d))$ where $R(d,d)$ is the cost of solving a $d \times d$ regression problem.



Iterative Leverage Score Computation

Review of what our goal is:

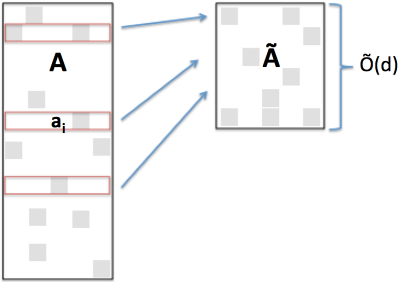
- Want to find $\tilde{\mathbf{A}}$ such that $\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 \approx_\epsilon \|\mathbf{A}\mathbf{x}\|_2^2$.
- Want to avoid JL Projection - just use random row sampling
- Need a way to efficiently compute approximations $\tilde{\tau}_i(\mathbf{A})$ such that $\sum \tilde{\tau}_i(\mathbf{A}) = O(d)$
- Efficient: $\tilde{O}(nnz(\mathbf{A}) + R(d,d))$ where $R(d,d)$ is the cost of solving a $d \times d$ regression problem.



Iterative Leverage Score Computation

Review of what our goal is:

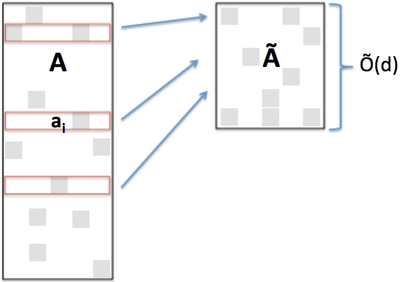
- Want to find $\tilde{\mathbf{A}}$ such that $\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 \approx_\epsilon \|\mathbf{A}\mathbf{x}\|_2^2$.
- Want to avoid JL Projection - just use random row sampling
- Need a way to efficiently compute approximations $\tilde{\tau}_i(\mathbf{A})$ such that $\sum \tilde{\tau}_i(\mathbf{A}) = O(d)$
- Efficient: $\tilde{O}(nnz(\mathbf{A}) + R(d,d))$ where $R(d,d)$ is the cost of solving a $d \times d$ regression problem.



Iterative Leverage Score Computation

Review of what our goal is:

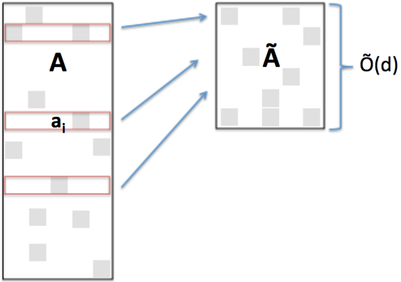
- Want to find $\tilde{\mathbf{A}}$ such that $\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 \approx_\epsilon \|\mathbf{A}\mathbf{x}\|_2^2$.
- Want to avoid JL Projection - just use random row sampling
- Need a way to efficiently compute approximations $\tilde{\tau}_i(\mathbf{A})$ such that $\sum \tilde{\tau}_i(\mathbf{A}) = O(d)$
- Efficient: $\tilde{O}(nnz(\mathbf{A}) + R(d,d))$ where $R(d,d)$ is the cost of solving a $d \times d$ regression problem.



Iterative Leverage Score Computation

Review of what our goal is:

- Want to find $\tilde{\mathbf{A}}$ such that $\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 \approx_\epsilon \|\mathbf{A}\mathbf{x}\|_2^2$.
- Want to avoid JL Projection - just use random row sampling
- Need a way to efficiently compute approximations $\tilde{\tau}_i(\mathbf{A})$ such that $\sum \tilde{\tau}_i(\mathbf{A}) = O(d)$
- Efficient: $\tilde{O}(nnz(\mathbf{A}) + R(d,d))$ where $R(d,d)$ is the cost of solving a $d \times d$ regression problem.



Uniform Row Sampling

- In practice, data is typically *incoherent* - no row has a high leverage score. [Kumar, Mohri, Talwalkar '12], [Avron, Maymounkov, Toledo '10].

$$\forall i \quad \tau_i(\mathbf{A}) \leq O(d/n)$$

.

Uniform Row Sampling

- In practice, data is typically *incoherent* - no row has a high leverage score. [Kumar, Mohri, Talwalkar '12], [Avron, Maymounkov, Toledo '10].

$$\forall i \quad \tau_i(\mathbf{A}) \leq O(d/n)$$

.

- In this case, we can *uniformly sample* $O(d \log d)$ rows to obtain a spectral approximation.

Uniform Row Sampling

- In practice, data is typically *incoherent* - no row has a high leverage score. [Kumar, Mohri, Talwalkar '12], [Avron, Maymounkov, Toledo '10].

$$\forall i \quad \tau_i(\mathbf{A}) \leq O(d/n)$$

.

- In this case, we can *uniformly sample* $O(d \log d)$ rows to obtain a spectral approximation.

$$\sum_i \tilde{\tau}_i(\mathbf{A}) = O\left(\frac{d}{n}\right) \cdot n = O(d)$$

Uniform Row Sampling

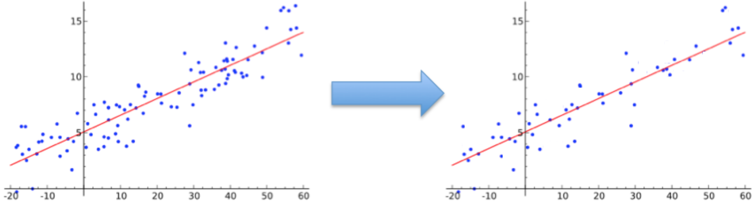
- In practice, data is typically *incoherent* - no row has a high leverage score. [Kumar, Mohri, Talwalkar '12], [Avron, Maymounkov, Toledo '10].

$$\forall i \quad \tau_i(\mathbf{A}) \leq O(d/n)$$

.

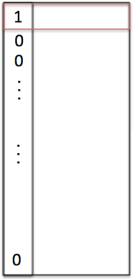
- In this case, we can *uniformly sample* $O(d \log d)$ rows to obtain a spectral approximation.

$$\sum_i \tilde{\tau}_i(\mathbf{A}) = O\left(\frac{d}{n}\right) \cdot n = O(d)$$



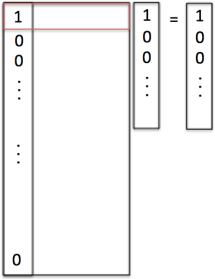
Uniform Row Sampling

No guarantees on what uniform sampling does on general matrices.



Uniform Row Sampling

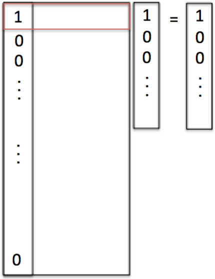
No guarantees on what uniform sampling does on general matrices.



- For $x = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, $\|\mathbf{Ax}\|_2^2 = 1$, but with good probability, $\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 = 0$.

Uniform Row Sampling

No guarantees on what uniform sampling does on general matrices.



- For $x = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, $\|\mathbf{A}\mathbf{x}\|_2^2 = 1$, but with good probability, $\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 = 0$.
- **Can we still use uniform sampling in some way?**

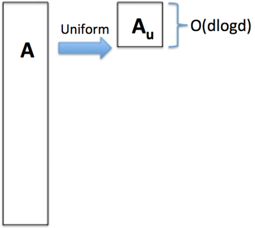
Uniform Row Sampling

One simple idea for an algorithm:

Uniform Row Sampling

One simple idea for an algorithm:

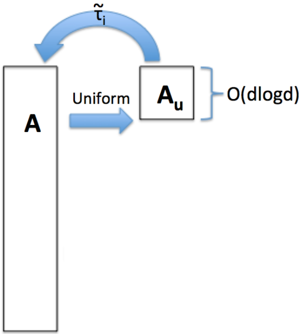
1. Uniformly sample $O(d \log d)$ rows of $\mathbf{A}$ to obtain $\mathbf{A}_u$



Uniform Row Sampling

One simple idea for an algorithm:

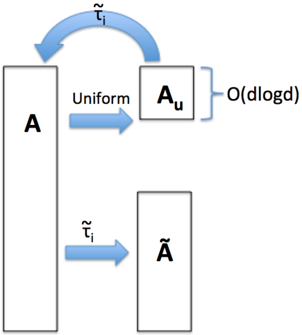
1. Uniformly sample $O(d \log d)$ rows of $\mathbf{A}$ to obtain $\mathbf{A}_u$
2. Compute $(\mathbf{A}_u^\top \mathbf{A}_u)^{-1}$, and use it to estimate leverage scores of $\mathbf{A}$



Uniform Row Sampling

One simple idea for an algorithm:

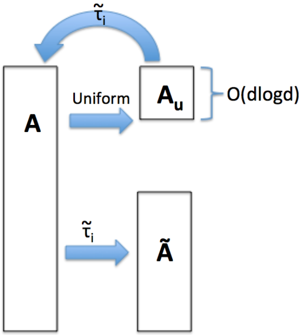
1. Uniformly sample $O(d \log d)$ rows of $\mathbf{A}$ to obtain $\mathbf{A}_u$
2. Compute $(\mathbf{A}_u^\top \mathbf{A}_u)^{-1}$, and use it to estimate leverage scores of $\mathbf{A}$
3. Sample rows of $\mathbf{A}$ using these estimated leverage scores to obtain $\tilde{\mathbf{A}}$ which spectrally approximates $\mathbf{A}$.



Uniform Row Sampling

One simple idea for an algorithm:

1. Uniformly sample $O(d \log d)$ rows of $\mathbf{A}$ to obtain $\mathbf{A}_u$
2. Compute $(\mathbf{A}_u^\top \mathbf{A}_u)^{-1}$, and use it to estimate leverage scores of $\mathbf{A}$
3. Sample rows of $\mathbf{A}$ using these estimated leverage scores to obtain $\tilde{\mathbf{A}}$ which spectrally approximates $\mathbf{A}$.



We want to bound how large $\tilde{\mathbf{A}}$ must be.

Uniform Row Sampling

Theorem

Let $\mathbf{A}_u$ *be obtained from uniformly sampling* m *rows of* $\mathbf{A}$. *Let* $\mathbf{A}_{u \cup i}$ *be* $\mathbf{A}_u$ *with* $\mathbf{a}_i$ *appended if not already included.* $\tilde{\tau}_i(\mathbf{A}) = \mathbf{a}_i^\top (\mathbf{A}_{u \cup i}^\top \mathbf{A}_{u \cup i})^{-1} \mathbf{a}_i$.

$$\tilde{\tau}_i(\mathbf{A}) \geq \tau_i(\mathbf{A})$$

$$\mathbb{E} \sum_i \tilde{\tau}_i(\mathbf{A}) \leq \frac{nd}{m}$$

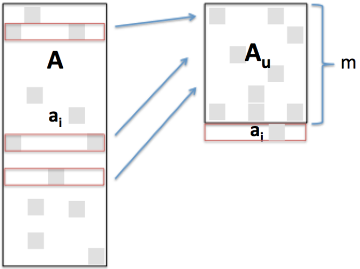
Uniform Row Sampling

Theorem

Let $\mathbf{A}_u$ *be obtained from uniformly sampling* m *rows of* $\mathbf{A}$*. Let* $\mathbf{A}_{u\cup i}$ *be* $\mathbf{A}_u$ *with* $\mathbf{a}_i$ *appended if not already included.* $\tilde{\tau}_i(\mathbf{A}) = \mathbf{a}_i^\top(\mathbf{A}_{u\cup i}^\top \mathbf{A}_{u\cup i})^{-1}\mathbf{a}_i$.

$$\tilde{\tau}_i(\mathbf{A}) \geq \tau_i(\mathbf{A})$$

$$\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) \leq \frac{nd}{m}$$



Uniform Row Sampling

Theorem

Let $\mathbf{A}_u$ be obtained from uniformly sampling m rows of $\mathbf{A}$. Let $\mathbf{A}_{u\cup i}$ be $\mathbf{A}_u$ with $\mathbf{a}_i$ appended if not already included. $\tilde{\tau}_i(\mathbf{A}) = \mathbf{a}_i^\top(\mathbf{A}_{u\cup i}^\top \mathbf{A}_{u\cup i})^{-1}\mathbf{a}_i$.

$$\tilde{\tau}_i(\mathbf{A}) \geq \tau_i(\mathbf{A})$$

$$\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) \leq \frac{nd}{m}$$

Note:

Sherman-Morrison gives equation to compute $\tilde{\tau}_i(\mathbf{A})$ from $\mathbf{a}_i^\top(\mathbf{A}_u^\top \mathbf{A}_u)^{-1}\mathbf{a}_i$

$$\tilde{\tau}_i(\mathbf{A}) = \begin{cases} \mathbf{a}_i^\top(\mathbf{A}_u^\top \mathbf{A}_u)^{-1}\mathbf{a}_i & \text{if } \mathbf{a}_i \in \mathbf{A}_u \\ \frac{1}{1+\frac{1}{\mathbf{a}_i^\top(\mathbf{A}_u^\top \mathbf{A}_u)^{-1}\mathbf{a}_i}} & \text{o.w.} \end{cases}$$

Uniform Row Sampling

What does this bound give us?

$$\mathbb{E} \sum_i \tilde{\tau}_i(\mathbf{A}) \leq \frac{nd}{m}$$

Uniform Row Sampling

What does this bound give us?

$$\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) \leq \frac{nd}{m}$$

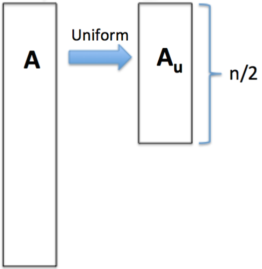
Set $m = \frac{n}{2}$. Then: $\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) \leq 2d$.

Uniform Row Sampling

What does this bound give us?

$$\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) \leq \frac{nd}{m}$$

Set $m = \frac{n}{2}$. Then: $\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) \leq 2d$.

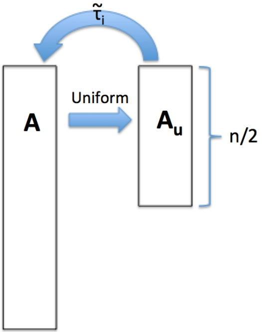


Uniform Row Sampling

What does this bound give us?

$$\mathbb{E} \sum_i \tilde{\tau}_i(\mathbf{A}) \leq \frac{nd}{m}$$

Set $m = \frac{n}{2}$. Then: $\mathbb{E} \sum_i \tilde{\tau}_i(\mathbf{A}) \leq 2d$.

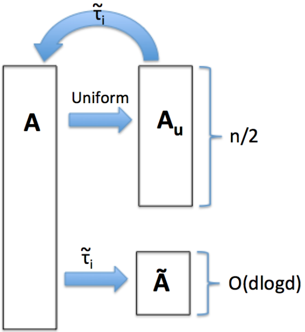


Uniform Row Sampling

What does this bound give us?

$$\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) \leq \frac{nd}{m}$$

Set $m = \frac{n}{2}$. Then: $\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) \leq 2d$.

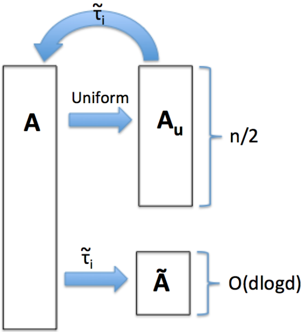


Uniform Row Sampling

What does this bound give us?

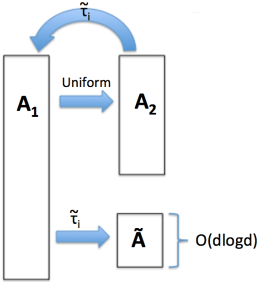
$$\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) \leq \frac{nd}{m}$$

Set $m = \frac{n}{2}$. Then: $\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) \leq 2d$.



- Reminiscent of the MST algorithm from [Karger, Klein, Tarjan '95].

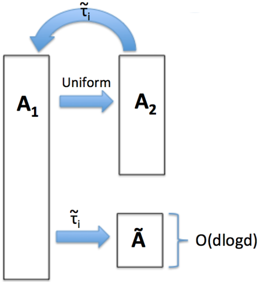
Iterative Leverage Score Computation



Immediately yields a recursive algorithm for obtaining $\tilde{\mathbf{A}}$.

1. Recursively obtain constant factor spectral approximation to $\mathbf{A}_2$.
2. Use this to approximate $\tilde{\tau}_i(\mathbf{A})$.
3. Use these approximate scores to obtain $\tilde{\mathbf{A}}$. (With any $(1 \pm \epsilon)$ multiplicative error you want).

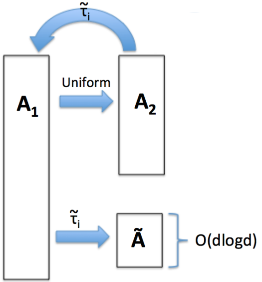
Iterative Leverage Score Computation



Immediately yields a recursive algorithm for obtaining $\tilde{\mathbf{A}}$.

1. Recursively obtain constant factor spectral approximation to $\mathbf{A}_2$.
2. Use this to approximate $\tilde{\tau}_i(\mathbf{A})$.
3. Use these approximate scores to obtain $\tilde{\mathbf{A}}$. (With any $(1 \pm \epsilon)$ multiplicative error you want).

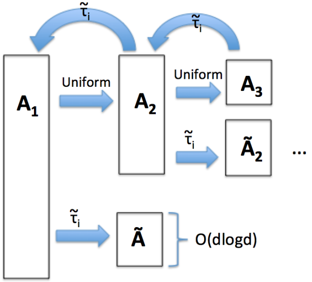
Iterative Leverage Score Computation



Immediately yields a recursive algorithm for obtaining $\tilde{\mathbf{A}}$.

1. Recursively obtain constant factor spectral approximation to $\mathbf{A}_2$.
2. Use this to approximate $\tilde{\tau}_i(\mathbf{A})$.
3. Use these approximate scores to obtain $\tilde{\mathbf{A}}$. (With any $(1 \pm \epsilon)$ multiplicative error you want).

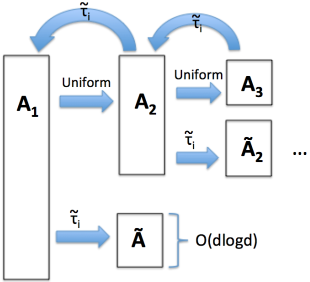
Iterative Leverage Score Computation



Immediately yields a recursive algorithm for obtaining $\tilde{\mathbf{A}}$.

1. Recursively obtain constant factor spectral approximation to $\mathbf{A}_2$.
2. Use this to approximate $\tilde{\tau}_i(\mathbf{A})$.
3. Use these approximate scores to obtain $\tilde{\mathbf{A}}$. (With any $(1 \pm \epsilon)$ multiplicative error you want).

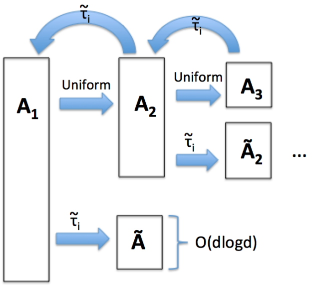
Iterative Leverage Score Computation



Immediately yields a recursive algorithm for obtaining $\tilde{\mathbf{A}}$.

1. Recursively obtain constant factor spectral approximation to $\mathbf{A}_2$.
2. Use this to approximate $\tilde{\tau}_i(\mathbf{A})$.
3. Use these approximate scores to obtain $\tilde{\mathbf{A}}$. (With any $(1 \pm \epsilon)$ multiplicative error you want).

Iterative Leverage Score Computation



Immediately yields a recursive algorithm for obtaining $\tilde{\mathbf{A}}$.

1. Recursively obtain constant factor spectral approximation to $\mathbf{A}_2$.
2. Use this to approximate $\tilde{\tau}_i(\mathbf{A})$.
3. Use these approximate scores to obtain $\tilde{\mathbf{A}}$. (With any $(1 \pm \epsilon)$ multiplicative error you want).

Iterative Leverage Score Computation

This algorithm, along with a few other variations on it are the main algorithmic results of our paper.

- Obtain spectral approximation to $\mathbf{A}$ with arbitrary error.
- Avoid JL projecting and densifying $\mathbf{A}$.
- $\tilde{O}(nnz(\mathbf{A}) + R(d, d))$ time.

Iterative Leverage Score Computation

This algorithm, along with a few other variations on it are the main algorithmic results of our paper.

- Obtain spectral approximation to $\mathbf{A}$ with arbitrary error.
- Avoid JL projecting and densifying $\mathbf{A}$.
- $\tilde{O}(nnz(\mathbf{A}) + R(d,d))$ time.

Iterative Leverage Score Computation

This algorithm, along with a few other variations on it are the main algorithmic results of our paper.

- Obtain spectral approximation to $\mathbf{A}$ with arbitrary error.
- Avoid JL projecting and densifying $\mathbf{A}$.
- $\tilde{O}(nnz(\mathbf{A}) + R(d, d))$ time.

Iterative Leverage Score Computation

This algorithm, along with a few other variations on it are the main algorithmic results of our paper.

- Obtain spectral approximation to $\mathbf{A}$ with arbitrary error.
- Avoid JL projecting and densifying $\mathbf{A}$.
- $\tilde{O}(nnz(\mathbf{A}) + R(d, d))$ time.

Proof of Main Theorem

Theorem

Let $\mathbf{A}_u$ be obtained from uniformly sampling m rows of $\mathbf{A}$. Let $\mathbf{A}_{u\cup i}$ be $\mathbf{A}_u$ with $\mathbf{a}_i$ appended if not already included. $\tilde{\tau}_i(\mathbf{A}) = \mathbf{a}_i^\top(\mathbf{A}_{u\cup i}^\top \mathbf{A}_{u\cup i})^{-1}\mathbf{a}_i$.

$$\tilde{\tau}_i(\mathbf{A}) \geq \tau_i(\mathbf{A}) \tag{1}$$

$$\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) \leq \frac{nd}{m} \tag{2}$$

Proof of Main Theorem

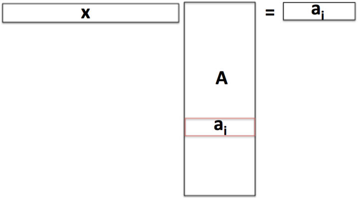
Theorem

Let $\mathbf{A}_u$ be obtained from uniformly sampling m rows of $\mathbf{A}$. Let $\mathbf{A}_{u\cup i}$ be $\mathbf{A}_u$ with $\mathbf{a}_i$ appended if not already included. $\tilde{\tau}_i(\mathbf{A}) = \mathbf{a}_i^\top(\mathbf{A}_{u\cup i}^\top \mathbf{A}_{u\cup i})^{-1}\mathbf{a}_i$.

$$\tilde{\tau}_i(\mathbf{A}) \geq \tau_i(\mathbf{A}) \tag{1}$$

$$\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) \leq \frac{nd}{m} \tag{2}$$

- (1) follows from the fact that removing rows of $\mathbf{A}$ can only increase leverage scores.



Proof of Main Theorem

$$\mathbb{E}_u \sum_i \tilde{\tau}_i(\mathbf{A}) \leq \frac{nd}{m}$$

Proof of Main Theorem

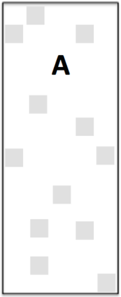
$$\mathop{\mathbb{E}}_{u} \sum_{i} \tilde{\tau}_i(\mathbf{A}) \leq \frac{nd}{m}$$

- Consider choosing a uniform random row $\mathbf{a}_j$.

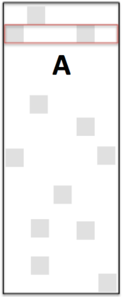
$$\mathop{\mathbb{E}}_{u} \sum_{i} \tilde{\tau}_i(\mathbf{A}) = n \cdot \mathop{\mathbb{E}}_{u,j} \tilde{\tau}_j(\mathbf{A})$$

.

Proof of Main Theorem

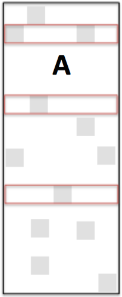


Proof of Main Theorem

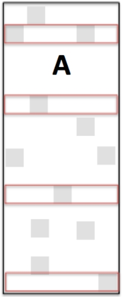


Proof of Main Theorem

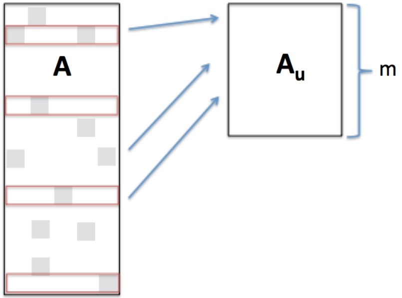
Proof of Main Theorem



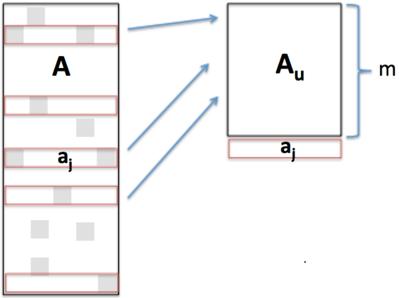
Proof of Main Theorem



Proof of Main Theorem

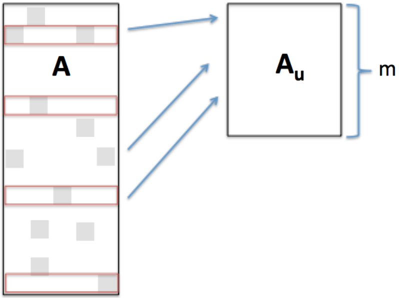


Proof of Main Theorem



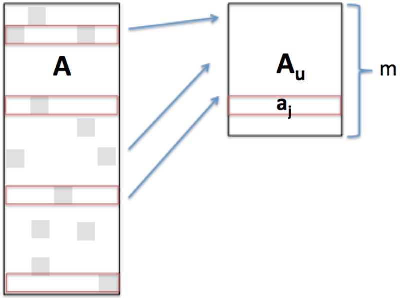
- Further, expectation of this process is *exactly* the same as expected leverage score if we sample m rows, and then compute the leverage score of a random one.

Proof of Main Theorem



- Further, expectation of this process is *exactly* the same as expected leverage score if we sample m rows, and then compute the leverage score of a random one.

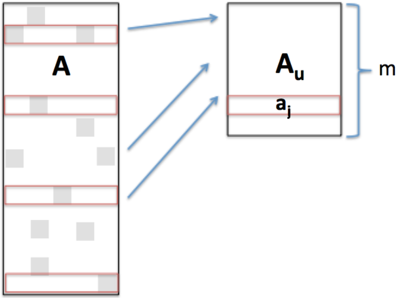
Proof of Main Theorem



- Further, expectation of this process is *exactly* the same as expected leverage score if we sample m rows, and then compute the leverage score of a random one.

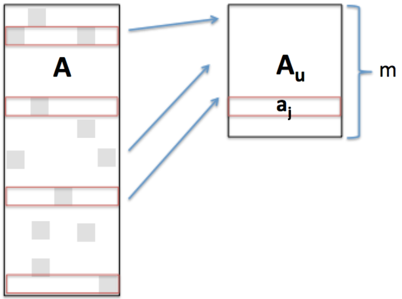
Proof of Main Theorem

- Remember, $\sum_i \tau_i(\mathbf{A}_u) = d$.
- m rows, so expected score of randomly chosen row is $\frac{d}{m}$.
- $\mathbb{E}_j \tilde{\tau}_j(\mathbf{A}) = \frac{d}{m}$
- $\mathbb{E}_u \sum_i \tilde{\tau}_i(\mathbf{A}) = \frac{nd}{m}$.



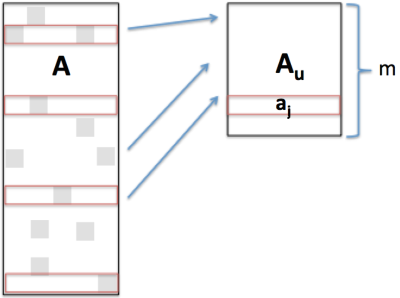
Proof of Main Theorem

- Remember, $\sum_i \tau_i(\mathbf{A}_u) = d$.
- m rows, so expected score of randomly chosen row is $\frac{d}{m}$.
- $\mathbb{E}_j \, \tilde{\tau}_j(\mathbf{A}) = \frac{d}{m}$
- $\mathbb{E}_u \sum_i \tilde{\tau}_i(\mathbf{A}) = \frac{nd}{m}$.



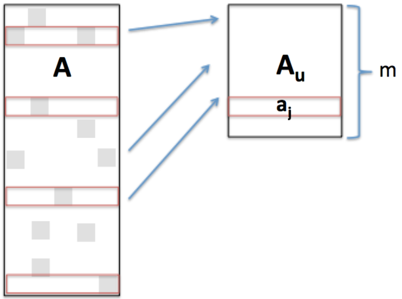
Proof of Main Theorem

- Remember, $\sum_i \tau_i(\mathbf{A}_u) = d$.
- m rows, so expected score of randomly chosen row is $\frac{d}{m}$.
- $\mathbb{E}_j \tilde{\tau}_j(\mathbf{A}) = \frac{d}{m}$
- $\mathbb{E}_u \sum_i \tilde{\tau}_i(\mathbf{A}) = \frac{nd}{m}$.



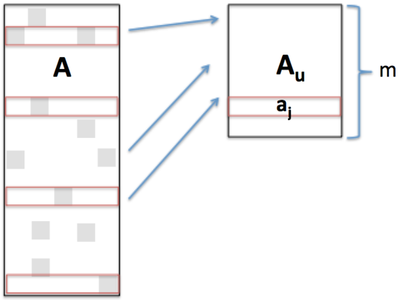
Proof of Main Theorem

- Remember, $\sum_i \tau_i(\mathbf{A}_u) = d$.
- m rows, so expected score of randomly chosen row is $\frac{d}{m}$.
- $\mathbb{E}_j \, \tilde{\tau}_j(\mathbf{A}) = \frac{d}{m}$
- $\mathbb{E}_u \sum_i \tilde{\tau}_i(\mathbf{A}) = \frac{nd}{m}$.



Proof of Main Theorem

- Remember, $\sum_i \tau_i(\mathbf{A}_u) = d$.
- m rows, so expected score of randomly chosen row is $\frac{d}{m}$.
- $\mathbb{E}_j \, \tilde{\tau}_j(\mathbf{A}) = \frac{d}{m}$
- $\mathbb{E}_u \sum_i \tilde{\tau}_i(\mathbf{A}) = \frac{nd}{m}$.



Review

What did this theorem just tell us?

- $\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) = \mathbb{E}\sum_i \mathbf{a}_i^\top(\mathbf{A}_{u\cup i}^\top \mathbf{A}_{u\cup i})^{-1}\mathbf{a}_i$ is bounded. And this is all we need!
- Recall that uniform sampling from $\mathbf{A}$ does *not* give us a spectral approximation.
- We cannot bound $\mathbf{a}_i^\top(\mathbf{A}_u^\top \mathbf{A}_u)^{-1}\mathbf{a}_i$.

Review

What did this theorem just tell us?

- $\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) = \mathbb{E}\sum_i \mathbf{a}_i^\top(\mathbf{A}_{u\cup i}^\top \mathbf{A}_{u\cup i})^{-1}\mathbf{a}_i$ is bounded. And this is all we need!
- Recall that uniform sampling from $\mathbf{A}$ does *not* give us a spectral approximation.
- We cannot bound $\mathbf{a}_i^\top(\mathbf{A}_u^\top \mathbf{A}_u)^{-1}\mathbf{a}_i$.

Review

What did this theorem just tell us?

- $\mathbb{E}\sum_i \tilde{\tau}_i(\mathbf{A}) = \mathbb{E}\sum_i \mathbf{a}_i^\top(\mathbf{A}_{u\cup i}^\top \mathbf{A}_{u\cup i})^{-1}\mathbf{a}_i$ is bounded. And this is all we need!
- Recall that uniform sampling from $\mathbf{A}$ does *not* give us a spectral approximation.
- We cannot bound $\mathbf{a}_i^\top(\mathbf{A}_u^\top \mathbf{A}_u)^{-1}\mathbf{a}_i$.

Overview

Coherence Reducing Reweighting

Reminder: If our data is *incoherent*, then all leverage scores are small $O(d/n)$ and we can uniformly sample rows and obtain a small spectral approximation.

Coherence Reducing Reweighting

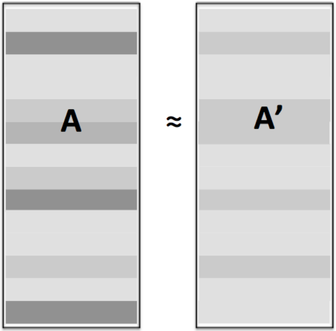
Reminder: If our data is *incoherent*, then all leverage scores are small $O(d/n)$ and we can uniformly sample rows and obtain a small spectral approximation.

We show: Even if our matrix is not incoherent, it is 'close' to some incoherent matrix.

Coherence Reducing Reweighting

Reminder: If our data is *incoherent*, then all leverage scores are small $O(d/n)$ and we can uniformly sample rows and obtain a small spectral approximation.

We show: Even if our matrix is not incoherent, it is 'close' to some incoherent matrix.



Coherence Reducing Reweighting

What do we mean by close?

Coherence Reducing Reweighting

What do we mean by close?

- We can reweight d/α rows of $\mathbf{A}$ to obtain $\mathbf{A}'$ with $\tau_i(\mathbf{A}') \leq \alpha$ for all i.

Coherence Reducing Reweighting

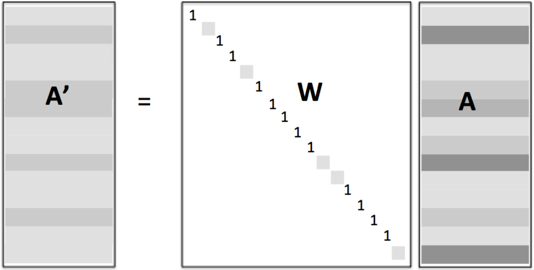
What do we mean by close?

- We can reweight d/α rows of $\mathbf{A}$ to obtain $\mathbf{A}'$ with $\tau_i(\mathbf{A}') \leq \alpha$ for all i.
- Can reweight $n/2$ rows so that τ_i is bounded by $2d/n$

Coherence Reducing Reweighting

What do we mean by close?

- We can reweight d/α rows of $\mathbf{A}$ to obtain $\mathbf{A}'$ with $\tau_i(\mathbf{A}') \leq \alpha$ for all i.
- Can reweight $n/2$ rows so that τ_i is bounded by $2d/n$



Coherence Reducing Reweighting

What does this result give us (without going into details)?

Coherence Reducing Reweighting

What does this result give us (without going into details)?

- Uniform sample m rows of $\mathbf{A}$ and set $\tilde{\tau}_i(\mathbf{A}) = \max\{\mathbf{a}_i^\top (\mathbf{A}_u^\top \mathbf{A}_u)^{-1} \mathbf{a}_i, 1\}$.

Coherence Reducing Reweighting

What does this result give us (without going into details)?

- Uniform sample m rows of $\mathbf{A}$ and set $\tilde{\tau}_i(\mathbf{A}) = \max\{\mathbf{a}_i^\top(\mathbf{A}_u^\top \mathbf{A}_u)^{-1}\mathbf{a}_i, 1\}$.
- This sampling does a good job of approximating $\mathbf{A}'$.

Coherence Reducing Reweighting

What does this result give us (without going into details)?

- Uniform sample m rows of $\mathbf{A}$ and set $\tilde{\tau}_i(\mathbf{A}) = \max\{\mathbf{a}_i^\top (\mathbf{A}_u^\top \mathbf{A}_u)^{-1} \mathbf{a}_i, 1\}$.
- This sampling does a good job of approximating $\mathbf{A}'$.
- Gives good estimates for leverage scores of the that are not reweighted.

Coherence Reducing Reweighting

What does this result give us (without going into details)?

- Uniform sample m rows of $\mathbf{A}$ and set $\tilde{\tau}_i(\mathbf{A}) = \max\{\mathbf{a}_i^\top(\mathbf{A}_u^\top\mathbf{A}_u)^{-1}\mathbf{a}_i, 1\}$.
- This sampling does a good job of approximating $\mathbf{A}'$.
- Gives good estimates for leverage scores of the that are not reweighted.
- Few rows are reweighted, so overall we are able to significantly reduce matrix size.

Coherence Reducing Reweighting

What does this result give us (without going into details)?

- Uniform sample m rows of $\mathbf{A}$ and set $\tilde{\tau}_i(\mathbf{A}) = \max\{\mathbf{a}_i^\top (\mathbf{A}_u^\top \mathbf{A}_u)^{-1} \mathbf{a}_i, 1\}$.
- This sampling does a good job of approximating $\mathbf{A}'$.
- Gives good estimates for leverage scores of the that are not reweighted.
- Few rows are reweighted, so overall we are able to significantly reduce matrix size.
- Don't need to actually compute $\mathbf{W}$. Just existence is enough.

How to prove existence of reweighting?

Coherence Reducing Reweighting

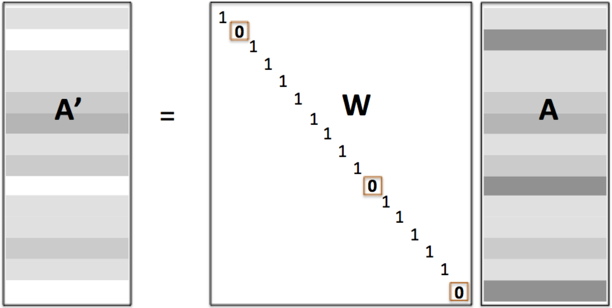
How to prove existence of reweighting?

- $\sum_i \tau_i(\mathbf{A}) = d$ so at most d/α rows with $\tau_i(\mathbf{A}) \geq \alpha$.

Coherence Reducing Reweighting

How to prove existence of reweighting?

- $\sum_i \tau_i(\mathbf{A}) = d$ so at most d/α rows with $\tau_i(\mathbf{A}) \geq \alpha$.
- Can we just delete them?

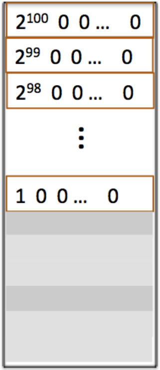


Coherence Reducing Reweighting

Not quite. Deleting some rows may cause leverage scores of other rows to increase.

Coherence Reducing Reweighting

Not quite. Deleting some rows may cause leverage scores of other rows to increase.



Alternative idea

Coherence Reducing Reweighting

Alternative idea

1. Cycle through rows

Coherence Reducing Reweighting

Alternative idea

1. Cycle through rows
2. Each time you see a row with $\tau_i(\mathbf{A}) \geq \alpha$ cut its weight so that $\tau_i(\mathbf{A}) = \alpha$

Coherence Reducing Reweighting

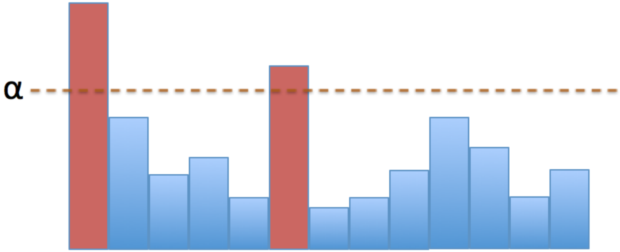
Alternative idea

1. Cycle through rows
2. Each time you see a row with $\tau_i(\mathbf{A}) \geq \alpha$ cut its weight so that $\tau_i(\mathbf{A}) = \alpha$
3. Repeat

Coherence Reducing Reweighting

Alternative idea

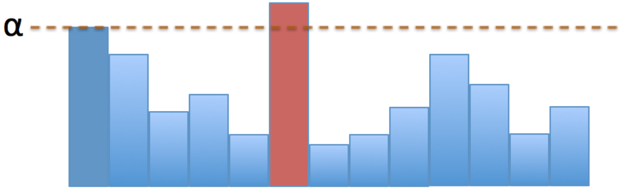
1. Cycle through rows
2. Each time you see a row with $\tau_i(\mathbf{A}) \geq \alpha$ cut its weight so that $\tau_i(\mathbf{A}) = \alpha$
3. Repeat



Coherence Reducing Reweighting

Alternative idea

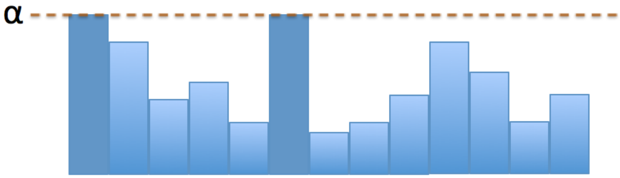
1. Cycle through rows
2. Each time you see a row with $\tau_i(\mathbf{A}) \geq \alpha$ cut its weight so that $\tau_i(\mathbf{A}) = \alpha$
3. Repeat



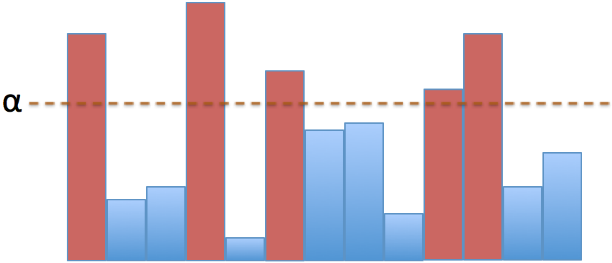
Coherence Reducing Reweighting

Alternative idea

1. Cycle through rows
2. Each time you see a row with $\tau_i(\mathbf{A}) \geq \alpha$ cut its weight so that $\tau_i(\mathbf{A}) = \alpha$
3. Repeat

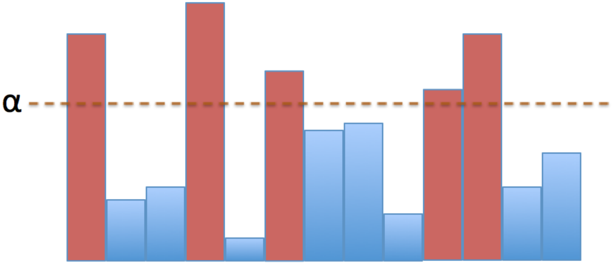


Key observation:



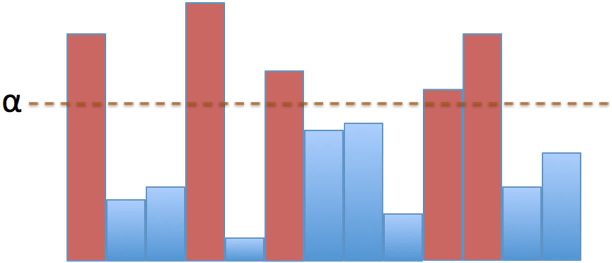
Key observation:

- Decreasing the weight of a row will only increase the leverage score of other rows



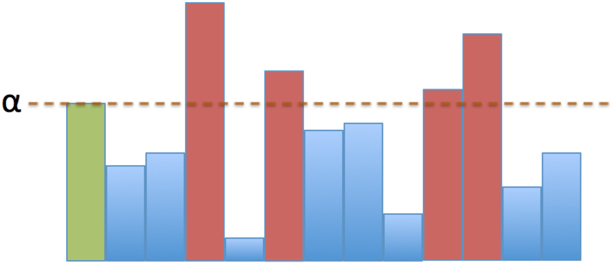
Key observation:

- Decreasing the weight of a row will only increase the leverage score of other rows
- At any step, *every* row that has been reweighted has leverage score $\geq \alpha$ so at most d/α reweighted rows.



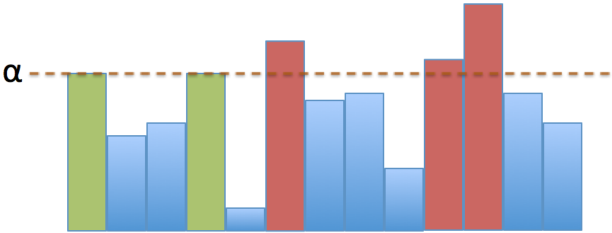
Key observation:

- Decreasing the weight of a row will only increase the leverage score of other rows
- At any step, *every* row that has been reweighted has leverage score $\geq \alpha$ so at most d/α reweighted rows.



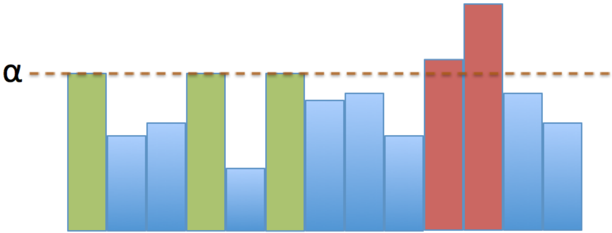
Key observation:

- Decreasing the weight of a row will only increase the leverage score of other rows
- At any step, *every* row that has been reweighted has leverage score $\geq \alpha$ so at most d/α reweighted rows.



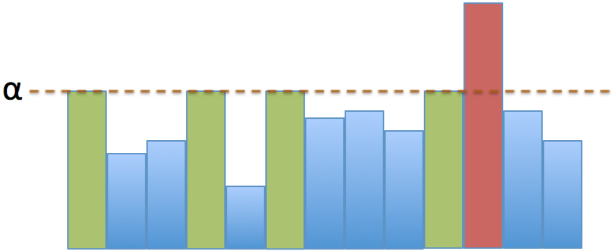
Key observation:

- Decreasing the weight of a row will only increase the leverage score of other rows
- At any step, *every* row that has been reweighted has leverage score $\geq \alpha$ so at most d/α reweighted rows.



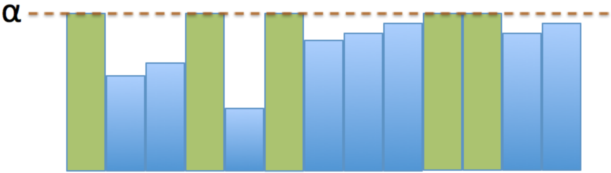
Key observation:

- Decreasing the weight of a row will only increase the leverage score of other rows
- At any step, *every* row that has been reweighted has leverage score $\geq \alpha$ so at most d/α reweighted rows.



Key observation:

- Decreasing the weight of a row will only increase the leverage score of other rows
- At any step, *every* row that has been reweighted has leverage score $\geq \alpha$ so at most d/α reweighted rows.

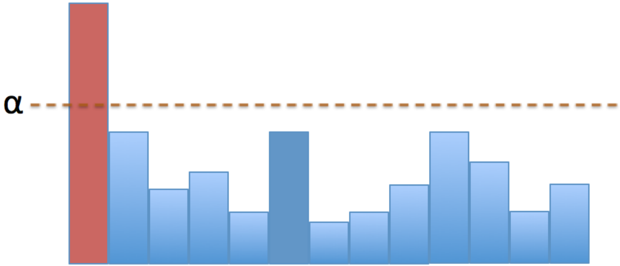


Coherence Reducing Reweighting

But does this reweighting process converge?

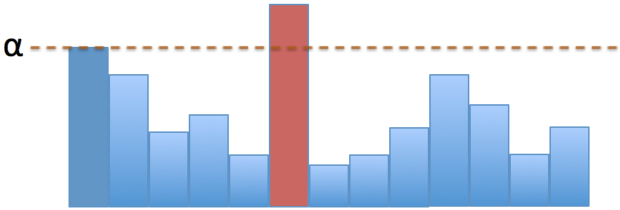
Coherence Reducing Reweighting

But does this reweighting process converge?



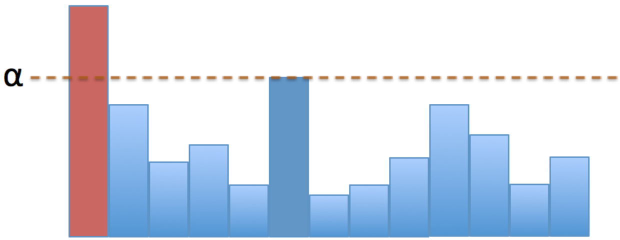
Coherence Reducing Reweighting

But does this reweighting process converge?



Coherence Reducing Reweighting

But does this reweighting process converge?



Coherence Reducing Reweighting

But does this reweighting process converge?

- Rows that keep violating constraint will have weight cut to ≈ 0.

Coherence Reducing Reweighting

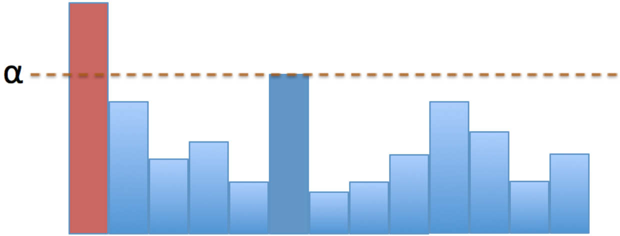
But does this reweighting process converge?

- Rows that keep violating constraint will have weight cut to ≈ 0.
- Can remove them without significantly effecting leverage scores of other rows.

Coherence Reducing Reweighting

But does this reweighting process converge?

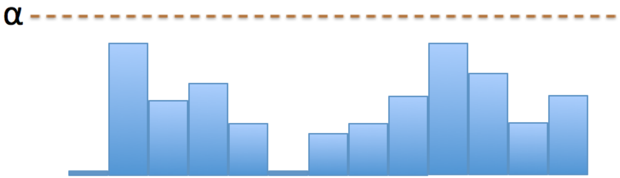
- Rows that keep violating constraint will have weight cut to ≈ 0.
- Can remove them without significantly effecting leverage scores of other rows.



Coherence Reducing Reweighting

But does this reweighting process converge?

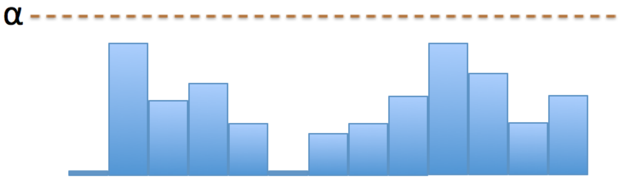
- Rows that keep violating constraint will have weight cut to ≈ 0.
- Can remove them without significantly effecting leverage scores of other rows.



Coherence Reducing Reweighting

But does this reweighting process converge?

- Rows that keep violating constraint will have weight cut to ≈ 0.
- Can remove them without significantly effecting leverage scores of other rows.
- So overall, reweighting d/α rows is enough to cut all leverage scores below α.



Conclusion

Take away:

- Very simple analysis shows how leverage scores can be approximated with uniform sampling.
- Simple iterative spectral approximation algorithms matching state of the art runtimes follow.

Conclusion

Take away:

- Very simple analysis shows how leverage scores can be approximated with uniform sampling.
- Simple iterative spectral approximation algorithms matching state of the art runtimes follow.

Open Questions:

- Analogous algorithms for low rank approximation?
- For other types of regression?
- Generally, our result shows that it is possible to go beyond relative condition number (spectral approximation) invariants for iterative algorithms. Can this type of analysis be useful elsewhere?

Conclusion

Take away:

- Very simple analysis shows how leverage scores can be approximated with uniform sampling.
- Simple iterative spectral approximation algorithms matching state of the art runtimes follow.

Open Questions:

- Analogous algorithms for low rank approximation?
- For other types of regression?
- Generally, our result shows that it is possible to go beyond relative condition number (spectral approximation) invariants for iterative algorithms. Can this type of analysis be useful elsewhere?

Thanks! Questions?