Length Squared Sampling in Matrices

November 2, 2017

Agenda: Sampling to deal with “big” matrices

- **Our definition of “Big data”** Doesn’t fit into RAM,

Agenda: Sampling to deal with “big” matrices

- **Our definition of “Big data”** Doesn’t fit into RAM,
- Obvious thought to deal with a big matrix: Sample some rows and compute only on sampled rows.

Agenda: Sampling to deal with “big” matrices

- **Our definition of “Big data”** Doesn’t fit into RAM,
- Obvious thought to deal with a big matrix: Sample some rows and compute only on sampled rows.
- Uniform Random Sampling won’t do: maybe only $o(1)$ fraction of rows/columns have significant entries.

Agenda: Sampling to deal with “big” matrices

- **Our definition of “Big data”** Doesn’t fit into RAM,
- Obvious thought to deal with a big matrix: Sample some rows and compute only on sampled rows.
- Uniform Random Sampling won’t do: maybe only $o(1)$ fraction of rows/columns have significant entries.
- This lecture: Two problems on matrices:

Agenda: Sampling to deal with “big” matrices

- **Our definition of “Big data”** Doesn’t fit into RAM,
- Obvious thought to deal with a big matrix: Sample some rows and compute only on sampled rows.
- Uniform Random Sampling won’t do: maybe only $o(1)$ fraction of rows/columns have significant entries.
- This lecture: Two problems on matrices:
  - (Approximate) Matrix Multiplication in “near-linear” time.

Agenda: Sampling to deal with "big" matrices

- **Our definition of "Big data"** Doesn't fit into RAM,
- Obvious thought to deal with a big matrix: Sample some rows and compute only on sampled rows.
- Uniform Random Sampling won't do: maybe only $o(1)$ fraction of rows/columns have significant entries.
- This lecture: Two problems on matrices:
  - (Approximate) Matrix Multiplication in "near-linear" time.
  - Compressed Representation of a matrix. Uses Matrix Multiplication Thm. twice in a curious way.

Agenda: Sampling to deal with "big" matrices

- **Our definition of "Big data"** Doesn't fit into RAM,
- Obvious thought to deal with a big matrix: Sample some rows and compute only on sampled rows.
- Uniform Random Sampling won't do: maybe only $o(1)$ fraction of rows/columns have significant entries.
- This lecture: Two problems on matrices:
  - (Approximate) Matrix Multiplication in "near-linear" time.
  - Compressed Representation of a matrix. Uses Matrix Multiplication Thm. twice in a curious way.
- Both will sample rows with probability proportional to length squared.

Agenda: Sampling to deal with "big" matrices

- **Our definition of "Big data"** Doesn't fit into RAM,
- Obvious thought to deal with a big matrix: Sample some rows and compute only on sampled rows.
- Uniform Random Sampling won't do: maybe only $o(1)$ fraction of rows/columns have significant entries.
- This lecture: Two problems on matrices:
  - (Approximate) Matrix Multiplication in "near-linear" time.
  - Compressed Representation of a matrix. Uses Matrix Multiplication Thm. twice in a curious way.
- Both will sample rows with probability proportional to length squared.
- Will use Length Squared Sampling later also for SVD.

Agenda: Sampling to deal with "big" matrices

- **Our definition of "Big data"** Doesn't fit into RAM,
- Obvious thought to deal with a big matrix: Sample some rows and compute only on sampled rows.
- Uniform Random Sampling won't do: maybe only $o(1)$ fraction of rows/columns have significant entries.
- This lecture: Two problems on matrices:
  - (Approximate) Matrix Multiplication in "near-linear" time.
  - Compressed Representation of a matrix. Uses Matrix Multiplication Thm. twice in a curious way.
- Both will sample rows with probability proportional to length squared.
- Will use Length Squared Sampling later also for SVD.
- Thrpughout: Algorithm tosses coins. ("Randomized Algorithm") Data does not. [Data: Worst-case, not average case.]

Matrix Multiplication

- Matrix Multiplication : $A_{m\times n}, B_{n\times q}$. Find AB. Exact (näive) algorithm in $O(mnq)$ time. Better Divide and Conquer Algorithms. Can we do linear time $O(mn + nq)$ approximately? What if A, B cannot be stored in RAM ?

Matrix Multiplication

- Matrix Multiplication : $A_{m\times n}, B_{n\times q}$. Find AB. Exact (näive) algorithm in $O(mnq)$ time. Better Divide and Conquer Algorithms. Can we do linear time $O(mn + nq)$ approximately? What if A, B cannot be stored in RAM ?
  - Can we just take a sample of A, B, multiply to get an approximation to product ?

Matrix Multiplication

- Matrix Multiplication : $A_{m\times n}, B_{n\times q}$. Find AB. Exact (näive) algorithm in $O(mnq)$ time. Better Divide and Conquer Algorithms. Can we do linear time $O(mn + nq)$ approximately? What if A, B cannot be stored in RAM ?
  - Can we just take a sample of A, B, multiply to get an approximation to product ?
  - Sample some entries of A, B ? Need compatible dimensions to multiply !

Matrix Multiplication

- Matrix Multiplication : $A_{m\times n}, B_{n\times q}$. Find AB. Exact (näive) algorithm in $O(mnq)$ time. Better Divide and Conquer Algorithms. Can we do linear time $O(mn + nq)$ approximately? What if A, B cannot be stored in RAM ?
  - Can we just take a sample of A, B, multiply to get an approximation to product ?
  - Sample some entries of A, B ? Need compatible dimensions to multiply !
  - Sample some rows/columns of A, B ?? Any rows/columns ??

Matrix Multiplication

- Matrix Multiplication : $A_{m\times n}, B_{n\times q}$. Find AB. Exact (näive) algorithm in $O(mnq)$ time. Better Divide and Conquer Algorithms. Can we do linear time $O(mn + nq)$ approximately? What if A, B cannot be stored in RAM ?
  - Can we just take a sample of A, B, multiply to get an approximation to product ?
  - Sample some entries of A, B ? Need compatible dimensions to multiply !
  - Sample some rows/columns of A, B ?? Any rows/columns ??
- $AB =$ (1st Col of A)(1st row of B)+(2nd Col of A)(2nd row of B) + $\cdots$. Check.

Matrix Multiplication

- Matrix Multiplication : $A_{m\times n}, B_{n\times q}$. Find AB. Exact (näive) algorithm in $O(mnq)$ time. Better Divide and Conquer Algorithms. Can we do linear time $O(mn + nq)$ approximately? What if A, B cannot be stored in RAM ?
  - Can we just take a sample of A, B, multiply to get an approximation to product ?
  - Sample some entries of A, B ? Need compatible dimensions to multiply !
  - Sample some rows/columns of A, B ?? Any rows/columns ??
- $AB =$ (1st Col of A)(1st row of B)+(2nd Col of A)(2nd row of B) + $\cdots$. Check.
- The r.h.s $=$ sum of n quantities (which happen to be matrices.)

Matrix Multiplication

- Matrix Multiplication : $A_{m\times n}, B_{n\times q}$. Find AB. Exact (näive) algorithm in $O(mnq)$ time. Better Divide and Conquer Algorithms. Can we do linear time $O(mn + nq)$ approximately? What if A, B cannot be stored in RAM ?
  - Can we just take a sample of A, B, multiply to get an approximation to product ?
  - Sample some entries of A, B ? Need compatible dimensions to multiply !
  - Sample some rows/columns of A, B ?? Any rows/columns ??
- $AB =$ (1st Col of A)(1st row of B)+(2nd Col of A)(2nd row of B) + $\cdots$. Check.
- The r.h.s $=$ sum of n quantities (which happen to be matrices.)
- Sample r of these n quantities and hope/prove that their sum (times $\frac{n}{r}$) is a good estimate.

Matrix Multiplication

- Matrix Multiplication : $A_{m\times n}, B_{n\times q}$. Find AB. Exact (näive) algorithm in $O(mnq)$ time. Better Divide and Conquer Algorithms. Can we do linear time $O(mn + nq)$ approximately? What if A, B cannot be stored in RAM ?
  - Can we just take a sample of A, B, multiply to get an approximation to product ?
  - Sample some entries of A, B ? Need compatible dimensions to multiply !
  - Sample some rows/columns of A, B ?? Any rows/columns ??
- $AB =$ (1st Col of A)(1st row of B)+(2nd Col of A)(2nd row of B) + $\cdots$. Check.
- The r.h.s $=$ sum of n quantities (which happen to be matrices.)
- Sample r of these n quantities and hope/prove that their sum (times $\frac{n}{r}$) is a good estimate.
- Try u.a.r. $T \subset \{1, 2, \ldots, n\}$ with $|T| = r$. Does this work for any A, B?

Matrix Multiplication-contd.

- $AB =$ (1st Col of A)(1st row of B)+(2nd Col of A)(2nd row of B) +..
  $AB = \sum_{i=1}^{n} A(:, i)B(i, :)$.

Matrix Multiplication-contd.

- $AB =$ (1st Col of A)(1st row of B)+(2nd Col of A)(2nd row of B) +..
  $AB = \sum_{i=1}^{n} A(:, i)B(i, :)$.
- Uniform Sampling does not work. General non-uniform Sampling: Prob.s of picking columns: $p_1, p_2, \ldots, p_n \geq 0$; Sum $= 1$.

Matrix Multiplication-contd.

- $AB =$ (1st Col of A)(1st row of B)+(2nd Col of A)(2nd row of B) +..
  $AB = \sum_{i=1}^{n} A(:,i)B(i,:)$.
- Uniform Sampling does not work. General non-uniform Sampling: Prob.s of picking columns: $p_1, p_2, \ldots, p_n \geq 0$; Sum $= 1$.
- Pick $j \in \{1, 2, \ldots, n\}$ with Prob$(j) = p_j$ Let $X = A(:,j)B(j,:)$. X is a matrix-valued r.v. Really want to take r i.i.d. copies of j, and of X and take average. But enough to compute mean and variance of one X.

Matrix Multiplication-contd.

- $AB =$ (1st Col of A)(1st row of B)+(2nd Col of A)(2nd row of B) +..
  $AB = \sum_{i=1}^{n} A(:,i)B(i,:)$.
- Uniform Sampling does not work. General non-uniform Sampling: Prob.s of picking columns: $p_1, p_2, \ldots, p_n \geq 0$; Sum $= 1$.
- Pick $j \in \{1, 2, \ldots, n\}$ with Prob$(j) = p_j$ Let $X = A(:,j)B(j,:)$. X is a matrix-valued r.v. Really want to take r i.i.d. copies of j, and of X and take average. But enough to compute mean and variance of one X.
- What is $E(X)$ (entry-wise expectation)?

Matrix Multiplication-contd.

- $AB =$ (1st Col of A)(1st row of B)+(2nd Col of A)(2nd row of B) +.. $AB = \sum_{i=1}^{n} A(:,i)B(i,:)$.
- Uniform Sampling does not work. General non-uniform Sampling: Prob.s of picking columns: $p_1, p_2, \ldots, p_n \geq 0$; Sum $= 1$.
- Pick $j \in \{1, 2, \ldots, n\}$ with Prob$(j) = p_j$ Let $X = A(:,j)B(j,:)$. X is a matrix-valued r.v. Really want to take r i.i.d. copies of j, and of X and take average. But enough to compute mean and variance of one X.
- What is $E(X)$ (entry-wise expectation)?
- $E(X) = \sum_{j=1}^{n} p_j (A(:,j)B(j,:))$. How we do we make it unbiased ?

Matrix Multiplication-contd.

- $AB =$ (1st Col of A)(1st row of B)+(2nd Col of A)(2nd row of B) +.. $AB = \sum_{i=1}^{n} A(:,i)B(i,:)$.
- Uniform Sampling does not work. General non-uniform Sampling: Prob.s of picking columns: $p_1, p_2, \ldots, p_n \geq 0$; Sum $= 1$.
- Pick $j \in \{1, 2, \ldots, n\}$ with Prob$(j) = p_j$ Let $X = A(:,j)B(j,:)$. X is a matrix-valued r.v. Really want to take r i.i.d. copies of j, and of X and take average. But enough to compute mean and variance of one X.
- What is $E(X)$ (entry-wise expectation)?
- $E(X) = \sum_{j=1}^{n} p_j (A(:,j)B(j,:))$. How we do we make it unbiased ?
- Step back: Want to estimate the sum of n real numbers $a_1, a_2, \ldots, a_n$. Pick a $j \in \{1, 2, \ldots, n\}$ with probabilities $p_1, p_2, \ldots, p_n$. How do we scale the picked a_j so that it is an unbiased estimator of sum?

- Pick a_j from $a_1, a_2, \ldots, a_n$ with probabilities $p_1, p_2, \ldots, p_n$. $E(a_j/p_j) = \sum_{j=1}^{n} a_j$.

- Pick a_j from $a_1, a_2, \ldots, a_n$ with probabilities $p_1, p_2, \ldots, p_n$. $E(a_j/p_j) = \sum_{j=1}^{n} a_j$.
- $E\left(\frac{1}{p_j} A(:, j) B(j, :)\right) = AB$. Better have $p_j > 0$.

- Pick a_j from $a_1, a_2, \ldots, a_n$ with probabilities $p_1, p_2, \ldots, p_n$. $E(a_j/p_j) = \sum_{j=1}^{n} a_j$.
- $E\left(\frac{1}{p_j} A(:, j) B(j, :)\right) = AB$. Better have $p_j > 0$.
- What about the error ?

- Pick a_j from $a_1, a_2, \ldots, a_n$ with probabilities $p_1, p_2, \ldots, p_n$. $E(a_j/p_j) = \sum_{j=1}^{n} a_j$.
- $E\left(\frac{1}{p_j} A(:,j)B(j,:)\right) = AB$. Better have $p_j > 0$.
- What about the error ?
- Try writing down the variance of one entry, say the (i,j) th entry. First try the second moment.

$$\sum_{l=1}^{n} p_l \frac{1}{p_l^2} A_{il}^2 B_{lj}^2.$$

- Pick a_j from $a_1, a_2, \ldots, a_n$ with probabilities $p_1, p_2, \ldots, p_n$. $E(a_j/p_j) = \sum_{j=1}^{n} a_j$.
- $E\left(\frac{1}{p_j} A(:,j)B(j,:)\right) = AB$. Better have $p_j > 0$.
- What about the error ?
- Try writing down the variance of one entry, say the (i,j) th entry. First try the second moment.

$$\sum_{l=1}^{n} p_l \frac{1}{p_l^2} A_{il}^2 B_{lj}^2.$$

- How do we bound it for different (i,j) ?

Variance of the Matrix

- Simple Idea (but without it, very complicated): Bound $\sum$ of variances of entries of X. Let $\mathrm{Var}(X)$ denote $\sum_{i,j} \mathrm{Var}(X_{ij})$.

Variance of the Matrix

- Simple Idea (but without it, very complicated): Bound $\sum$ of variances of entries of X. Let $\mathrm{Var}(X)$ denote $\sum_{i,j} \mathrm{Var}(X_{ij})$.
- $\mathrm{Var}(X) = \sum_{i=1}^{m} \sum_{j=1}^{q} \mathrm{Var}\left(x_{ij}\right) \leq \sum_{ij} E\left(x_{ij}^2\right) \leq \sum_{ij} \sum_{l} p_l \frac{1}{p_l^2} a_{il}^2 b_{lj}^2.$

Variance of the Matrix

- Simple Idea (but without it, very complicated): Bound $\sum$ of variances of entries of X. Let $\mathrm{Var}(X)$ denote $\sum_{i,j} \mathrm{Var}(X_{ij})$.
- $\mathrm{Var}(X) = \sum_{i=1}^{m} \sum_{j=1}^{q} \mathrm{Var}\left(x_{ij}\right) \leq \sum_{ij} E\left(x_{ij}^2\right) \leq \sum_{ij} \sum_{l} p_l \frac{1}{p_l^2} a_{il}^2 b_{lj}^2$.
- Exchange order of summations:
  $\sum_{l} \frac{1}{p_l} \sum_{i} a_{il}^2 \sum_{j} b_{lj}^2 = \sum_{l} \frac{1}{p_l} |A(:,l)|^2 |B(l,:)|^2$.

Variance of the Matrix

- Simple Idea (but without it, very complicated): Bound $\sum$ of variances of entries of X. Let $\mathrm{Var}(X)$ denote $\sum_{i,j} \mathrm{Var}(X_{ij})$.
- $\mathrm{Var}(X) = \sum_{i=1}^{m} \sum_{j=1}^{q} \mathrm{Var}\left(x_{ij}\right) \leq \sum_{ij} E\left(x_{ij}^2\right) \leq \sum_{ij} \sum_{l} p_l \frac{1}{p_l^2} a_{il}^2 b_{lj}^2$.
- Exchange order of summations: $\sum_{l} \frac{1}{p_l} \sum_{i} a_{il}^2 \sum_{j} b_{lj}^2 = \sum_{l} \frac{1}{p_l} |A(:,l)|^2 |B(l,:)|^2$.
- What is the best choice of p_j ? It is the one which minimizes the variance of X.

Variance of the Matrix

- Simple Idea (but without it, very complicated): Bound $\sum$ of variances of entries of X. Let $\mathrm{Var}(X)$ denote $\sum_{i,j} \mathrm{Var}(X_{ij})$.
- $\mathrm{Var}(X) = \sum_{i=1}^{m} \sum_{j=1}^{q} \mathrm{Var}\left(x_{ij}\right) \leq \sum_{ij} E\left(x_{ij}^2\right) \leq \sum_{ij} \sum_{l} p_l \frac{1}{p_l^2} a_{il}^2 b_{lj}^2$.
- Exchange order of summations: $\sum_{l} \frac{1}{p_l} \sum_{i} a_{il}^2 \sum_{j} b_{lj}^2 = \sum_{l} \frac{1}{p_l} |A(:,l)|^2 |B(l,:)|^2$.
- What is the best choice of p_j ? It is the one which minimizes the variance of X.
- Suffices to minimize second moment, since $E(X)$ does not depend on p_l (Unbiased)!

Probabilities which minimize variance

- Choose p_k to minimize $\sum_k \frac{1}{p_k} |A(:,k)|^2 |B(k,:)|^2$.

Probabilities which minimize variance

- Choose p_k to minimize $\sum_k \frac{1}{p_k}|A(:,k)|^2|B(k,:)|^2$.
- Calculus: If $a_1, a_2, \ldots, a_n$ are any positive reals, the $p_1, p_2, \ldots, p_n$ minimizing $\sum \frac{a_l}{p_l}$ subject to $p_l \geq 0, \sum_l p_l = 1$ are proportional to $\sqrt{a_l}$. Check.

Probabilities which minimize variance

- Choose p_k to minimize $\sum_k \frac{1}{p_k}|A(:,k)|^2|B(k,:)|^2$.
- Calculus: If $a_1, a_2, \ldots, a_n$ are any positive reals, the $p_1, p_2, \ldots, p_n$ minimizing $\sum \frac{a_l}{p_l}$ subject to $p_l \geq 0, \sum_l p_l = 1$ are proportional to $\sqrt{a_l}$. Check.
- Best Choice- $p_k \propto |A(:,k)|\,|B(k,:)|$, i.e., $p_k = |A(:,k)|\,|B(k,:)|/\sum_l |A(:,l)|\,|B(l,:)|$

Probabilities which minimize variance

- Choose p_k to minimize $\sum_k \frac{1}{p_k}|A(:,k)|^2|B(k,:)|^2$.
- Calculus: If $a_1, a_2, \ldots, a_n$ are any positive reals, the $p_1, p_2, \ldots, p_n$ minimizing $\sum \frac{a_l}{p_l}$ subject to $p_l \geq 0, \sum_l p_l = 1$ are proportional to $\sqrt{a_l}$. Check.
- Best Choice- $p_k \propto |A(:,k)|\,|B(k,:)|$, i.e., $p_k = |A(:,k)|\,|B(k,:)| / \sum_l |A(:,l)|\,|B(l,:)|$
- In the important special case when $B = A^T$, $p_k = |A(:,k)|^2/||A||_F^2$, where, $||A||_F^2 = \sum_{i,j} A_{ij}^2$ is called the Frobenius norm of A.

Probabilities which minimize variance

- Choose p_k to minimize $\sum_k \frac{1}{p_k}|A(:,k)|^2|B(k,:)|^2$.
- Calculus: If $a_1, a_2, \ldots, a_n$ are any positive reals, the $p_1, p_2, \ldots, p_n$ minimizing $\sum \frac{a_l}{p_l}$ subject to $p_l \geq 0, \sum_l p_l = 1$ are proportional to $\sqrt{a_l}$. Check.
- Best Choice- $p_k \propto |A(:,k)|\,|B(k,:)|$, i.e., $p_k = |A(:,k)|\,|B(k,:)|/\sum_l |A(:,l)|\,|B(l,:)|$
- In the important special case when $B = A^T$, $p_k = |A(:,k)|^2/||A||_F^2$, where, $||A||_F^2 = \sum_{i,j} A_{ij}^2$ is called the Frobenius norm of A.
- **Pick columns of A with probabilities proportional to the squared length of the columns.** . Length-Squared Sampling.

Probabilities which minimize variance

- Choose p_k to minimize $\sum_k \frac{1}{p_k}|A(:,k)|^2|B(k,:)|^2$.
- Calculus: If $a_1, a_2, \ldots, a_n$ are any positive reals, the $p_1, p_2, \ldots, p_n$ minimizing $\sum \frac{a_l}{p_l}$ subject to $p_l \geq 0, \sum_l p_l = 1$ are proportional to $\sqrt{a_l}$. Check.
- Best Choice- $p_k \propto |A(:,k)|\,|B(k,:)|$, i.e., $p_k = |A(:,k)|\,|B(k,:)|/\sum_l |A(:,l)|\,|B(l,:)|$
- In the important special case when $B = A^T$, $p_k = |A(:,k)|^2/||A||_F^2$, where, $||A||_F^2 = \sum_{i,j} A_{ij}^2$ is called the Frobenius norm of A.
- **Pick columns of A with probabilities proportional to the squared length of the columns.** . Length-Squared Sampling.
- With these probabilities, we have $\text{Var}(X) \leq \sum_k \frac{|A(:,k)|^2|B(k,:)|^2}{|A(:,k)|\,|B(k,:)|} \sum_{l=1}^n |A(:,l)|\,|B(l,:)|$ $= (\sum_l |A(:,l)|\,|B(l,:)|)^2 \leq ||A||_F^2||B||_F^2$. [Why?]

Probabilities which minimize variance

- Choose p_k to minimize $\sum_k \frac{1}{p_k}|A(:,k)|^2|B(k,:)|^2$.
- Calculus: If $a_1, a_2, \ldots, a_n$ are any positive reals, the $p_1, p_2, \ldots, p_n$ minimizing $\sum \frac{a_l}{p_l}$ subject to $p_l \geq 0, \sum_l p_l = 1$ are proportional to $\sqrt{a_l}$. Check.
- Best Choice- $p_k \propto |A(:,k)|\,|B(k,:)|$, i.e., $p_k = |A(:,k)|\,|B(k,:)|/\sum_l |A(:,l)|\,|B(l,:)|$
- In the important special case when $B = A^T$, $p_k = |A(:,k)|^2/||A||_F^2$, where, $||A||_F^2 = \sum_{i,j} A_{ij}^2$ is called the Frobenius norm of A.
- **Pick columns of A with probabilities proportional to the squared length of the columns.** . Length-Squared Sampling.
- With these probabilities, we have $\text{Var}(X) \leq \sum_k \frac{|A(:,k)|^2|B(k,:)|^2}{|A(:,k)|\,|B(k,:)|} \sum_{l=1}^n |A(:,l)|\,|B(l,:)|$ $= (\sum_l |A(:,l)|\,|B(l,:)|)^2 \leq ||A||_F^2||B||_F^2$. [Why?]
- Another set of probabilities (really length squared): $\frac{|A(:,k)|^2}{||A||_F^2}$ $\frac{||A||_F^2||B||_F^2}{}$

Approximately Length Squared

- Suppose $p_k \geq c|A(:,k)|^2/||A||_F^2$ for some $c \in \Omega(1)$. It may be possible to find such p_k more easily than finding exact lengths. [For eg. by sampling.]

Approximately Length Squared

- Suppose $p_k \geq c|A(:,k)|^2/||A||_F^2$ for some $c \in \Omega(1)$. It may be possible to find such p_k more easily than finding exact lengths. [For eg. by sampling.]
- Still $X = A(:,k)B(k,:)/p_k$ is unbiased estimator of AB (in fact for any p_k!

Approximately Length Squared

- Suppose $p_k \geq c|A(:,k)|^2/||A||_F^2$ for some $c \in \Omega(1)$. It may be possible to find such p_k more easily than finding exact lengths. [For eg. by sampling.]
- Still $X = A(:,k)B(k,:)/p_k$ is unbiased estimator of AB (in fact for any p_k!
- Now, $\text{Var}(X) \leq \sum_k \frac{|A(:,k)|^2|B(k,:)|^2}{p_k} = \frac{1}{c}||A||_F^2||B||_F^2$. Loose only a factor of $1/c^2$.

Reducing Variance

- We saw for r.v. X, we have: $E(X) = AB$ and $\text{Var}(X) \leq ||A||_F ||B||_F$. Good Enough ?

Reducing Variance

- We saw for r.v. X, we have: $E(X) = AB$ and $\text{Var}(X) \leq ||A||_F||B||_F$. Good Enough ?
- But $||AB||_F \leq ||A||_F||B||_F$ and equality could hold. So in best case, error is as much as $||AB||_F$! No good.

Reducing Variance

- We saw for r.v. X, we have: $E(X) = AB$ and $\text{Var}(X) \leq ||A||_F||B||_F$. Good Enough ?
- But $||AB||_F \leq ||A||_F||B||_F$ and equality could hold. So in best case, error is as much as $||AB||_F$! No good.
- What is a general method of reducing the variance ?

Reducing Variance

- We saw for r.v. X, we have: $E(X) = AB$ and $\text{Var}(X) \leq ||A||_F||B||_F$. Good Enough ?
- But $||AB||_F \leq ||A||_F||B||_F$ and equality could hold. So in best case, error is as much as $||AB||_F$! No good.
- What is a general method of reducing the variance ?
- Take s i.i.d copies of X and take average. Variance cut down by a factor of s.

Matrix Multiplication Theorem

Matrix Multiplication Theorem

- A $m \times n$. B $n \times q$.

Matrix Multiplication Theorem

- A $m \times n$. B $n \times q$.
- $AB \approx C\tilde{B}$, where,

Matrix Multiplication Theorem

- A $m \times n$. B $n \times q$.
- $AB \approx C\tilde{B}$, where,
  - $C = [\frac{1}{p_{j_1}} A(:, j_1) | \frac{1}{p_{j_2}} A(:, j_2) \ldots | \frac{1}{p_{j_s}} A(;, j_s)]$, where, $j_1, j_2, \ldots, j_s$ are picked in i.i.d trials according to $\{p_j : j = 1, 2, \ldots, n\}$ satisfying $p_j \geq c|A(:, j)|^2 / ||A||_F^2 \forall j$.

Matrix Multiplication Theorem

- A $m \times n$. B $n \times q$.
- $AB \approx C\tilde{B}$, where,
  - $C = [\frac{1}{p_{j_1}} A(:, j_1) | \frac{1}{p_{j_2}} A(:, j_2) \ldots | \frac{1}{p_{j_s}} A(;, j_s)]$, where, $j_1, j_2, \ldots, j_s$ are picked in i.i.d trials according to $\{p_j : j = 1, 2, \ldots, n\}$ satisfying $p_j \geq c|A(:, j)|^2 / ||A||_F^2 \forall j$.
  - $\tilde{B}$ is the $s \times q$ matrix of corresponding rows of B.

Matrix Multiplication Theorem

- A $m \times n$. B $n \times q$.
- $AB \approx C\tilde{B}$, where,
  - $C = [\frac{1}{p_{j_1}} A(:, j_1) | \frac{1}{p_{j_2}} A(:, j_2) \ldots | \frac{1}{p_{j_s}} A(;, j_s)]$, where, $j_1, j_2, \ldots, j_s$ are picked in i.i.d trials according to $\{p_j : j = 1, 2, \ldots, n\}$ satisfying $p_j \geq c|A(:, j)|^2 / ||A||_F^2 \forall j$.
  - $\tilde{B}$ is the $s \times q$ matrix of corresponding rows of B.
- $E\left(||AB - C\tilde{B}||_F^2\right) \leq \frac{||A||_F^2 \, ||B||_F^2}{cs}$. Implies
  $E\left(||AB - C\tilde{B}||_F\right) \leq \frac{||A||_F \, ||B||_F}{\sqrt{cs}}$.

Matrix Multiplication Theorem

- A $m \times n$. B $n \times q$.
- $AB \approx C\tilde{B}$, where,
  - $C = [\frac{1}{p_{j_1}} A(:, j_1) | \frac{1}{p_{j_2}} A(:, j_2) \ldots | \frac{1}{p_{j_s}} A(;, j_s)]$, where, $j_1, j_2, \ldots, j_s$ are picked in i.i.d trials according to $\{p_j : j = 1, 2, \ldots, n\}$ satisfying $p_j \geq c|A(:, j)|^2 / ||A||_F^2 \forall j$.
  - $\tilde{B}$ is the $s \times q$ matrix of corresponding rows of B.
- $E\left(||AB - C\tilde{B}||_F^2\right) \leq \frac{||A||_F^2 \, ||B||_F^2}{cs}$. Implies
  $E\left(||AB - C\tilde{B}||_F\right) \leq \frac{||A||_F \, ||B||_F}{\sqrt{cs}}$.
- Words: Frobenius norm error goes down as $1/\sqrt{s}$.

Big Data: Implement in 2 passes

- “Big Data” = Cannot be held in RAM.

Big Data: Implement in 2 passes

- "Big Data" = Cannot be held in RAM.
- Do one pass through A, B to compute all the probabilities p_k.

Big Data: Implement in 2 passes

- "Big Data" = Cannot be held in RAM.
- Do one pass through A, B to compute all the probabilities p_k.
- With p_k on hand, toss coins to figure out which set of s columns of A (and corresponding rows of B) we are going to sample.

Big Data: Implement in 2 passes

- "Big Data" = Cannot be held in RAM.
- Do one pass through A, B to compute all the probabilities p_k.
- With p_k on hand, toss coins to figure out which set of s columns of A (and corresponding rows of B) we are going to sample.
- Make a second pass through A, B and pull out the sample.

Big Data: Implement in 2 passes

- "Big Data" = Cannot be held in RAM.
- Do one pass through A, B to compute all the probabilities p_k.
- With p_k on hand, toss coins to figure out which set of s columns of A (and corresponding rows of B) we are going to sample.
- Make a second pass through A, B and pull out the sample.
- Multiply the sample in RAM and return result.
- For error $\leq \varepsilon ||A||_F ||B||_F$ in expectation, $s \geq c/\varepsilon^2$ suffices. For $\varepsilon \in \Omega(1)$, $s \in O(1)$.

Big Data: Implement in 2 passes

- "Big Data" = Cannot be held in RAM.
- Do one pass through A, B to compute all the probabilities p_k.
- With p_k on hand, toss coins to figure out which set of s columns of A (and corresponding rows of B) we are going to sample.
- Make a second pass through A, B and pull out the sample.
- Multiply the sample in RAM and return result.
- For error $\leq \varepsilon ||A||_F ||B||_F$ in expectation, $s \geq c/\varepsilon^2$ suffices. For $\varepsilon \in \Omega(1)$, $s \in O(1)$.
- If $s \in O(1)$, then RAM space needed is linear in $mn + nq$.

Problems solved by length squared and its cousins

- Matrix Multiplication.

Problems solved by length squared and its cousins

- Matrix Multiplication.
- Sketch (Compressed representation) of a matrix (Discussed Next)

Problems solved by length squared and its cousins

- Matrix Multiplication.
- Sketch (Compressed representation) of a matrix (Discussed Next)
- Principal Component Analysis (SVD) (Coming)

Problems solved by length squared and its cousins

- Matrix Multiplication.
- Sketch (Compressed representation) of a matrix (Discussed Next)
- Principal Component Analysis (SVD) (Coming)
- Tensor Optimization

Problems solved by length squared and its cousins

- Matrix Multiplication.
- Sketch (Compressed representation) of a matrix (Discussed Next)
- Principal Component Analysis (SVD) (Coming)
- Tensor Optimization
- Graph Sparsification

Sketch of a large Matrix

- A is a $m \times n$ matrix. m, n large.

Sketch of a large Matrix

- A is a $m \times n$ matrix. m, n large.
- Will show: A can be approximated given just a random sample of rows of A and a random sample of columns of A, provided, the sampling is length-squared. (Not known for other probabilities.)

Sketch of a large Matrix

- A is a $m \times n$ matrix. m, n large.
- Will show: A can be approximated given just a random sample of rows of A and a random sample of columns of A, provided, the sampling is length-squared. (Not known for other probabilities.)
- Can we sketch (approximate) a matrix by a sample of rows ? No. Sample tells us noting about unsampled rows.

Sketch of a large Matrix

- A is a $m \times n$ matrix. m, n large.
- Will show: A can be approximated given just a random sample of rows of A and a random sample of columns of A, provided, the sampling is length-squared. (Not known for other probabilities.)
- Can we sketch (approximate) a matrix by a sample of rows ? No. Sample tells us noting about unsampled rows.
- Say: rank$(A) = k << m, n$. If in "general position", a sample of $100k$ rows should pin down row space. Still don't know for an unsampled row, what linear combination of sampled rows it is.

Sketch of a large Matrix

- A is a $m \times n$ matrix. m, n large.
- Will show: A can be approximated given just a random sample of rows of A and a random sample of columns of A, provided, the sampling is length-squared. (Not known for other probabilities.)
- Can we sketch (approximate) a matrix by a sample of rows ? No. Sample tells us noting about unsampled rows.
- Say: rank$(A) = k << m, n$. If in "general position", a sample of $100k$ rows should pin down row space. Still don't know for an unsampled row, what linear combination of sampled rows it is.
- A sample of $O(k)$ columns should yield this information.

Sketch of a large Matrix

- A is a $m \times n$ matrix. m, n large.
- Will show: A can be approximated given just a random sample of rows of A and a random sample of columns of A, provided, the sampling is length-squared. (Not known for other probabilities.)
- Can we sketch (approximate) a matrix by a sample of rows ? No. Sample tells us noting about unsampled rows.
- Say: rank$(A) = k << m, n$. If in "general position", a sample of $100k$ rows should pin down row space. Still don't know for an unsampled row, what linear combination of sampled rows it is.
- A sample of $O(k)$ columns should yield this information.
- Will rigourously prove error bound without assuming A is low rank.

$$\begin{pmatrix} \\ A \\ \\ \end{pmatrix} \approx$$

$$\begin{pmatrix} \\ C \\ \\ \end{pmatrix} \cdot \begin{pmatrix} \\ U \\ \\ \end{pmatrix} \cdot \begin{pmatrix} \\ R \\ \\ \end{pmatrix}$$

Col Sample $C, R \to U$ Row Sample

Using *CUR*- an example

- Large Corpus of documents. Each doc is a word-frequency vector. Forms a column of large word-doc matrix A.

Using *CUR*- an example

- Large Corpus of documents. Each doc is a word-frequency vector. Forms a column of large word-doc matrix A.
- New Document $\mathbf{v}$ comes in. Want its similarity to each doc in corpus. If similarity = dot product, then, want $\mathbf{v}^T A$.

Using *CUR*- an example

- Large Corpus of documents. Each doc is a word-frequency vector. Forms a column of large word-doc matrix A.
- New Document $\mathbf{v}$ comes in. Want its similarity to each doc in corpus. If similarity = dot product, then, want $\mathbf{v}^T A$.
- Problem: Preprocess A, so that at "query time" given $\mathbf{v}$, can find approximate $\mathbf{v}^T A$ fast. But must bound error for EVERY $\mathbf{v}$. Say we want to find $\mathbf{u}$ so that $|\mathbf{u} - \mathbf{v}^T A| \leq \delta |\mathbf{v}|$.

Using *CUR*- an example

- Large Corpus of documents. Each doc is a word-frequency vector. Forms a column of large word-doc matrix A.
- New Document $\mathbf{v}$ comes in. Want its similarity to each doc in corpus. If similarity = dot product, then, want $\mathbf{v}^T A$.
- Problem: Preprocess A, so that at "query time" given $\mathbf{v}$, can find approximate $\mathbf{v}^T A$ fast. But must bound error for EVERY $\mathbf{v}$. Say we want to find $\mathbf{u}$ so that $|\mathbf{u} - \mathbf{v}^T A| \leq \delta |\mathbf{v}|$.
- Will set $\mathbf{u} = \mathbf{v}^T CUR$. Fast: Do $\mathbf{v}^T C$ then times U then times R.

Using *CUR*- an example

- Large Corpus of documents. Each doc is a word-frequency vector. Forms a column of large word-doc matrix A.
- New Document $\mathbf{v}$ comes in. Want its similarity to each doc in corpus. If similarity = dot product, then, want $\mathbf{v}^T A$.
- Problem: Preprocess A, so that at "query time" given $\mathbf{v}$, can find approximate $\mathbf{v}^T A$ fast. But must bound error for EVERY $\mathbf{v}$. Say we want to find $\mathbf{u}$ so that $|\mathbf{u} - \mathbf{v}^T A| \leq \delta |\mathbf{v}|$.
- Will set $\mathbf{u} = \mathbf{v}^T CUR$. Fast: Do $\mathbf{v}^T C$ then times U then times R.
- Want

$$\text{Max}_{\mathbf{v}} \left| \mathbf{v}^T (CUR - A) \right| / |\mathbf{v}| \leq \delta.$$

Using *CUR*- an example

- Large Corpus of documents. Each doc is a word-frequency vector. Forms a column of large word-doc matrix A.
- New Document $\mathbf{v}$ comes in. Want its similarity to each doc in corpus. If similarity = dot product, then, want $\mathbf{v}^T A$.
- Problem: Preprocess A, so that at "query time" given $\mathbf{v}$, can find approximate $\mathbf{v}^T A$ fast. But must bound error for EVERY $\mathbf{v}$. Say we want to find $\mathbf{u}$ so that $|\mathbf{u} - \mathbf{v}^T A| \leq \delta |\mathbf{v}|$.
- Will set $\mathbf{u} = \mathbf{v}^T CUR$. Fast: Do $\mathbf{v}^T C$ then times U then times R.
- Want
$$\text{Max}_{\mathbf{v}} \left| \mathbf{v}^T (CUR - A) \right| / |\mathbf{v}| \leq \delta.$$
- The maximum has a name - Spectral norm of $A - CUR$. So, want $||A - CUR||_2 \leq \delta$. Will show $E\left(||A - CUR||_2^2\right) \leq \frac{||A||_F^2}{s^{1/3}}$, where $s =$ number of sampled columns. Number of sampled rows $= r$.

Idea

- Write $A = AI$. Pretend multiplying A with I by sampling s columns of A. Proved: Error $\leq ||A||_F||I||_F/\sqrt{s} = ||A||_F\frac{\sqrt{n}}{\sqrt{s}}$.

Idea

- Write $A = AI$. Pretend multiplying A with I by sampling s columns of A. Proved: Error $\leq ||A||_F ||I||_F / \sqrt{s} = ||A||_F \frac{\sqrt{n}}{\sqrt{s}}$.
- Needs $s \geq n$ to get error $\leq ||A||_F$. Useless. Why?

Idea

- Write $A = AI$. Pretend multiplying A with I by sampling s columns of A. Proved: Error $\leq ||A||_F ||I||_F / \sqrt{s} = ||A||_F \frac{\sqrt{n}}{\sqrt{s}}$.
- Needs $s \geq n$ to get error $\leq ||A||_F$. Useless. Why?
- Assume RR^T is invertible. [true if A is not degenerate. Why?]

Idea

- Write $A = AI$. Pretend multiplying A with I by sampling s columns of A. Proved: Error $\leq ||A||_F ||I||_F / \sqrt{s} = ||A||_F \frac{\sqrt{n}}{\sqrt{s}}$.
- Needs $s \geq n$ to get error $\leq ||A||_F$. Useless. Why?
- Assume RR^T is invertible. [true if A is not degenerate. Why?]
- $P = R^T(RR^T)^{-1}R$ acts as identity on row space of R:

Idea

- Write $A = AI$. Pretend multiplying A with I by sampling s columns of A. Proved: Error $\leq ||A||_F ||I||_F / \sqrt{s} = ||A||_F \frac{\sqrt{n}}{\sqrt{s}}$.
- Needs $s \geq n$ to get error $\leq ||A||_F$. Useless. Why?
- Assume RR^T is invertible. [true if A is not degenerate. Why?]
- $P = R^T(RR^T)^{-1}R$ acts as identity on row space of R:
  - (1) $\mathbf{x} \in V \Rightarrow \mathbf{x}^T = \mathbf{y}^T R$. So, $P\mathbf{x} = R^T(RR^T)^{-1}RR^T\mathbf{y} = R^T\mathbf{y} = \mathbf{x}$.

- Instead of the pretend AI, do pretend AP.

Idea

- Write $A = AI$. Pretend multiplying A with I by sampling s columns of A. Proved: Error $\leq ||A||_F ||I||_F / \sqrt{s} = ||A||_F \frac{\sqrt{n}}{\sqrt{s}}$.
- Needs $s \geq n$ to get error $\leq ||A||_F$. Useless. Why?
- Assume RR^T is invertible. [true if A is not degenerate. Why?]
- $P = R^T(RR^T)^{-1}R$ acts as identity on row space of R:
  - (1) $\mathbf{x} \in V \Rightarrow \mathbf{x}^T = \mathbf{y}^T R$. So, $P\mathbf{x} = R^T(RR^T)^{-1}RR^T\mathbf{y} = R^T\mathbf{y} = \mathbf{x}$.
  - (2) If $\mathbf{x} \in V^{\perp}$, then, $P\mathbf{x} = R^T(RR^T)^{-1}R\mathbf{x} = \mathbf{0}$.
- Instead of the pretend AI, do pretend AP.
- Will prove two things which together imply $||A - CUR||$ is small:

Idea

- Write $A = AI$. Pretend multiplying A with I by sampling s columns of A. Proved: Error $\leq ||A||_F ||I||_F / \sqrt{s} = ||A||_F \frac{\sqrt{n}}{\sqrt{s}}$.
- Needs $s \geq n$ to get error $\leq ||A||_F$. Useless. Why?
- Assume RR^T is invertible. [true if A is not degenerate. Why?]
- $P = R^T(RR^T)^{-1}R$ acts as identity on row space of R:
  - (1) $\mathbf{x} \in V \Rightarrow \mathbf{x}^T = \mathbf{y}^T R$. So, $P\mathbf{x} = R^T(RR^T)^{-1}RR^T\mathbf{y} = R^T\mathbf{y} = \mathbf{x}$.
  - (2) If $\mathbf{x} \in V^{\perp}$, then, $P\mathbf{x} = R^T(RR^T)^{-1}R\mathbf{x} = \mathbf{0}$.
- Instead of the pretend AI, do pretend AP.
- Will prove two things which together imply $||A - CUR||$ is small:
  - $||A - AP||_2$ is small from (1) and (2).

Idea

- Write $A = AI$. Pretend multiplying A with I by sampling s columns of A. Proved: Error $\leq ||A||_F ||I||_F / \sqrt{s} = ||A||_F \frac{\sqrt{n}}{\sqrt{s}}$.
- Needs $s \geq n$ to get error $\leq ||A||_F$. Useless. Why?
- Assume RR^T is invertible. [true if A is not degenerate. Why?]
- $P = R^T(RR^T)^{-1}R$ acts as identity on row space of R:
  - (1) $\mathbf{x} \in V \Rightarrow \mathbf{x}^T = \mathbf{y}^T R$. So, $P\mathbf{x} = R^T(RR^T)^{-1}RR^T\mathbf{y} = R^T\mathbf{y} = \mathbf{x}$.
  - (2) If $\mathbf{x} \in V^{\perp}$, then, $P\mathbf{x} = R^T(RR^T)^{-1}R\mathbf{x} = \mathbf{0}$.
- Instead of the pretend AI, do pretend AP.
- Will prove two things which together imply $||A - CUR||$ is small:
  - $||A - AP||_2$ is small from (1) and (2).
  - $C =$ length squared sample of col.s of A. Corres rows of P- can be written as UR. [Hint: P ends in R. Note: R 's rows do not corres to col.s of C.] So, $||AP - CUR||$ small.

Proofs

- **Proposition** $A \approx AP$. I.e., $E\left(||A - AP||_2^2\right)$ is at most $\frac{1}{\sqrt{r}}||A||_F^2$.

Proofs

- **Proposition** $A \approx AP$. I.e., $E\left(||A - AP||_2^2\right)$ is at most $\frac{1}{\sqrt{r}}||A||_F^2$.
- Recall: $||A - AP||_2^2 = \max_{\{\mathbf{x}:|\mathbf{x}|=1\}} |(A - AP)\mathbf{x}|^2$.

Proofs

- **Proposition** $A \approx AP$. I.e., $E\left(||A - AP||_2^2\right)$ is at most $\frac{1}{\sqrt{r}}||A||_F^2$.
- Recall: $||A - AP||_2^2 = \max_{\{\mathbf{x}:|\mathbf{x}|=1\}} |(A - AP)\mathbf{x}|^2$.
- If $\mathbf{x}$ in the row space V of R, $P\mathbf{x} = \mathbf{x}$, so, $(A - AP)\mathbf{x} = \mathbf{0}$.

Proofs

- **Proposition** $A \approx AP$. I.e., $E\left(||A - AP||_2^2\right)$ is at most $\frac{1}{\sqrt{r}}||A||_F^2$.
- Recall: $||A - AP||_2^2 = \max_{\{\mathbf{x}:|\mathbf{x}|=1\}} |(A - AP)\mathbf{x}|^2$.
- If $\mathbf{x}$ in the row space V of R, $P\mathbf{x} = \mathbf{x}$, so, $(A - AP)\mathbf{x} = \mathbf{0}$.
- Every vector is sum of a vector in V plus a vector in $V^\perp$. So, max at some $\mathbf{x} \in V^\perp$ and so $P\mathbf{x} = \mathbf{0}$; $(A - AP)\mathbf{x} = A\mathbf{x}$.

Proofs

- **Proposition** $A \approx AP$. I.e., $E\left(||A - AP||_2^2\right)$ is at most $\frac{1}{\sqrt{r}}||A||_F^2$.
- Recall: $||A - AP||_2^2 = \max_{\{\mathbf{x}:|\mathbf{x}|=1\}} |(A - AP)\mathbf{x}|^2$.
- If $\mathbf{x}$ in the row space V of R, $P\mathbf{x} = \mathbf{x}$, so, $(A - AP)\mathbf{x} = \mathbf{0}$.
- Every vector is sum of a vector in V plus a vector in $V^\perp$. So, max at some $\mathbf{x} \in V^\perp$ and so $P\mathbf{x} = \mathbf{0}$; $(A - AP)\mathbf{x} = A\mathbf{x}$.
- $|A\mathbf{x}|^2 = \mathbf{x}^T A^T A\mathbf{x} = \mathbf{x}^T(A^T A - R^T R)\mathbf{x} \leq ||A^T A - R^T R||_2 |\mathbf{x}|^2 \leq ||A^T A - R^T R||_2$.

Proofs

- **Proposition** $A \approx AP$. I.e., $E\left(||A - AP||_2^2\right)$ is at most $\frac{1}{\sqrt{r}}||A||_F^2$.
- Recall: $||A - AP||_2^2 = \max_{\{\mathbf{x}:|\mathbf{x}|=1\}} |(A - AP)\mathbf{x}|^2$.
- If $\mathbf{x}$ in the row space V of R, $P\mathbf{x} = \mathbf{x}$, so, $(A - AP)\mathbf{x} = \mathbf{0}$.
- Every vector is sum of a vector in V plus a vector in $V^\perp$. So, max at some $\mathbf{x} \in V^\perp$ and so $P\mathbf{x} = \mathbf{0}$; $(A - AP)\mathbf{x} = A\mathbf{x}$.
- $|A\mathbf{x}|^2 = \mathbf{x}^T A^T A\mathbf{x} = \mathbf{x}^T(A^T A - R^T R)\mathbf{x} \leq ||A^T A - R^T R||_2 |\mathbf{x}|^2 \leq ||A^T A - R^T R||_2$.
- Suffices to prove $||A^T A - R^T R||_2^2 \leq ||A||_F^4 / r$.

Proofs

- **Proposition** $A \approx AP$. I.e., $E\left(||A - AP||_2^2\right)$ is at most $\frac{1}{\sqrt{r}}||A||_F^2$.
- Recall: $||A - AP||_2^2 = \max_{\{\mathbf{x}:|\mathbf{x}|=1\}} |(A - AP)\mathbf{x}|^2$.
- If $\mathbf{x}$ in the row space V of R, $P\mathbf{x} = \mathbf{x}$, so, $(A - AP)\mathbf{x} = \mathbf{0}$.
- Every vector is sum of a vector in V plus a vector in $V^\perp$. So, max at some $\mathbf{x} \in V^\perp$ and so $P\mathbf{x} = \mathbf{0}$; $(A - AP)\mathbf{x} = A\mathbf{x}$.
- $|A\mathbf{x}|^2 = \mathbf{x}^T A^T A\mathbf{x} = \mathbf{x}^T(A^T A - R^T R)\mathbf{x} \leq ||A^T A - R^T R||_2 |\mathbf{x}|^2 \leq ||A^T A - R^T R||_2$.
- Suffices to prove $||A^T A - R^T R||_2^2 \leq ||A||_F^4 / r$.
- Matrix Multiplication Theorem! Why?

Proofs

- **Proposition** $A \approx AP$. I.e., $E\left(||A - AP||_2^2\right)$ is at most $\frac{1}{\sqrt{r}}||A||_F^2$.
- Recall: $||A - AP||_2^2 = \max_{\{\mathbf{x}:|\mathbf{x}|=1\}} |(A - AP)\mathbf{x}|^2$.
- If $\mathbf{x}$ in the row space V of R, $P\mathbf{x} = \mathbf{x}$, so, $(A - AP)\mathbf{x} = \mathbf{0}$.
- Every vector is sum of a vector in V plus a vector in $V^\perp$. So, max at some $\mathbf{x} \in V^\perp$ and so $P\mathbf{x} = \mathbf{0}$; $(A - AP)\mathbf{x} = A\mathbf{x}$.
- $|A\mathbf{x}|^2 = \mathbf{x}^T A^T A\mathbf{x} = \mathbf{x}^T(A^T A - R^T R)\mathbf{x} \leq ||A^T A - R^T R||_2 |\mathbf{x}|^2 \leq ||A^T A - R^T R||_2$.
- Suffices to prove $||A^T A - R^T R||_2^2 \leq ||A||_F^4 / r$.
- Matrix Multiplication Theorem! Why?
- Pretend we are multiplying A^T by A by picking col.s of A by length squared sampling....

Proof -II

- **Lemma** $AP \approx CUR$.

Proof -II

- **Lemma** $AP \approx CUR$.
- C is a length squared sample of cols of A.

Proof -II

- **Lemma** $AP \approx CUR$.
- C is a length squared sample of cols of A.
- Want to pick corres rows of $P = R^T(RR^T)^{-1}R$. Can be written as UR for some U.

Proof -II

- **Lemma** $AP \approx CUR$.
- C is a length squared sample of cols of A.
- Want to pick corres rows of $P = R^T(RR^T)^{-1}R$. Can be written as UR for some U.
- Error $E(||AP - CUR||_F^2) \leq ||A||_F^2||P||_F^2/s$ by Matrix Mult Thm.

Proof -II

- **Lemma** $AP \approx CUR$.
- C is a length squared sample of cols of A.
- Want to pick corres rows of $P = R^T(RR^T)^{-1}R$. Can be written as UR for some U.
- Error $E(||AP - CUR||_F^2) \leq ||A||_F^2||P||_F^2/s$ by Matrix Mult Thm.
- Bound $||P||_F$: P has rank r and is an identity matrix on an r dim subspace. Prove any such P has $||P||_F^2 = r$.

Proof -II

- **Lemma** $AP \approx CUR$.
- C is a length squared sample of cols of A.
- Want to pick corres rows of $P = R^T(RR^T)^{-1}R$. Can be written as UR for some U.
- Error $E(||AP - CUR||_F^2) \leq ||A||_F^2||P||_F^2/s$ by Matrix Mult Thm.
- Bound $||P||_F$: P has rank r and is an identity matrix on an r dim subspace. Prove any such P has $||P||_F^2 = r$.
- Putting together, we get $E(||A - CUR||_2^2) \leq ||A||_F^2 \left(\frac{1}{\sqrt{r}} + \frac{r}{s}\right)$.

  Optimal choice: $r = s^{2/3}$.

CUR Theorem

- Hypothesis: A $m \times n$ matrix. r and s be positive integers.

CUR Theorem

- Hypothesis: A $m \times n$ matrix. r and s be positive integers.
- Hypothesis: C an $m \times s$ matrix of s columns of A picked according to length squared sampling and R a matrix of r rows of A picked according to length squared sampling.

CUR Theorem

- Hypothesis: A $m \times n$ matrix. r and s be positive integers.
- Hypothesis: C an $m \times s$ matrix of s columns of A picked according to length squared sampling and R a matrix of r rows of A picked according to length squared sampling.
- Conclusion: We can find from C and R an $s \times r$ matrix U so that

$$E\left(||A - CUR||_2^2\right) \leq ||A||_F^2 \left(\frac{2}{\sqrt{r}} + \frac{2r}{s}\right) = ||A||_F^2 O(1/s^{1/3}),$$

choosing $r = s^{2/3}$