IIT KHARAGPUR

IIT GANDHINAGAR

NPTEL ONLINE
CERTIFICATION COURSES

NPTEL

Scalable Data Science

Lecture 15a: Introduction to Rand NLA Matrix Multiplication

Anirban Dasgupta

Computer Science and Engineering

IIT GANDHINAGAR

IIT Gandhinagar

Indian Institute of Technology Gandhinagar

Numerical Linear Algebra

- Large fraction of ML problems boil down to solving a linear algebraic optimization
  - recommendation systems → matrix factorization
  - ranking → eigenvector calculation
  - supervised models → regression with appropriate loss
  - …
- Such matrices are typically "big", or the method is computationally expensive

Randomized Numerical Linear Algebra

- Randomization (sampling, sketching) allows us to build algorithms that can
  - return approximate solutions (with a quality control parameter)
  - succeed with high confidence
  - scale to massive data
  - have improved computational complexity

[Slides courtesy Michael Mahoney]

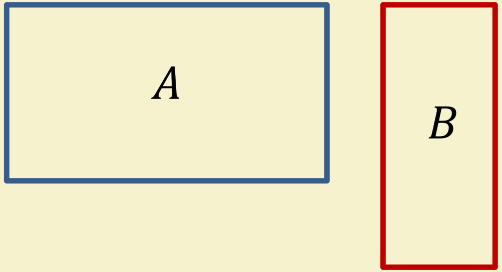
Basic Theme

- By sampling rows/columns/entries of matrices, we will create a smaller representative matrix
  - solve the problem on the smaller matrix

- Alternately, take random combinations of rows/columns of original matrix
  - random projections

Matrix Multiplication

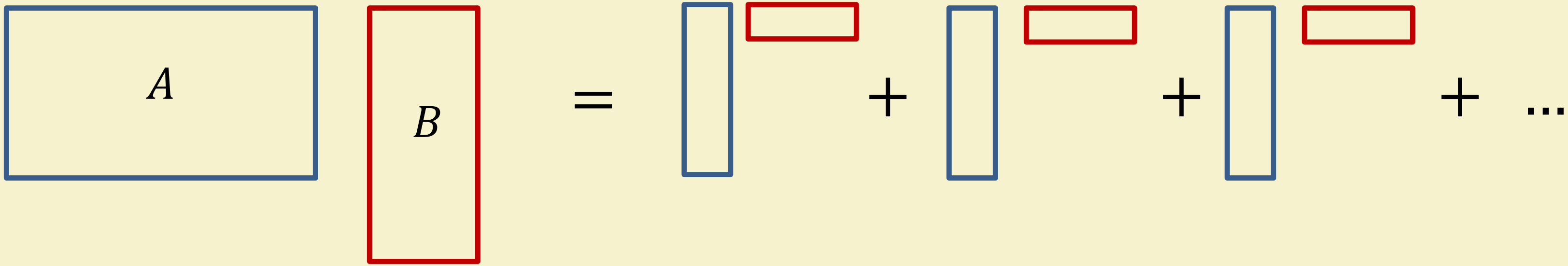
Given $A \in \Re^{m \times n}$, $B \in \Re^{n \times p}$, find out AB



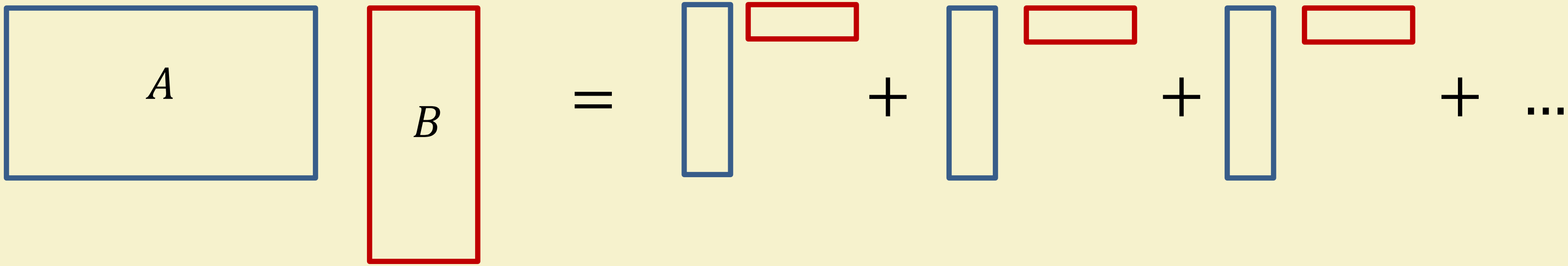
Can we find an approximate solution faster?

Matrix multiplication

- View the product as a sum of rank-one matrices



Matrix multiplication



Design $p_1, p_2 \ldots p_n,\ \sum_i p_i = 1$

For $t = 1 \ldots . c$

pick the i^{th} rank one matrix in the summation with probability p_i , with replacement

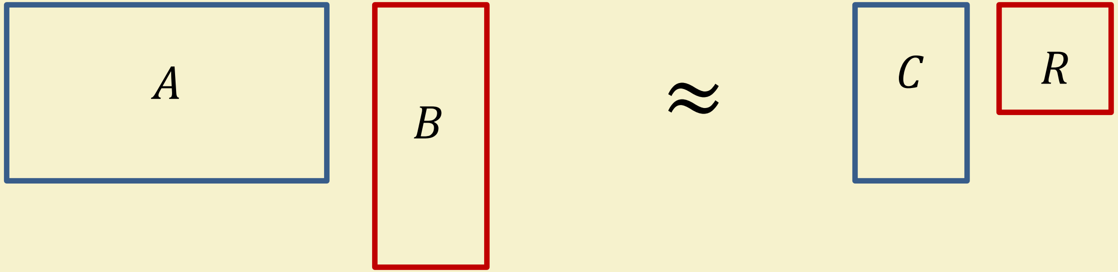
normalize this matrix by $\frac{1}{p_i}$

Approximate this sum by the c terms

Example

$$AB = \sum_{i \in [1,n]} A_{*i}\, B_{i*} \approx \frac{1}{c} \sum_{t \in [1,c]} A_{*j_t} B_{j_t *}$$

Algorithm



1. Pick c columns of A with replacement to form the matrix C and the corresponding rows of B to create R
2. Calculate CR and return

<u>Notes:</u> The columns (rows) are picked with non-uniform probabilities and scaled appropriately

How to choose the probabilities

Choice 1: Uniform, $p_i = \frac{1}{n}\ \forall i$

What we want is $AB \approx CR$

Imagine we had numbers $a_1 b_1, a_2 b_2 \dots a_n b_n$ and we want $\sum_i a_i b_i$

How to choose the probabilities

Choice 1: Uniform, $p_i = \frac{1}{n}\ \forall i$

What we want is $AB \approx CR$

Imagine we had numbers $a_1 b_1, a_2 b_2 \dots a_n b_n$ and we want $\sum_i a_i b_i$

Choosing uniformly has high variance, some $a_i b_i$ could be very large

How to choose the probabilities

Weighted by row and column norm, $p_i = \frac{|A_{*i}||B_{i*}|}{\sum_j |A_{*j}||B_{j*}|}$

For $t = 1 \dots c$

choose j_t column of A and row of B

include $\frac{A_{*j_t}}{\sqrt{c p_{j_t}}}$ in C and $\frac{B_{j_t *}}{\sqrt{c p_{j_t}}}$ in R

Algo in matrix notation

We can write this in terms of a sampling matrix $S \in \Re^{n \times c}$ as

Each column $t \in [1, c]$ has single nonzero entry

$$S_{j_t t} = 1/\sqrt{c p_{j_t}}$$

$$CR = (AS)(S^t B)$$

Simple guarantees

$$(CR)_{ij} = \frac{1}{c} \sum_{t \in [1,c]} \frac{A_{ij_t} B_{j_t j}}{p_{j_t}}$$

Easy to see: $E\left[(CR)_{ij}\right] = (AB)_{ij}$

$$var\left((CR)_{ij}\right) = \frac{1}{c} \sum_{k} (A_{ik}^2 B_{kj}^2)/p_k - \frac{1}{c} (AB)_{ij}^2$$

Derivation

Error in Frobenius norm

We want to bound $|AB \ - CR|_F$

$$E\left[|AB \ - CR|_F^2\right] = E\left[|AB \ - (AS)(S^t B)|_F^2\right] \ = \sum var(CR)_{ij}$$

Using the discussed values for p_i allows us to bound

$$E\left[|AB \ - CR|_F^2\right] \leq \frac{1}{c} \ |A|^2{}_F |B|^2{}_F$$

Error in Frobenius norm

We want to bound $|AB - CR|_F$

$$E\left[|AB - CR|_F^2\right] = \sum var(CR)_{ij}$$

Using the discussed values for p_i allows us to bound

$$E\left[|AB - CR|_F^2\right] \leq \frac{1}{c}\, |A|^2{}_F |B|^2{}_F$$

We can now use Markov's inequality to bound in probability

Can also improve bound using Chernoff style inequalities

Special case $B = A^t$

Sampling probabilities are $p_i = \frac{|A_{*i}|^2}{\sum_j |A_{*j}|^2} = \frac{|A_{*i}|^2}{|A|_F^2}$

Bound becomes

$$|AA^t - CC^t|_F \leq \frac{1}{\sqrt{c}} |A|_F^2$$

Can get improved bound to spectral norm in this case

Using a dense S

Instead of sampling matrix S, can also use a JL matrix $\rightarrow$ data oblivious

E.g. $S \in \Re^{n\times c}$ $\quad S_{ij} = N(0,1)$ $or \pm 1$

S could also be FJLT

Easier to get bound on $|AA^t - CC^t|_F = |AA^t - (AS)(S^tA^t)|_F$

Running time

Using a sampling matrix: $O(mn + np) + O(mcp)$

Using FJLT : $O\big((mn + np + c)\log n\big) + O(mcp)$

(assume $n > m, p$)

Summary

- Randomization and approximation a powerful tool in numerical linear algebra
- Saw two applications
  - Approximating PCA using random projections
  - Approximating matrix multiplication (basis of many results)
- Interchangeable role of sampling and sketching

References:

- Primary reference
  - Lecture notes on Randomized Numerical Linear Algebra by Petros Drineas and Michael Mahoney, https://arxiv.org/abs/1712.08880

Thank You!!