EE270
Large scale matrix computation, optimization and learning

Instructor : Mert Pilanci

Stanford University

Thursday, Jan 14 2020

Randomized Linear Algebra
Lecture 3: Applications of AMM, Error Analysis, Trace Estimation and Bootstrap

Approximate Matrix Multiplication

Algorithm 1 Approximate Matrix Multiplication via Sampling

Input: An $n \times d$ matrix A and an $d \times p$ matrix B, an integer m and probabilities $\{p_k\}_{k=1}^{d}$

Output: Matrices CR such that $CR \approx AB$

1: **for** $t = 1$ to m **do**

2: Pick $i_t \in \{1, ..., d\}$ with probability $\mathbb{P}[i_t = k] = p_k$ in i.i.d. with replacement

3: Set $C^{(t)} = \frac{1}{\sqrt{mp_{i_t}}} A^{(i_t)}$ and $R_{(t)} = \frac{1}{\sqrt{mp_{i_t}}} B_{(i_t)}$

4: **end for**

- We can multiply CR using the classical algorithm
- Complexity $O(nmp)$

AMM mean and variance

$$AB \approx CR = \frac{1}{m}\sum_{t=1}^{m}\frac{1}{p_{i_t}}A^{(i_t)}B_{(i_t)}$$

- Mean and variance of the matrix multiplication estimator

Lemma

- $\mathbb{E}\left[(CR)_{ij}\right] = (AB)_{ij}$
- $\textbf{Var}\left[(CR)_{ij}\right] = \frac{1}{m}\sum_{k=1}^{d}\frac{A_{ik}^2 B_{kj}^2}{p_k} - \frac{1}{m}(AB)_{ij}^2$
- $\mathbb{E}\|AB - CR\|_F^2 = \sum_{ij}\mathbb{E}(AB - CR)_{ij}^2 = \sum_{ij}\textbf{Var}[(CR)_{ij}]$

$$= \frac{1}{m}\sum_{k=1}^{d}\frac{\sum_i A_{ik}^2 \sum_j B_{kj}^2}{p_k} - \frac{1}{m}\|AB\|_F^2$$

$$= \frac{1}{m}\sum_{k=1}^{d}\frac{1}{p_k}\|A^{(k)}\|_2^2\|B_{(k)}\|_2^2 - \frac{1}{m}\|AB\|_F^2$$

Optimal sampling probabilities

- Nonuniform sampling

$$p_k = \frac{\|A^{(k)}\|_2 \|B^{(k)}\|_2}{\sum_i \|A^{(k)}\|_2 \|B^{(k)}\|_2}$$

- minimizes $\mathbb{E}\|AB - CR\|_F$
- $\mathbb{E}\|AB - CR\|_F^2 = \frac{1}{m} \sum_{k=1}^{d} \frac{1}{p_k} \|A^{(k)}\|_2^2 \|B_{(k)}\|_2^2 - \frac{1}{m}\|AB\|_F^2$

$$= \frac{1}{m} \left(\sum_{k=1}^{d} \|A^{(k)}\|_2 \|B_{(k)}\|_2 \right)^2 - \frac{1}{m}\|AB\|_F^2$$

  is the optimal error

Final Probability Bound for ℓ_2-norm sampling

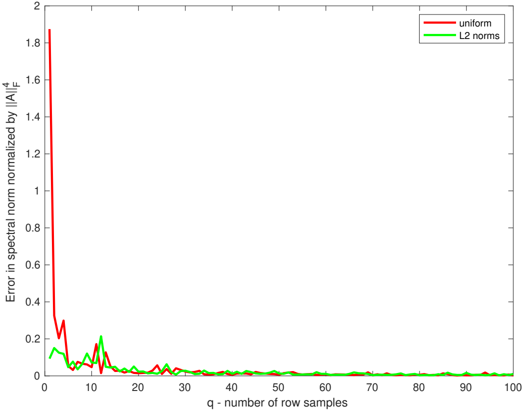
- For any $\delta > 0$, set $m = \frac{1}{\delta\,\epsilon^2}$ to obtain

$$\mathbb{P}\left[\|AB - CR\|_F > \epsilon \|A\|_F \|B\|_F\right] \leq \delta \tag{1}$$

- i.e., $\|AB - CR\|_F < \epsilon \|A\|_F \|B\|_F$ with probability $1 - \delta$
- note that m is independent of any dimensions

Numerical simulations for AMM

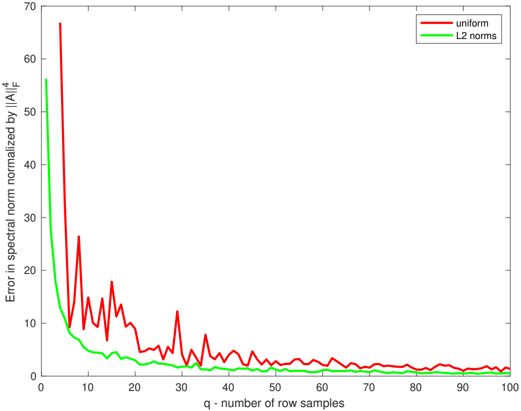
- Approximating $A^T A$
  a subset of the CIFAR dataset



Numerical simulations for AMM

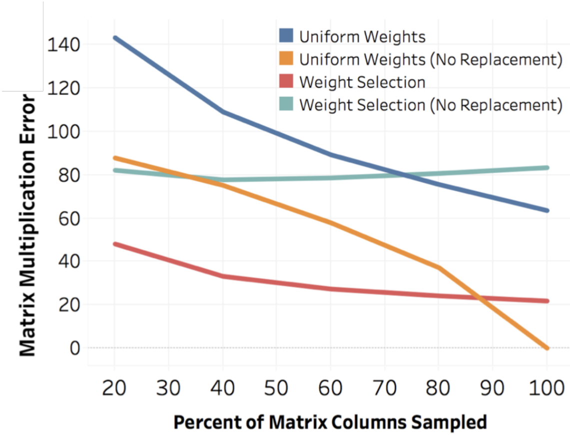
- Approximating $A^T A$

  sparse matrix from a computational fluid dynamics model



SuiteSparse Matrix Collection: https://sparse.tamu.edu

Sampling with replacement vs without replacement



SuiteSparse Matrix Collection: https://sparse.tamu.edu

Plancher et. al. Application of Approximate Matrix Multiplication to Neural Networks and Distributed SLAM,2019.

Applications of Approximate Matrix Multiplication

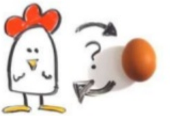
- Simultaneous Localization and Mapping (SLAM)

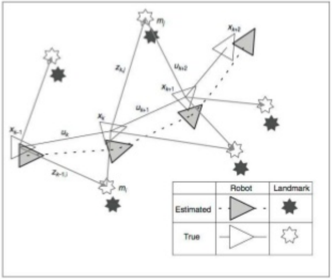
The task of SLAM

Given a Robot with sensor set, at the same time:

- Construct a model (*the Map*) of the environment.
- Estimate *the State* of the robot (pose, velocity, etc.) in *the Map*

SLAM is *chicken-or-egg* problem.





Applications of Approximate Matrix Multiplication

Algorithm 1 DSLAM

$X_0, \Sigma_0 \leftarrow X_{init}, \Sigma_{init}$
for $i = 1 \ldots T$ **do**
    $X_{t|t-1} = f(X_{t-1}, U_t)$
    $F = \frac{\partial f(X_{t-1}, U_t)}{\partial X_{t-1}}$
    $\Sigma_{t|t-1} = F \Sigma_{t-1} F^T + Q_t$
    $y_t = h(X_{t-1})$
    $y_{t|t-1} = h(X_{t|t-1})$
    $H = \frac{\partial h(X_{t-1})}{\partial X_{t-1}}$
    $S = H \Sigma_{t|t-1} H^T + R_t$
    $K = \Sigma_{t|t-1} H^T S^{-1}$
    $X_t = X_{t|t-1} + K(y_t - y_{t|t-1})$
    $\Sigma_t = (I - KH)\Sigma_{t|t-1}$
end for

Motion Update

Measurement Update

Plancher et. al. Application of Approximate Matrix Multiplication to Neural Networks and Distributed SLAM,2019.

Applications of Approximate Matrix Multiplication

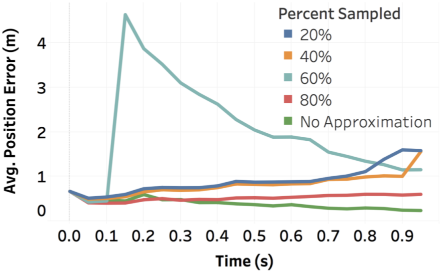
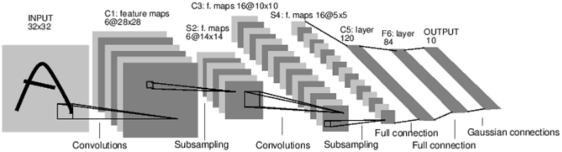


Fig. 6. Error in position estimations over time averaged over 10 trials for DSLAM under various levels of approximation.

Plancher et. al. Application of Approximate Matrix Multiplication to Neural Networks and Distributed SLAM,2019.

Neural Networks

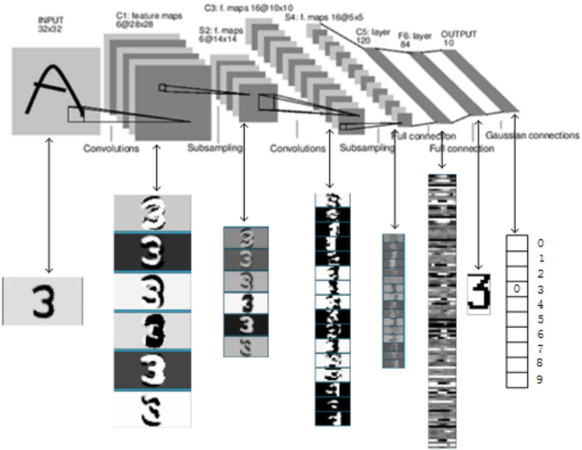
- Given image x
- Classify into M classes
- Neural network $f(x) = W_L(\ldots s(W_2(s(W_1 x))))$
- $W_1, \ldots, W_L$ are trained weight matrices



A Full Convolutional Neural Network (LeNet)

LeCun et al. (1998)

Neural Networks



LeCun et al. (1998)

AMM for neural networks

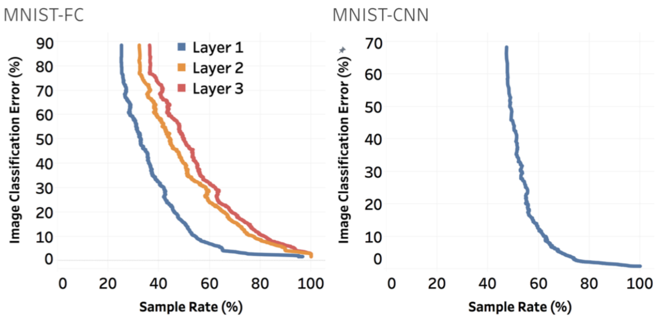


Fig. 3. Average image classification error for Fully-Connected (MNIST-FC, left) and Convolutional (MNIST-CNN, right) NN layers and corresponding rate of sampling. To maintain 97% classification accuracy, only the first layer in MNIST-FC should be approximated (sample rate 76%), while both convolutional layers of MNIST-CNN can be approximated (sample rate 82%).

Plancher et. al. Application of Approximate Matrix Multiplication to Neural Networks and Distributed SLAM,2019.

Probing the actual error

- $AB \approx CR$
- $\Delta \triangleq AB - CR$
- How large is the error $\|\Delta\|_F$?
- $\|\Delta\|_F^2 = \mathbf{tr}\left(\Delta^T \Delta\right)$
- trace of a matrix B
- $\mathbf{tr}\, B) \triangleq \sum_i B_{ii}$
- trace estimation

Trace estimation

- Let B an $n \times n$ symmetric matrix
- Let $u_1, \ldots, u_n$ be n i.i.d. samples of a random variable U with mean zero and variance σ^2
- **Lemma**
  $\mathbb{E}[u^T B u] = \sigma^2 \mathbf{tr}(B)$

  $\mathbf{Var}[u^T B u] = 2\sigma^4 \sum_{i \neq j} B_{ij}^2 + \left(\mathbb{E}[U^4] - \sigma^4\right) \sum_i B_{ii}^2$

Trace estimation: optimal sampling distribution

- Let B an $n \times n$ symmetric matrix
- Let $u_1, \ldots, u_n$ be n i.i.d. samples of a random variable U with mean zero and variance σ^2

  $\mathbb{E}[u^T B u] = \sigma^2 \mathbf{tr}(B)$

  $\mathbf{Var}[u^T B u] = 2\sigma^4 \sum_{i \neq j} B_{ij}^2 + \left(\mathbb{E}[U^4] - \sigma^4\right) \sum_i B_{ii}^2$
- minimum variance unbiased estimator

$$\min_{p(U)} \ \mathbf{Var}[u^T B u]$$

$$\text{subject to } \mathbb{E}[u^T B u] = \mathbf{tr}(B)$$

Trace estimation: optimal sampling distribution

- Let B an $n \times n$ symmetric matrix
- Let $u_1, \ldots, u_n$ be n i.i.d. samples of a random variable U with mean zero and variance σ^2

  $\mathbb{E}[u^T B u] = \sigma^2 \mathbf{tr}(B)$

  $\mathbf{Var}[u^T B u] = 2\sigma^4 \sum_{i \neq j} B_{ij}^2 + \left(\mathbb{E}[U^4] - \sigma^4\right) \sum_i B_{ii}^2$
- minimum variance unbiased estimator

$$\min_{p(U)} \ \mathbf{Var}[u^T B u]$$

$$\text{subject to } \mathbb{E}[u^T B u] = \mathbf{tr}(B)$$

- $\mathbf{Var}(U^2) = \mathbb{E}[U^4] - \sigma^4 \geq 0$
- minimized when $\mathbf{Var}(U^2) = 0$
- $U^2 = 1$ with probability one

Optimal trace estimation

- Let B be an $n \times n$ symmetric matrix with non-zero trace

  Let U be the discrete random variable which takes values $1, -1$ each with probability $\frac{1}{2}$ (Rademacher distribution)

  Let $u = [u_1, ..., u_n]^T$ be i.i.d. $\sim U$
- $u^T Bu$ is an unbiased estimator $\mathbf{tr}(B)$ and

$$\mathbf{Var}[u^T Bu] = 2 \sum_{i \neq j} B_{ij}^2 .$$

- U is the unique variable amongst zero mean random variables for which $u^T Bu$ is a minimum variance, unbiased estimator of $\mathbf{tr}(B)$.

  Hutchinson (1990)

Application to Approximate Matrix Multiplication

- $\|AB - CR\|_F^2 = \mathbf{tr}\big((AB - CR)^T(AB - CR)\big)$
- can be estimated via
- $u^T(AB - CR)^T(AB - CR)u = \|(AB - CR)u\|_2^2$
- only requires matrix-vector products

  where $u = [u_1, ..., u_n]^T$ is i.i.d. ± 1 each with probability $\frac{1}{2}$
- variance can be reduced by averaging independent trials

Sampling/Sketching Matrix Formalism

- Define the sampling matrix

$$\hat{S}_{ij} = \begin{cases} 1 & \text{if the } i\text{-th column of } A \text{ is chosen in the } j\text{-th trial} \\ 0 & \text{otherwise} \end{cases}$$

- diagonal re-weighting matrix

$$D_{tt} = \frac{1}{\sqrt{mp_{i_t}}}$$

Sampling/Sketching Matrix Formalism

- Define the sampling matrix

$$\hat{S}_{ij} = \begin{cases} 1 & \text{if the } i\text{-th column of } A \text{ is chosen in the } j\text{-th trial} \\ 0 & \text{otherwise} \end{cases}$$

- diagonal re-weighting matrix

$$D_{tt} = \frac{1}{\sqrt{mp_{i_t}}}$$

- $AB \approx CR$

  $C = A\hat{S}D$ and $R = D\hat{S}^T B$
- let $S = D\hat{S}^T$

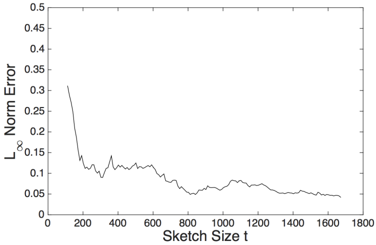
  $CR = A\hat{S}DD\hat{S}^T B = AS^T SB$

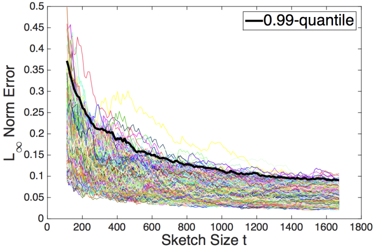
Estimating the entry-wise error

- infinity norm error
- $\varepsilon(S) \triangleq \|AS^TSB - AB\|_\infty = \max_{ij} |(AS^TSB)_{ij} - (AB)_{ij}|$
- 0.99-quantile of $\varepsilon(S)$ is the tightest upper bound that holds with probability at least 0.99

Estimating the entry-wise error

- infinity norm error
- $\varepsilon(S) \triangleq \|AS^T SB - AB\|_\infty = \max_{ij} |(AS^T SB)_{ij} - (AB)_{ij}|$
- 0.99-quantile of $\varepsilon(S)$ is the tightest upper bound that holds with probability at least 0.99





Estimating the entry-wise error

- infinity norm error
- $\varepsilon(S) \triangleq \|AS^TSB - AB\|_\infty = \max_{ij} |(AS^TSB)_{ij} - (AB)_{ij}|$
- 0.99-quantile of $\varepsilon(S)$ is the tightest upper bound that holds with probability at least 0.99
- Bootstrap procedure:

  For $b = 1, ..., B$ **do**

  sample m numbers with replacement from $\{1, ..., m\}$

  form S_b by selecting the the respective rows of S

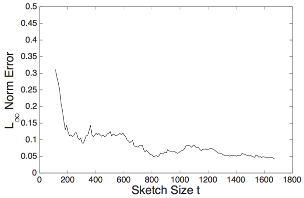
  compute $\hat{\varepsilon}_b = \|AS_b^TS_bB - AS^TSB\|_\infty$

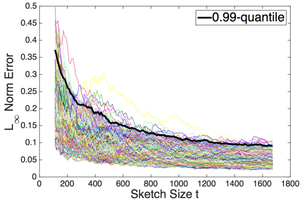
  return 0.99-quantile of the values $\hat{\varepsilon}_1, ..., \hat{\varepsilon}_B$

  e.g., sort in increasing order and return $\lfloor 0.99B \rfloor$-th value
- imitates the random mechanism that originally generated AS^TSB

A Bootstrap Method for Error Estimation in Randomized Matrix Multiplication. Lopes et al.

Extrapolating the error





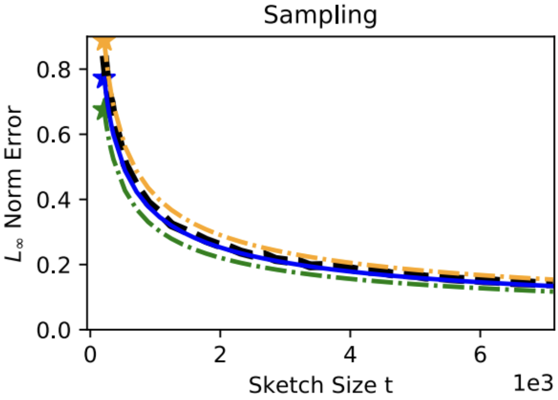
- $\varepsilon(S) \triangleq \|AS^T SB - AB\|_\infty$
- for sufficiently large m
- 0.99-quantile of $\varepsilon(S) \approx \frac{\kappa}{\sqrt{m}}$

  where κ is an unknown number
- given initial sketch of size m_0

  we can extrapolate the error for $m > m_0$ via the Bootstrap estimate as

$$\frac{\sqrt{m_0}}{\sqrt{m}}\hat{\varepsilon}(S)$$

Extrapolation: Numerical example



- Protein dataset ($n = 17766, d = 356$)
  The black line is the 0.99-quantile as a function of m. The blue star is the average bootstrap estimate at the initial sketch size $m_0 = 500$, and the blue line represents the average extrapolated estimate derived from the starting value m_0.

Questions?