Random-sampling based techniques for approximating matrix multiplication

Shahrzad Kianidehkordi

Feb 2021, IT@UofT

Speed Up Large-Scale Matrix Multiplication

Motivation:

- Matrices are fundamental mathematical structures for representing data and can be too large to fit in memory.
- Matrix multiplication is a core building block for numerous scientific computing and can be too time consuming.

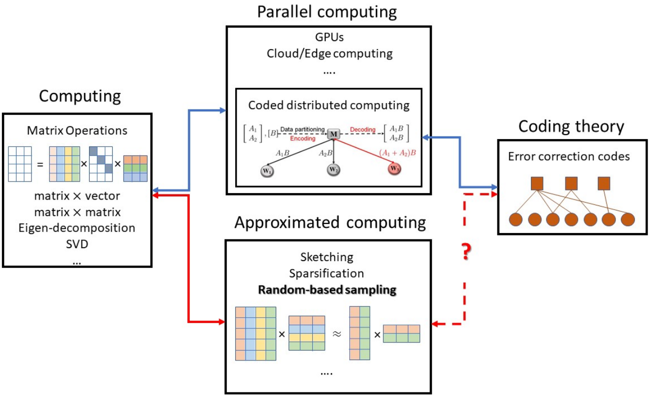
⇒ Large-scale matrix multiplication is parallelized and/or approximated.

Speed Up Large-Scale Matrix Multiplication

Motivation:

- Matrices are fundamental mathematical structures for representing data and can be too large to fit in memory.
- Matrix multiplication is a core building block for numerous scientific computing and can be too time consuming.

⇒ Large-scale matrix multiplication is parallelized and/or approximated.



Randomized Approximated Matrix Multiplication

Problem statement: Given an $m \times n$ matrix A and an $n \times p$ matrix B approximate the product AB.

Randomized Approximated Matrix Multiplication

Problem statement: Given an $m \times n$ matrix A and an $n \times p$ matrix B approximate the product AB.

Classic randomized algorithms:

- P. Drineas, R. Kannan, and M. W. Mahoney, "Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication," SIAM Journal on Computing, 36(1), pp.132-157, 2006.
- P. Drineas, R. Kannan, "Fast Monte-Carlo algorithms for approximate matrix multiplication," Proceedings IEEE Symposium on Foundations of Computer Science (pp. 452-459). 2001.
- E. Cohen, D. Lewis, "Approximating matrix multiplication for pattern recognition tasks," Journal of Algorithms, 30(2), pp.211-252, 1999.

Randomized Approximated Matrix Multiplication

Problem statement: Given an $m \times n$ matrix A and an $n \times p$ matrix B approximate the product AB.

Classic randomized algorithms:

- P. Drineas, R. Kannan, and M. W. Mahoney, "Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication," SIAM Journal on Computing, 36(1), pp.132-157, 2006.
- P. Drineas, R. Kannan, "Fast Monte-Carlo algorithms for approximate matrix multiplication," Proceedings IEEE Symposium on Foundations of Computer Science (pp. 452-459). 2001.
- E. Cohen, D. Lewis, "Approximating matrix multiplication for pattern recognition tasks," Journal of Algorithms, 30(2), pp.211-252, 1999.

The general idea ...

- Randomly sample columns/rows/entries of the matrices with carefully constructed sampling probabilities to form an approximated result.

Column/Row Uniform Sampling[1]

Problem Statement: Approximate the sum of n rank-one matrices,
(Notation: A_{i*} is the i-th row of A and A_{*j} is the j-th column)

$$AB = \sum_{k=1}^{n} A_{*k} B_{k*}, \text{ where } A_{*k} B_{k*} \in \mathbb{R}^{m \times p}$$

A sampling approach:

- Choose uniformly at random $c = \mathcal{O}(1)$ integers from $\{1, \ldots, n\}$.
- In one pass, form the matrix C consisting of the chosen columns in A and form the matrix R consisting of the corresponding rows in B.
- Is CR product a good approximation for AB product?
  Often NO. But if for all $k = 1, \ldots, n$, $|A_{*k}||B_{k*}|$ is close to its mean value $(\frac{1}{n} \sum_k |A_{*k}||B_{k*}|)$, then YES.
  (Notation: $|x|^2 = \sum_i x_i^2$ is Euclidean norm of a vector)

[1] Drineas et al., "Fast Monte-Carlo algorithms for approximate matrix multiplication," FOCS' 2001.

Column/Row NU Sampling [2,3]

Problem Statement: Approximate the sum of n rank-one matrices,

$$AB = \sum_{k=1}^{n} A_{*k} B_{k*}, \text{ where } A_{*k} B_{k*} \in \mathbb{R}^{m \times p}$$

[2] Mahoney et al., "Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication," SIAM J. Comput' 2006

[3] Drineas et al., "Fast Monte-Carlo algorithms for approximate matrix multiplication," FOCS' 2001.

Column/Row NU Sampling [2,3]

Problem Statement: Approximate the sum of n rank-one matrices,

$$AB = \sum_{k=1}^{n} A_{*k} B_{k*}, \text{ where } A_{*k} B_{k*} \in \mathbb{R}^{m \times p}$$

A sampling approach:

- Fix a set of probabilities $p_i, i = 1, \ldots, n$, summing up to 1
- For $t = 1, \ldots, c$ set $j_t = i$ with probability $P(j_t = i) = p_i$ (pick c terms of the sum, with replacement, with respect to the p_i)
- Approximate the product AB by summing the c terms, after scaling

$$AB = \sum_{k=1}^{n} A_{*k} B_{k*} \approx \sum_{t=1}^{c} \frac{1}{c p_{j_t}} A_{*j_t} B_{j_t *}$$

---

[2] Mahoney et al., "Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication," SIAM J. Comput' 2006

[3] Drineas et al., "Fast Monte-Carlo algorithms for approximate matrix multiplication," FOCS' 2001.

Column/Row NU Sampling (Matrix Notation) [4] [5]

The same algorithm in matrix notation:

- Pick c columns of A to form an $m \times c$ matrix C and the corresponding c rows of B to form an $c \times p$ matrix R.
- Rescale the columns/rows prior to including them in C/R
- Approximate AB multiplication by CR multiplication. ($A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}, C \in \mathbb{R}^{m\times c}, R \in \mathbb{R}^{c\times p}$)

In other words, use a sampling matrix S. S is a $n \times c$ matrix, the t-th column ($t = 1, \ldots, c$) has one non-zero:

$$S_{j,t} = \frac{1}{\sqrt{cp_{j_t}}}$$

Clearly $AB \approx CR = (AS)(S^T B)$. (Random Sketch)

[4] Mahoney et al., "Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication," SIAM J. Comput' 2006

[5] Drineas et al., "Fast Monte-Carlo algorithms for approximate matrix multiplication," FOCS' 2001.

Column/Row NU Sampling (Some Lemmas) [6][7]

Some lemmas:

- For any sampling probabilities:

$$\mathbb{E}((CR)_{i,j}) = \sum_{t=1}^{c}\sum_{k=1}^{n} p_k \frac{A_{i,k}B_{k,j}}{cp_k} = (AB)_{i,j}$$

$$Var((CR)_{i,j}) = \frac{1}{c}\sum_{k=1}^{n} \frac{A_{i,k}^2 B_{k,j}^2}{p_k} - \frac{1}{c}(AB)_{i,j}^2$$

- From these, it's easy to bound $\mathbb{E}(\|AB - CR\|_F)$.
  (Notation: The Frobenius norm for a matrix A is $\|A\|_F^2 = \sum_{i,j} A_{i,j}^2$.)
- For probabilities $p_i \propto |A_{*i}||B_{i*}|$, $\mathbb{E}(\|AB - CR\|_F)$ is minimized and

$$\|AB - CR\|_F = \mathcal{O}(\|A\|_F \|B\|_F / \sqrt{c})$$

[6] Mahoney et al., "Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication," SIAM J. Comput' 2006

[7] Drineas et al., "Fast Monte-Carlo algorithms for approximate matrix multiplication," FOCS' 2001.

Column/Row Sampling Without Replacement[8]

Problem Statement: Approximate the sum of n rank-one matrices,

$$AB = \sum_{k=1}^{n} A_{*k} B_{k*}, \text{ where } A_{*k} B_{k*} \in \mathbb{R}^{m \times p}$$

Lemma: If $p_k = \frac{1}{n}$, then with probability $1 - \sigma$

$$\|AB - CR\|_F^2 = \mathcal{O}\left(\frac{1}{\sigma} \left(\frac{n}{c} - 1 \right) \sum_k |A_{*k}|^2 |B_{k*}|^2 \right)$$

Advantage: For n=c, the above error becomes zero, while in Sampling with Replacement this was not the case.
Disadvantage: For uniform sampling ($p_k = \frac{1}{n}$), the error can blow up. For non-uniform sampling, error analysis is hard.

---

[8] Drineas et al., "Fast Monte-Carlo algorithms for approximate matrix multiplication," FOCS' 2001.

Element-Wise NU Sampling [9][10]

Problem statement: Approximate each element of AB matrix, i.e., $(AB)_{i,j}$ for all $i \in [m]$ and $j \in [p]$

[9]Mahoney et al., "Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication," SIAM J. Comput' 2006

[10]Drineas et al., "Fast Monte-Carlo algorithms for approximate matrix multiplication," FOCS' 2001.

Element-Wise NU Sampling [9][10]

Problem statement: Approximate each element of AB matrix, i.e., $(AB)_{i,j}$ for all $i \in [m]$ and $j \in [p]$

A sampling approach:

- Fix two sets of probabilities $p_{i,j}$ and $q_{j,k}$, $i = 1, \ldots, m$, $j = 1, \ldots, n$, $k = 1, \ldots, p$ such that $0 \le p_{i,j} \le 1$ and $0 \le q_{j,k} \le 1$.
- Each element (i, j) of A and B are randomly and independently either zeroed out or kept and rescaled, constructing matrices C and R.

$$C_{i,j} = \begin{cases} \frac{A_{i,j}}{p_{i,j}}, & \text{with prob. } p_{i,j} \\ 0, & \text{otherwise} \end{cases}, R_{i,j} = \begin{cases} \frac{B_{i,j}}{q_{i,j}}, & \text{with prob. } q_{i,j} \\ 0, & \text{otherwise} \end{cases}.$$

- Approximate AB with CR multiplication

In other words, use random matrices E and D

$C = A + E, R = B + D \to C$ and R are sparse. (Random Sparsification)

[9] Mahoney et al., "Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication," SIAM J. Comput' 2006

[10] Drineas et al., "Fast Monte-Carlo algorithms for approximate matrix multiplication," FOCS' 2001.

Element-Wise NU Sampling (Some Lemmas) [11][12]

Some lemmas:

- For any sampling probabilities:

$$\mathbb{E}((CR)_{i,j}) = \sum_{k=1}^{n} \mathbb{E}(C_{i,k})\mathbb{E}(R_{k,j}) = (AB)_{i,j}$$

$$Var((CR)_{i,j}) = \sum_{k=1}^{n} \frac{A_{i,k}^2}{p_{i,k}} \frac{B_{k,j}^2}{q_{k,j}} - \sum_{k=1}^{n} A_{i,k}^2 B_{k,j}^2$$

- From these, it's easy to bound $\mathbb{E}(\|AB - CR\|_2)$. (Notation: The spectral norm is $\|A\|_2 = \sup_{x \in \mathbb{R}^n, x \neq 0} |Ax|/|x|$.)
- We have $\|AB - CR\|_2 = \mathcal{O}(\|A\|_F \|B\|_F / \sqrt{c})$ for probabilities

$$p_{i,j} \propto \begin{cases} A_{i,j}^2, & \text{if } |A_{i,j}| > T \\ |A_{i,j}|, & \text{otherwise} \end{cases}, q_{i,j} \propto \begin{cases} B_{i,j}^2, & \text{if } |B_{i,j}| > T \\ |B_{i,j}|, & \text{otherwise} \end{cases}$$

---

[11] Mahoney et al., "Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication," SIAM J. Comput' 2006

[12] Drineas et al., "Fast Monte-Carlo algorithms for approximate matrix multiplication," FOCS' 2001.

Sequential Matrix Multiplication [13]

Problem statement: Approximate products of a sequence of non-negative matrices, e.g., ABC, $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, and $C \in \mathbb{R}^{p \times q}$.
Idea: Do Random walks in a graph representation of the input matrices and identify all high-valued entries in non-negative matrix products.

[13] Cohen et al., "Approximating matrix multiplication for pattern recognition tasks," J. Alg' 1999.

Sequential Matrix Multiplication [13]

Problem statement: Approximate products of a sequence of non-negative matrices, e.g., ABC, $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, and $C \in \mathbb{R}^{p \times q}$.
Idea: Do Random walks in a graph representation of the input matrices and identify all high-valued entries in non-negative matrix products.

[13] Cohen et al., "Approximating matrix multiplication for pattern recognition tasks," J. Alg' 1999.