EE 270: Large Scale Matrix Computation and Optimization **Winter 2020**

Lecture 2 — January 9

Lecturer: Mert Pilanci *Scribe: Salil Deshpande*

2.1 Randomized Algorithms

2.1.1 Overview

Randomized algorithms are a class of algorithms that make use of randomness and random numbers to guide their behavior. The performance of a randomized algorithm always behaves as a random variable, and randomized algorithms hope to achieve good performance in the average case.

The uncertainty and approximations employed by randomized algorithms may or may not be acceptable depending on the application. Approximations and errors made by randomized algorithms are often acceptable in machine learning applications, for example, when they do not exceed statistical precision. In some high-fidelity scientific computing applications, on the other hand, randomized algorithms and other approximate methods may be less tolerable. However, even when their results are not satisfactory, randomized algorithms can often be used to produce starting points for more costly exact methods. In cases when exact methods cannot handle large problem sizes, attempting to find solutions to large problems may only be possible by employing randomized algorithms.

To understand the behavior of randomized algorithms, one must first be able to analyze the probabilistic behavior of random variables.

2.1.2 Probability Fundamentals

Let X be a discrete random variable that can take on values $x_1, x_2, ..., x_n$.

Definition 1. Expectation
The **expectation**, or expected value, of X is a measure of its center and is defined as

$$\mathbb{E}\left[X\right] \equiv \sum_{i=1}^{n} x_i \mathbb{P}\left(X = x_i\right).$$

Expectation is a linear operator on random variables. This means that $\forall\ a, b \in \mathbb{R}$, random variables X, Y, $\mathbb{E}\left[aX + bY\right] = a\mathbb{E}\left[X\right] + b\mathbb{E}\left[Y\right]$.

Definition 2. Variance
The **variance** of a random variable is a measure of the extent or width of its distribution and is defined as

$$\begin{aligned}\mathbf{Var}[X] &\equiv \mathbb{E}\left[(X-\mathbb{E}[X])^2\right] = \mathbb{E}\left[X^2\right] - 2\mathbb{E}[X\mathbb{E}[X]] + \mathbb{E}\left[\mathbb{E}[X]^2\right] \\ &= \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2.\end{aligned}$$

The variance is not linear. $\mathbf{Var}[cX] = c^2\mathbf{Var}[X]$. And, for two random variables X, Y,

$$\begin{aligned}\mathbf{Var}[X+Y] &= \mathbb{E}[X+Y]^2 - (\mathbb{E}[X+Y])^2 \\ &= \mathbb{E}[X]^2 + 2\mathbb{E}[XY] + \mathbb{E}[Y]^2 - (\mathbb{E}[X]^2 + 2\mathbb{E}[X]\mathbb{E}[Y] + \mathbb{E}[Y]^2) \\ &= (\mathbb{E}[X]^2 - \mathbb{E}[X]^2) + (\mathbb{E}[Y]^2 - \mathbb{E}[Y]^2) + 2(\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]) \\ &= \mathbf{Var}[X] + \mathbf{Var}[Y] + 2(\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y])\end{aligned}$$

Definition 3. Covariance The **covariance** of two random variables X and Y is a measure of variation of their joint probability distribution and is defined as

$$\mathrm{Cov}XY = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y].$$

If X and Y are independent and uncorrelated, then $\mathrm{Cov}XY = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] = 0$. This also means that for independent variables, $\mathbf{Var}[X+Y] = \mathbf{Var}[X] + \mathbf{Var}[Y]$.

This has important implications for the sampling and independent realizations of random variables. If $X_1, X_2, ..., X_n$ are independent and identically distributed random variables, then the variance of their average is

$$\begin{aligned}\mathbf{Var}\left[\frac{X_1 + X_2 + ... + X_n}{n}\right] &= \frac{1}{n^2}\mathbf{Var}[X_1 + X_2 + ... + X_n] = \frac{1}{n^2}\left(\mathbf{Var}[X_1] + \mathbf{Var}[X_2] + ... + \mathbf{Var}[X_n]\right) \\ &= \frac{1}{n}\mathbf{Var}[X_1]\end{aligned}$$

This means that by taking many independent measurements from a single random variable, one could reduce the variance in the average of the measurements taken.

2.1.3 Example: Approximate Counting Algorithm

The counting problem is as follows. Suppose some number of items need to be counted. How does one count them?

Algorithm 1 Deterministic Counting Algorithm

```
$n = 0$
for each item that passes do
    $n = n + 1$
end for
Return $n$
```

The classic deterministic counting algorithm increments a counter for each item that is considered and requires $\mathcal{O}(\log_2 n)$ bits of storage to accommodate the counter variable.

Is it possible to do better? In 1977, Robert Morris of Bell Labs proposed the following algorithm, which provides an approximate count for the number of items only using $\mathcal{O}(\log_2(\log_2 n))$ bits.

Algorithm 2 Approximate Counting Algorithm

```
$n = 0$
**for** each item that passes **do**
    Generate $n$ independent random variables, $X_i \in \{0, 1\}$, $p(X_i = 0) = 0.5$
    **if** All $X_i = 1$ **then**
        $n = n + 1$
    **end if**
**end for**
Return $2^n - 1$
```

This algorithm aims to provide an approximate count by maintaining a counter which captures the order of magnitude of the current count. That is to say that the algorithm does not make a distinction between the numbers 1040 or 1060, but simply captures that both are on the order of 10^{10}, or 1024. By only keep track of the order of magnitude of the count, only the exponent of the current number of items needs to be stored, hence the storage size being $\mathcal{O}(\log_2(\log_2 n))$.

This algorithm makes use of the fact that 2^n items should pass when counting from 2^n up to 2^{n+1} (that is, $2^{n+1} = 2^n + 2^n$). While this algorithm does not explicitly keep track of the 2^n items that must pass by before incrementing its count from n to $n+1$, it maintains a probability of incrementing the count from n to $n+1$ equal to $\frac{1}{2^n}$ for every item considered after the counter reaches n. This means that in expectation, 2^n items pass before the number of items counted (according to the algorithm) is "incremented" from 2^n to 2^{n+1}.

One drawback that this algorithm has is the large variance in the final count. However, as discussed earlier, the variance can be reduced by averaging a count across multiple trials.

2.2 Matrix Multiplication

Matrix multiplication is a fundamental computation in linear algebra and appears in all applications of linear algebra.

Definition 4. Matrix Product
If matrix $A \in \mathbb{R}^{nxd}$ and $B \in \mathbb{R}^{dxp}$, then the **matrix product** $AB \in \mathbb{R}^{nxp}$ is defined such that

$$AB_{ij} = \sum_{k=1}^{d} A_{ik} B_{kj}.$$

2.2.1 Classical Matrix Multiplication

One classical matrix multiplication algorithm makes direct use of the definition of matrix multiplication by sequentially computing the value of each entry of the new matrix.

Algorithm 3 Classical Matrix Multiplication Algorithm - Inner Product Method

```
for $i = 1 : n$ do
    for $j = 1 : p$ do
        $AB_{ij} = 0$
        for $k = 1 : d$ do
            $AB_{ij} + = A_{ik} B_{kj}$
        end for
    end for
end for
```

Another classical algorithm for matrix multiplication comes from the observation that the product of two matrices can be expressed as the a sum of the outer products of the columns of the left matrix and the rows of the right matrix. That is,

$$AB = \sum_{k=1}^{d} A^{(k)} B_{(k)}.$$

Algorithm 4 Classical Matrix Multiplication Algorithm - Inner Product Method

```
$AB = 0, AB \in \mathbb{R}^{nxp}$
for $k = 1 : d$ do
    $AB = AB + A^{(k)} B_{(k)}$
end for
```

Both formulations of classical matrix multiplication are $\mathcal{O}(ndp)$.

2.2.2 Approximate Matrix Multiplication

The outer product formulation of the matrix product offers a natural application for which to apply randomized algorithms. Instead of taking the sum over all d outer products, one could take a sum over some m sampled outer products:

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

Suppose that the d outer products, defined by the pairs of d columns of A and rows of B, are sampled independently and with replacement. And, for each of the m elements of the sum, let p_k be the probability that outer product $A^{(k)} B_{(k)}$ is picked. Then $\forall t$, $\mathbb{P}\left(i_t = k\right) = p_k$.

When sampling in such a manner, a normalization constant must be added to the sum and the approximate matrix multiplication becomes

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

To see why the new normalization constant is needed, consider the expectation of the new approximation.

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{m} \frac{1}{mp_{i_t}} A^{(i_t)} B_{(i_t)}\right] &= \sum_{t=1}^{m} \frac{1}{m} \mathbb{E}\left[\frac{1}{p_{i_t}} A^{(i_t)} B_{(i_t)}\right] = \sum_{t=1}^{m} \frac{1}{m}\left[\sum_{k=1}^{d} \mathbb{P}\left(i_t = k\right) \frac{1}{p_k} A^{(k)} B_{(k)}\right] \\
&= \sum_{t=1}^{m} \frac{1}{m}\left[\sum_{k=1}^{d} p_k \frac{1}{p_k} A^{(k)} B_{(k)}\right] = \sum_{t=1}^{m} \frac{1}{m}\left[\sum_{k=1}^{d} A^{(k)} B_{(k)}\right] \\
&= \sum_{t=1}^{m} \frac{1}{m}\left[AB\right] \\
&= AB
\end{aligned}$$

Note that without the normalization constant $\frac{1}{mp_{i_t}}$, the expectation of this approximation would be incorrect.

This sampled sum of outer products can be seen at a matrix product in its own right. In particular, the normalized, sampled columns of A can be arranged to be the columns of a new matrix C and the normalized, sampled rows of B can be arranged to be the rows of a new matrix R, such that the product $AB \approx CR$. In particular, C and R are of the form

$$C = \begin{bmatrix} \sqrt{\frac{1}{mp_{i_1}}} A^{(i_1)} & \dots & \sqrt{\frac{1}{mp_{i_m}}} A^{(i_m)} \end{bmatrix}$$

$$R = \begin{bmatrix} \sqrt{\frac{1}{mp_{i_1}}} B_{(i_1)} \\ \vdots \\ \sqrt{\frac{1}{mp_{i_m}}} B_{(i_m)} \end{bmatrix}.$$

This form makes it clear that with this approximation, the cost to calculate the new matrix product $CR \approx AB$ is $\mathcal{O}(nmp)$, reduced down from $\mathcal{O}(ndp)$.

Uniform Sampling

The choice of how to sample and decide on the probabilities p_k are crucial in determining the variance of the algorithm. One choice may be to be sample the outer products uniformly at random. That is, let the probability $p_k = \frac{1}{d}\ \forall k$. Then, the normalizing constant $\frac{1}{mp_{i_t}}$ becomes $\frac{d}{m}$ and the matrices C and R become

$$C = \begin{bmatrix} \sqrt{\frac{d}{m}} A^{(i_1)} & \dots & \sqrt{\frac{d}{m}} A^{(i_m)} \end{bmatrix}$$

$$R = \begin{bmatrix} \sqrt{\frac{d}{m}} B_{(i_1)} \\ \vdots \\ \sqrt{\frac{d}{m}} B_{(i_m)} \end{bmatrix}.$$

To understand how effective this choice of sampling probabilities is, consider the variance and error of the calculation. The variance of any particular element CR_{ij} is

$$\mathbf{Var}[CR_{ij}] = \frac{1}{m}\sum_{k=1}^{d} \frac{{A_{ik}}^2 {B_{kj}}^2}{p_k} - \frac{1}{m}(AB_{ij})^2.$$

One way to define the error in the calculation might be to take the sum of the squares of the element-by-element difference between AB and CR. That would be exactly

$$\sum_{i,j}(AB - CR)_{ij}^2 = \|AB - CR\|_F.$$

The expectation of that error would be

$$\begin{aligned} \mathbb{E}\left[\|AB - CR\|_F^2\right] &= \sum_{i,j}\mathbb{E}\left[(AB - CR)_{ij}^2\right] = \sum_{i,j}\mathbb{E}\left[(AB_{ij} - CR_{ij})^2\right] \\ &= \sum_{i,j}\mathbb{E}\left[(\mathbb{E}\left[CR_{ij}\right] - CR_{ij})^2\right] = \sum_{i,j}\mathbf{Var}[CR_{ij}] \\ &= \sum_{i,j}\left(\frac{1}{m}\sum_{k=1}^{d}\frac{{A_{ik}}^2{B_{kj}}^2}{p_k} - \frac{1}{m}(AB)_{ij}^2\right) \\ &= \frac{1}{m}\sum_{k=1}^{d}\frac{\sum_i {A_{ik}}^2 \sum_j {B_{kj}}^2}{p_k} - \frac{1}{m}\sum_{ij}(AB)_{ij}^2 \\ &= \frac{1}{m}\sum_{k=1}^{d}\frac{\|A^{(k)}\|_2^2\|B_{(k)}\|_2^2}{p_k} - \frac{1}{m}\|AB\|_F^2 \end{aligned}$$

Applying the uniform sampling probabilities to the error gives

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

Importance Sampling

Instead of sampling the outer products uniformly at random, we can instead seek to find the sampling distribution that minimizes the chosen error. That is, we hope to solve the following optimization problem:

$$\begin{aligned} \textbf{minimize} \quad & \mathbb{E}\left[\|AB - CR\|_F^2\right] \\ \textbf{subject to} \quad & \sum_{k=1}^{d} p_k = 1. \end{aligned}$$

This can be solved using the method of Lagrange multipliers. Define $q_k \equiv \frac{\|A^{(k)}\|_2 \|B_{(k)}\|_2}{\sqrt{m}}$ and begin as

$$\mathcal{L}(p_1, p_2, ..., p_d, \lambda) = \sum_{k=1}^{d} \frac{q_k^2}{p_k} - \frac{1}{m}\|AB\|_F^2 + \lambda\left(p_1 + p_2 + ... + p_d - 1\right)$$

$$\nabla \mathcal{L} = \left(-\frac{q_1^2}{p_1^2} + \lambda, -\frac{q_2^2}{p_2^2} + \lambda, ..., -\frac{q_d^2}{p_d^2} + \lambda, p_1 + p_2 + ... + p_d - 1\right).$$

The solution to these equations occurs when

$$\frac{q_1^2}{p_1^2} = \frac{q_2^2}{p_2^2} = ... = \frac{q_d^2}{p_d^2},$$

which implies that

$$\frac{q_1}{p_1} = \frac{q_2}{p_2} = ... = \frac{q_d}{p_d}$$

$$q_k = \sqrt{\lambda} p_k \quad \forall k.$$

And finally the constraint $\sum_{k=1}^{d} p_k = 1$ requires that

$$\sum_{k=1}^{d} q_k = \sqrt{\lambda},$$

$$p_k = \frac{q_k}{\sum_{k=1}^{d} q_k} = \frac{\|A^{(k)}\|_2 \|B_{(k)}\|_2}{\sum_{k=1}^{d} \|A^{(k)}\|_2 \|B_{(k)}\|_2}.$$

These probabilities are optimal in minimizing the error $\mathbb{E}\left[\|AB - CR\|_F^2\right]$ and lead to minimal error

$$\mathbb{E}_{min}\left[\|AB - CR\|_F^2\right] = \frac{1}{m}\left(\sum_{k=1}^{d} \|A^{(k)}\|_2 \|B_{(k)}\|_2\right)^2 - \frac{1}{m}\|AB\|_F^2.$$

Probabilistic Bounds On Error

Although we now have an understanding of the expectation of this error, we have not fully captured how it may be distributed. In order to do this, we can use Markov's Inequality.

Theorem 1. Markov's Inequality
Suppose Z be a non-negative random variable and $t > 0$. Then, $\mathbb{P}\left(Z > t\right) \leq \frac{\mathbb{E}[Z]}{t}$.

Applied to this case, we can say that

$$\mathbb{P}\left(\|AB - CR\|_F^2 > \epsilon \|A\|_F^2 \|B\|_F^2\right) \leq \frac{\mathbb{E}\left[\|AB - CR\|_F^2\right]}{\epsilon \|A\|_F^2 \|B\|_F^2}.$$

And, one bound that can be placed on the expectation $\mathbb{E}\left[\|AB - CR\|_F^2\right]$ is

$$\begin{aligned}
\mathbb{E}\left[\|AB - CR\|_F^2\right] &= \frac{1}{m}\left(\sum_{k=1}^{d}\|A^{(k)}\|_2\|B_{(k)}\|_2\right)^2 - \frac{1}{m}\|AB\|_F^2 \\
&\leq \frac{1}{m}\left(\sum_{k=1}^{d}\|A^{(k)}\|_2\|B_{(k)}\|_2\right)^2 \\
&\leq \frac{1}{m}\left(\sqrt{\sum_{k=1}^{d}\|A^{(k)}\|_2^2}\sqrt{\sum_{k=1}^{d}\|B_{(k)}\|_2^2}\right)^2 \\
&\leq \frac{1}{m}\left(\sqrt{\|A\|_F^2}\sqrt{\|B\|_F^2}\right)^2 = \frac{\|A\|_F^2\|B\|_F^2}{m}.
\end{aligned}$$

It follows that

$$\mathbb{P}\left(\|AB - CR\|_F^2 > \epsilon\|A\|_F^2\|B\|_F^2\right) \leq \frac{\mathbb{E}\left[\|AB - CR\|_F^2\right]}{\epsilon\|A\|_F^2\|B\|_F^2} \leq \frac{1}{m\epsilon^2}.$$

This result allows us to provide a confidence on the error bounds of approximate matrix multiplication. It also allows us to understand the number of outer products that should be sampled, m, in order to have an approximate matrix multiplication that has error $\leq \epsilon$ with probability $1 - \delta$. That is,

$$\mathbb{P}\left(\|AB - CR\|_F^2 > \epsilon\|A\|_F^2\|B\|_F^2\right) \leq \delta \quad \text{if} \quad m \geq \frac{1}{\delta\epsilon^2}.$$