EE 381V: Large Scale Learning **Fall 2012**

Lecture 25 — April 18

Lecturer: Caramanis & Sanghavi *Scribe: Kai-Yang Chiang*

25.1 Review of Streaming Model

Streaming model is a new model for presenting massive data. In this model, we consider data (input as matrices) are too large to fit in memory or even in the local disk. Thus, the *streaming algorithms* are designed to handle the data presenting in streaming manner. There are two settings for presenting streaming data:

- Streaming data are given in sequential way. One can regard this setting as reading data stored in "tape".
- Data are given with arbitrary order, and data possibly cannot be stored anywhere.

We will consider the second streaming model with matrices as input. For matrices, there are two reasonable settings for presenting data in each time:

- Turnstyle: each time we are given an entry (i, j) of the matrix, or
- Every time we are given a column or a row.

25.2 Last Time: Matrix Multiplication in Streaming Model

Recall the last time we introduce the randomized matrix multiplication problem (with approximation sense) in streaming model. The problem is as follows:

- Given two $p \times n$ matrices A, B ($p \gg n$ and both are large), output an approximation $P \approx A^T B$ in the sense that:

$$\Pr(\|P - A^T B\|_F \leq \epsilon \|A\|_F \|B\|_F) \geq 1 - \delta. \tag{25.1}$$

The idea of solving this problem under streaming model is using *sketching* to generate matrices $S^T A$, $S^T B$ with different S, where S is a $p \times m$ sign matrix (± 1 in each entry) with only few m columns (that is, $p, n \gg m$). One then calculates several approximate resulting matrices $(S^T A)^T (S^T B)$, and picks one "good" approximation from them. Since matrices $S^T A$, $S^T B$ is now much smaller, they can be stored in memory. We proved this algorithm solves the randomized matrix multiplication problem. Moreover, the smaller dimension m in S depends on rank of A, $\log \frac{1}{\delta}$ and $\log \frac{1}{\epsilon}$.

We have already seen under sketching, the memory it requires is $O(m \log \frac{1}{\epsilon^2} \log \frac{1}{\delta})$. This gives this problem an upper bound of memory use. Today we will see how to derive a space *lower bound* of the matrix multiplication problem using communication complexity.

25.3 Recap of Communication Complexity

25.3.1 Basic Definition

- Communication Complexity (CC):
  Recall that in one-way communication model, Alice and Bob both have some information x and $y \in [0,1]^n$. Given a function $f : [0,1]^n \times [0,1]^n \to \{0,1\}$, the definition of communication complexity is the minimum number of bits that Alice requires to send to Bob, such that Bob can answer $f(x,y)$ correctly.

- Randomized Communication Complexity (RCC):
  In randomized one-way communication model, we are interested in how many bits that Alice requires to send to Bob, such that Bob can answer $f(x,y)$ correctly with probability at least $1-\delta$. Alice and Bob are allowed to do randomization based on a same random generator. Apparently, the complexity depends on how high the probability is. For example, if $\delta = \frac{1}{2}$, the randomized communication complexity is always 0, since Bob can always achieve success probability $\frac{1}{2}$ by flipping a random coin. Thus, we are always interested in RCC with $\delta < \frac{1}{2}$.

25.3.2 Facts of Communication Complexity

Here we list some of the known results for randomized communication complexity.

Theorem 25.1. *Under one-way randomized communication model, the RCC for computing $f(x,y)$ with probability $\geq \alpha$ is $C(n,\alpha)$.*

The above theorem says the RCC of a problem is a function of how high the success probability is, and the length of input string x, y.

Theorem 25.2. *Let Q be a randomized problem defined as below: given input x, y in any order, we want to compute correct $f(x,y)$ with probability at least α. In addition, consider under randomized communication model, if Alice is given x and Bob is given y, the RCC that Bob can answer $f(x,y)$ correctly with probability $\geq \alpha$ is R. Then, any algorithm that can solve the problem Q requires storage space at least $\Omega(R)$.*

Proof: The above theorem can be seen by simple reduction. Suppose an algorithm $A(x, y)$ can solve problem Q. This means A will succeed to give the answer with probability $\geq \alpha$ in *any order of input* x, y. Now consider a scenario that Alice firstly given algorithm A with input x, and pass the whole state of the algorithm to Bob. The communication cost is R by definition of RCC. Bob then continue to input y into A, and output the answer of $A(x, y)$. Thus, such algorithm must need at least R bits of storage under this particular scenario. Notice that A might need more than R storage in general since A is an algorithm that can answer $f(x, y)$ in *any order of input*, while in this scenario the input order is first x then y. Thus, $\Omega(R)$ is the space lower bound for any algorithm that solves problem Q. □

25.4 Lower Bound of Storage on Matrix Multiplication Problem

We will prove the space lower bound of the matrix multiplication in streaming model using Theorem 25.2, by reduction to augmented indexing problem. The main result we will prove is stated as follows.

Theorem 25.3. *Any algorithm that solves randomized matrix multiplication problem with probability at least* $\frac{4}{5}$ *requires space complexity at least* $\Omega(c\epsilon^{-2} \log nc)$*, where* c *is a constant such that every entry in* A, B *can be stored in* $O(\log nc)$ *bits.*

25.4.1 Augmented Indexing Problem

The problem is defined as follows. Alice is given a string $x \in [0, 1]^n$, and Bob is given an index $\hat{i} \in \{1 \dots n\}$ and $x_{\hat{i}+1}, \dots x_n$. The goal for Bob is to answer what the $x_{\hat{i}}$ is (with high probability). In other word, $f(x, y) = x_{\hat{i}}$. Notice that Alice does not know what index $\hat{i}$ is. The following result states the RCC of the augmented index problem.

Theorem 25.4. *Consider augmented index problem under randomized one-way communication model. If we want that Bob answer the correct bit of* $x_{\hat{i}}$ *with probability at least* $\frac{2}{3}$*, then the randomized communication complexity is* $\Omega(n)$*.*

In other word, if we ask Bob to give the correct bit of $x_{\hat{i}}$ with probability $\frac{2}{3}$, then there is no "clever" way to do so; Alice basically needs to send all string information to Bob to achieve the goal.

25.4.2 Preparation: Constructing Matrices

Using reduction technique, we now show that if we have a black box that solves matrix multiplication problem, then we can use it to solve augmented index problem in one-way communication model. Thus, as Theorem 25.2 states, the RCC for augmented index problem will be the lower bound of space required by any matrix multiplication algorithm.

Suppose we are given a black box that solves matrix multiplication problem, with equation (25.1) hold. Now consider the randomized communication model. Alice has a string $x \in [0,1]^{cr/2}$. Bob is given an index $\hat{i} \in \{1 \ldots \frac{cr}{2}\}$, and all values of $x_{\hat{i}+1} \ldots x_{cr/2}$. We will specify the value of r later.

For Alice, she constructs a $\frac{c}{2} \times n$ matrix U. U can be divided into $\log cn + 1$ submatrices. The first $\log cn$ submatrices are $U_0, U_1 \ldots U_{\log cn - 1}$ with each dimension $\frac{c}{2} \times \frac{r}{\log cn}$. The last submatrix Z is an all zero matrix with dimension $\frac{c}{2} \times (n-r)$. Thus the structure of U is:

$$U = \left[U_0 \mid U_1 \mid \cdots \mid U_{\log cn - 1} \mid Z \right].$$

Alice fills in U with x in column-first order. That is, she fills first column of U_0, second column of U_0, ..., last column of U_0, and the first column of U_1, and so on. When she fills in an entry in U_k, if the corresponding entry in $x = 1$, then she fill the entry with 10^k, and if that entry in $x = -1$ she fill the entry with -10^k. Thus, formally, the $U_k(i,j)$ can be defined as follows:

$$U_k(i,j) = \begin{cases} 10^k, & \text{if } x_t = 1, \text{where } t = k \times \frac{r}{\log cn} + (j-1) \times \frac{c}{2} + i \\ -10^k, & \text{if } x_t = 0. \end{cases}$$

On the other hand, Bob will create two matrices, V and B. Suppose the index $\hat{i}$ that Bob is given corresponds to the (i^*, j^*) entry in U_{k^*} submatrix of U. Bob creates the $\frac{cr}{2} \times n$ matrix V as follows. For all entries after the (i^*, j^*) entry in k^*-th block, $V_k(i,j) = -U_k(i,j)$, and for all other entries, $V_k(i,j) = 0$. Thus V has the structure as:

$$V = \left[\begin{array}{c|c|cccccc|c|c|c} & & 0 & \cdots & 0 & -u & \cdots & -u & & & \\ 0 & \cdots & 0 & \cdots & 0^{*} & -u & \cdots & -u & \cdots & -U_{\log cn-1} & Z \\ & & 0 & \cdots & -u & -u & \cdots & -u & & & \end{array}\right],$$

where the index of 0^* is the (i^*, j^*) entry of the matrix U_{k^*}. Bob will also create another $\frac{c}{2} \times n$ matrix B defined as follows:

$$B_{ij} = \begin{cases} 1, & \text{if } i = (k^*-1) \times \frac{r}{\log cn} + j^* \text{ and } j = 1, \\ 0, & \text{otherwise}, \end{cases}$$

So B only contains one entry $= 1$ and all other entries are zero.

In summary, Alice creates V using string x, and Bob creates U and B using given index $\hat{i}$ and $x_{\hat{i}+1} \cdots x_{cr/2}$. Now we show that under this setting, how to use matrix multiplication black box to solve the augmented index problem under communication model.

25.4.3 The Proof of Space Lower Bound for Matrix Multiplication

In the previous subsection we see Alice creates V and Bob creates U, B. To solve augmented index problem in communication model, Alice first send V to Bob. Bob compute $A^T = U + V$, and input A, B into matrix multiplication black box, which will output a matrix P such that $\|P - A^T B\|_F \geq \epsilon \|A\|_F \|B\|_F$ with probability $1 - \delta$.

By definition of A, we have:

$$\|A\|_F^2 \leq \sum_{k=0}^{k^*} \|U_k\|_F^2 = \frac{c}{2} \times \frac{r}{\log cn} \times \sum_{k=0}^{k^*} 100^k,$$

and

$$\|B\|_F^2 = 1.$$

Choosing $r = \frac{\log cn}{8\delta}$, then with probability $1 - \delta$, the resulting matrix P output from matrix multiplication black box satisfies:

$$\begin{aligned} \|P - A^T B\|_F^2 &\leq \epsilon^2 \|A\|_F^2 \|B\|_F^2 \\ &\leq \frac{c}{2} \times 100^{k^*} \times \frac{25}{198}. \end{aligned} \tag{25.2}$$

By our construction, the matrix $A^T B$ has the following structure. The first column of it is the j^*-th column in U^{k^*}, and all other columns are zeros. Since P and $A^T B$ has difference (which is small with high probability), there will be some sign disagreement between $A^T B_{i1}$ and P_{i1} for all i. However, any sign disagreement between $A^T B_{i1}$ and P_{i1} will contribute squared error $(A^T B_{i1} - P_{i1})^2 \geq 100^{k^*}$. This implies if the bound (25.2) holds, then the fraction of entries in $A^T B_{:1}$ and $P_{:1}$ that have sign disagreement will no more than $\frac{25}{198}$, which implies:

$$\Pr(\mathrm{sgn}(P_{i^*1}) = \mathrm{sgn}(U_{k^*}(i^*, j^*))) \geq \frac{173}{198}.$$

Finally, if we choose δ as $\frac{1}{5}$, then with probability at least $\frac{4}{5}$ the matrix multiplication black box outputs P satisfies (25.2), which further implies that Bob will answer $x_{\hat{i}}$ (by looking the sign of P_{i^*1}) correctly with probability at least $\frac{173}{198}$. Putting these together, we conclude that by using this matrix multiplication box,

$$\Pr(\text{Bob correctly answer the bit } x_{\hat{i}}) = \frac{4}{5} \times \frac{173}{198} > 0.69 > \frac{2}{3}. \tag{25.3}$$

Therefore, from Theorem 25.4, the communication complexity for this augmented index problem is $\Omega(\frac{cr}{2})$. By Theorem 25.2, this implies the space lower bound for the matrix multiplication problem is $\Omega(\frac{cr}{2}) = \Omega(c\epsilon^{-2} \log nc)$. The Theorem 25.3 is thus proved.

25.5 Sketching for Regression

The regression problem can also be approximately solved using sketching. That is, instead of solving:

$$\min_X \|AX - B\|_F^2, \tag{25.4}$$

we solve the problem:

$$\min_X \|S^T AX - S^T B\|_F^2, \tag{25.5}$$

with some random sign matrix S with a small dimension. There is a similar result proving that using sketching, we can obtain a good approximation $\tilde{X}$ to the true solution X^*. The result is summarized as follows.

Theorem 25.5. *Let A, B be two $p \times n$ matrices with rank$(A) = k$, and let S be a $p \times m$ random sign matrix, where the small dimension $m = O(k(\log \frac{1}{\delta})/\epsilon)$. Then using sketching to solve* (25.5) *instead of* (25.4), *with probability $1-\delta$ we have $\|A\tilde{X} - B\| \leq (1+\epsilon)\|AX^* - B\|$.*

Reference

Kenneth L. Clarkson and David P. Woodruff, "Numerical Linear Algebra in the Streaming Model", in proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC), 2009.