CS 15-859: Algorithms for Big Data	Fall 2022

Lecture lecture 10 — 11/17/2022

Prof. David Woodruff — *Scribe: Shilin Ma*

1 Fano's Inequality

Theorem 1 (Fano's Inequality)**.** *For any estimate $X' : X \to Y \to X'$ with $P_e = \mathbf{Pr}[X' \neq X]$, we have $H(X|Y) \leq H(P_e) + P_e \cdot \log(|X| - 1)$*

Here, $X \to Y \to X'$ is a *Markov Chain*, meaning X' and X are independent given Y. One way to understand this is is to say "Past and future are conditionally independent given the present".

To prove Fano's Inequality, we first need the *Data Processing Inequality.*

Proposition 1 (Data Processing Inequality)**.** Suppose $X \to Y \to Z$ is a Markov Chian, then $I(X;Y) \geq I(X;Z)$.

Again, we can express this idea as "no clever combination of the data can improve the estimation".

Data Processing Inequality. Since, $I(X;Y) = H(X) - H(X|Y)$ and conditioning cannot increase entropy, we know that mutual information is non-negative. Recall the Chain rule of mutual information, $I(X;Y,Z) = I(X;Z) + I(X;Y|Z) = I(X;Y) + I(X;Z|Y)$. If we can show that $I(X;Z|Y) = 0$, then $I(X;Z) + I(X;Y|Z) = I(X;Y)$ and $I(X;Z) \leq I(X;Y)$.

$I(X;Z|Y) = H(X|Y) - H(X|Y,Z)$. Given Y, X and Z are independent, $H(X|Y,Z) = H(X|Y)$. Thus, $I(X;Z|Y) = 0$. ■

Data Processing Inequality implies the following:

$$\begin{aligned} I(X;Y) &\geq I(X;Z) \\ H(X) - H(X|Y) &\geq H(X) - H(X|Z) \\ -H(X|Y) &\geq -H(X|Z) \\ H(X|Y) &\leq H(X|Z) \end{aligned}$$

Now, we are ready to prove Fano's Inequality,

Proof of Fano's Inequality. Let $E = 1$ if X' is not equal to X, and $E = 0$ otherwise.

Using the Chain rule of Entropy, $H(E, X|X') = H(X|X') + H(E|X, X')$. $H(E|X, X') = 0$ since there is no uncertain about E if both X and X' are known. So, $H(E, X|X') = H(X|X')$.

On the other hand, $H(E, X|X') = H(E|X') + H(X|E, X') \leq H(P_e) + H(X|E, X')$ where we use $H(E|X') \leq H(E) = H(P_e)$.

$$\begin{aligned} H(X|E, X') &= \mathbf{Pr}[E = 0]H(X|X', E = 0) + \mathbf{Pr}[E = 1]H(X|X', E = 1) \\ &= (1 - P_e) \cdot 0 + P_e \cdot H(X|X', E = 1) \\ &\leq (1 - P_e) \cdot 0 + P_e \cdot \log_2(|X| - 1) \\ &\quad \text{(Since } X' \neq X\text{, the worst case is when } X' \text{ is uniform over } |X| - 1 \text{ possibilities)} \end{aligned}$$

Combining these, we have $H(X|X') \leq H(P_e) + P_e \cdot \log_2(|X| - 1)$. By Data Processing, $H(X|Y) \leq H(X|X') \leq H(P_e) + P_e \cdot \log_2(|X| - 1)$. ■

Proposition 2. Fano's Inequality is tight.

Proof. Suppose the distribution p of X satisfies $p_1 \geq p_2 \geq \cdots \geq p_n$. Suppose additionally that Y is a constant. Therefore, $I(X; Y) = H(X) - H(X|Y) = 0$, and the best predictor X' of X is $X = 1$.

$P_e = \mathbf{Pr}[X' \neq X] = 1 - p_1$. Fano's inequality gives $H(X|Y) \leq H(1 - p_1) + (1 - p_1)\log_2(n - 1)$. Recall that $H(X) = H(X|Y)$. Let $p_2 = p_3 = \cdots = p_n = \frac{1-p_1}{n-1}$. Then,

$$\begin{aligned} H(X) &= \sum_{i=1}^{n} p_i \log(\frac{1}{p_i}) \\ &= p_1 \log(\frac{1}{p_1}) + \sum_{i=2}^{n} \frac{1 - p_1}{n - 1} \log(\frac{n - 1}{1 - p_1}) \\ &= p_1 \log(\frac{1}{p_1}) + (1 - p_1)\log(\frac{1}{1 - p_1}) + (1 - p_1)\log(n - 1) \\ &= H(p_1) + (1 - p_1)\log(n - 1) \end{aligned}$$

So, $H(X|Y) = H(p_1) + (1 - p_1)\log(n - 1)$, and Fano's Inequality is tight. ■

2 Communication Lower Bounds

2.1 Randomized 1-Way Communication Complexity

First, we introduce the **Index Problem**. Alice has $x \in \{0, 1\}^n$ and Bob has $j \in \{1, 2, 3, \cdots, n\}$. Alice sends a single message M to Bob. Bob, given M and j should output x_j with probability at least 2/3 (over the coin tosses, not the inputs). We can prove, using Fano's Inequality, that for some inputs and coin tosses, M must be $\Omega(n)$ bits long.

Consider a uniform distribution μ on X. We can think of Bob's output as an estimate X'_j to X_j. For all j, $\mathbf{Pr}[X'_j = X_j] \geq 2/3$, and $X_j \to M \to X'_j$ is a Markov Chain. By Fano's Inequality, for all j, $H(X_j|M) \leq H(\frac{1}{3}) + \frac{1}{3}(\log_2(2) - 1) = H(\frac{1}{3})$.

Consider the mutual information $I(M; X)$. By the chain rule, $I(X; M) = \sum_i I(X_i; M|X_{<i}) = \sum_i H(X_i|X_{<i}) - H(X_i|M, X_{<i})$. Since the coordinates of X are independent bits, $H(X_i|X_{<i}) = H(X_i) = 1$, and because conditioning cannot increase entropy $H(X_i|M, X_{<i}) \leq H(X_i|M)$.

So, $I(X; M) \geq n - \sum_i H(X_i|M) \geq n - H(\frac{1}{3})n$, and we can conclude that

$$|M| \geq H(M) \geq I(X; M) = \Omega(n)$$

2.2 Typical Communication Reduction

Alice has $a \in \{0,1\}^n$ and creates stream $s(a)$. Bob has $b \in \{0,1\}^n$ and creates stream $s(b)$.

The **Lower Bound Technique** has three steps:

1. Run the streaming algorithm on $s(a)$, transmit state of $alg(s(a))$ to Bob.
2. Bob computes $alg(s(a), s(b))$
3. If Bob solves $g(a,b)$, then the space complexity of the algorithm is at least the 1-way communication complexity of g.

Here are a couple of examples of applying this reduction technique with the Index Problem.

2.2.1 Distinct Elements

Suppose we have an algorithm that answers the question: Given $a_1, \cdots, a_m$ in $[n]$, how many distinct numbers are there?

For the index problem, Alice has a bit string x in $\{0,1\}^n$, Bob has an index i in $[n]$, and Bob wants to know if $x_i = 1$.

So, we perform the following reduction: Let $s(a) = i_1, \cdots, i_r$ where i_j appears if and only if $x_{i_j} = 1$. For example, if $x = 01101$, then $s(a) = 2, 3, 5$. Let $s(b) = i$. If $alg(s(a), s(b)) = alg(s(a)) + 1$, then Bob guesses $x_i = 0$, otherwise $x_i = 1$.

If the algorithm returns correctly with constant probability, this procedure solves the Index problem.

Therefore, the space complexity of the algorithm is at least the 1-way communication complexity of the Index Problem.

2.2.2 Rank of Matrix

Suppose we have an algorithm that returns the rank of a matrix. Use the reduction technique to give an $\Omega(n)$ space complexity lower bound.

Let $s(a)$ be the diagonal matrix where $s(a)_{ii} = x_i$ and $s(b)$ be the matrix where $s(b)_{ii} = 1$ and every other entry is 0. If $alg(s(a) + s(b)) = alg(s(a)) + 1$, then Bob decides $x_i = 0$, otherwise $x_i = 1$.

2.3 Strengthening Index: Augmented Index

In the augmented-index problem, Alice has $x \in \{0,1\}^n$, Bob has $i \in [n]$ and $x_1, \cdots, x_{i-1}$. Bob wants to learn x_i. We can again obtain an $\Omega(n)$ lower bound for this problem.

Just as before, we have $I(M;X) = \sum_i I(M; X_i | X_{<i}) = n - \sum_i H(X_i | M, X_{<i})$. In the previous case, we bounded $H(X_i|M, X_{<i})$ by dropping the conditioning on $X_{<i}$, but now Bob has exactly $X_{<i}$, and the Markov Chain is $X_i \to M, X_1, \cdots, X_{i-1} \to X_i'$. So, we can apply Fano's inequality and get $H(X_i|M, X_{<i}) \leq H(\delta)$ if Bob can predict X_i with probability at least $1 - \delta$ from M,

$X_{<i}$. Putting these together, $I(M;X) = n - \sum_i H(X_i|M, X_{<i}) = n - H(\delta)n = \Omega(n)$. Again, $|M| \geq H(M) \geq I(X;M) = \Omega(n)$.

2.3.1 $\log n$-bit Lower Bound for Estimating Norms

Consider the special case of a vector of length one — a counter. We can use Augmented Index to show that storing a counter that estimates the value up to a constant factor requires $\log(n)$ space.

Alice has $x \in \{0,1\}^{\log n}$ as the input to the Augmented Index problem. She creates a vector v with a single coordinate equal to $\sum_j 10^j x_j$. Alice then sends Bob the state of the data stream algorithm after feeding in the input v.

Bob has $i \in [\log n]$ and $x_{i+1}, x_{i+2}, \cdots, x_{\log n}$ as part of the Augmented Index problem. Bob creates a vector $w = \sum_{j>i} 10^j x_j$ which is the sum of the high order bits. Bob feeds $-w$ into the state of the algorithm. If the output of the streaming algorithm is at least $10^i/2$, Bob guesses $x_i = 1$, otherwise $x_i = 0$.

Note that $\sum_j 10^j X_j - \sum_{j>i} 10^j X_j = \sum_{j=1}^{i} 10^j x_j$. If $x_i = 1$, then $\sum_{j=1}^{i} 10^j x_j \geq 10^i$. If $x_i = 0$, then $\sum_{j=1}^{i} 10^j x_j \leq \frac{10^i}{9}$. So, there is a constant factor gap between the two cases. Therefore, if the algorithm can estimate the norm of the vector up-to a constant factor with high probability, then this procedure can solve the Augmented Index Problem with inputs of $\log n$ bits.

This shows that the streaming algorithm requires at least $\Omega(\log n)$ bits of storage.