Streaming Algorithms

Meng-Tsung Tsai

05/25/2018

References

- "Lecture Notes of Topics in Information Theory in Computer Science," Braverman (2013)
- "Tight Bounds for Graph Problems in Insertion Streams," Sun and Woodruff (2015)

Shannon Entropy

Let $X \sim (\Omega, p)$ be a random variable. The *entropy* of X is defined to be

$$H(X) = E[\log_2 1/p(X)].$$

Let Y be a random variable so that

$$p(Y = 0) = 1/2 \text{ and } p(Y = 1) = 1/2,$$

then $H(Y) = 1$.

What is the semantics of entropy?

Shannon Entropy

Let $X_1, X_2, ..., X_n$ be i.i.d. copies of random variable X. Let $x_1, x_2, ..., x_n$ be the outcomes of $X_1, X_2, ..., X_n$.

Step 1. Alice sends a coded message $f(x_1, x_2, ..., x_n)$ to Bob.

Step 2. Bob recover the outcomes of $X_1, X_2, ..., X_n$ by decoding the message and obtaining $g(f(x_1, x_2, ..., x_n))$.

Goal. The protocol ensures that for some constant $\varepsilon > 0$

$$\Pr[g(f(x_1, x_2, ..., x_n)) \neq (x_1, x_2, ..., x_n)] < \varepsilon, \text{ and}$$

make the coded message as short as possible, **on average**.

$nH(X)+o(n)$ upper bounds the length of the shortest code.
any $(nH(x) - \Omega(n))$-length code has the failure rate $1-o(1)$.

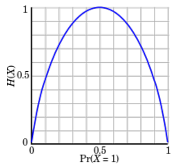
Tail Probabilities of Binomial Distribution

If $k \le n/2$, then $\sum_{0 \le i \le k} C(n, i) \le 2^{nH(k/n)}$.

Let $X_1, X_2, \ldots, X_n$ be the random bit-string **uniformly sampled** from the set of n-bit string with $\le k$ 1's.

Then $H(X_1X_2 \ldots X_n) = \log(\sum_{0 \le i \le k} C(n, i))$. (Why?)

$H(X_1X_2 \ldots X_n) \le H(X_1)+H(X_2)+\ldots+H(X_n) = nH(X_1) \le nH(n/k)$.



Tail Probabilities of Binomial Distribution

$\sum_{0 \le i \le n/4} C(n, i) \le 2^{nH(1/4)}$.

$H(1/4) = 1/4 * \log 4 + 3/4 * \log 4/3 \le 0.82$

$\Rightarrow \sum_{0 \le i \le n/4} C(n, i) \le 2^{0.82n}$

$\Rightarrow \Pr[X \sim B(n, 1/2) \le n/4] \le 2^{\Omega(n)}$

Applications

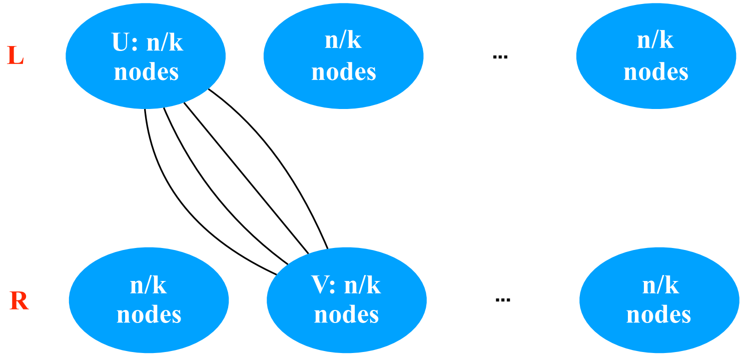
k-EC certificate

Input: a simple undirected graph G

Output: "Yes," if G is k-edge connected; "No," otherwise.

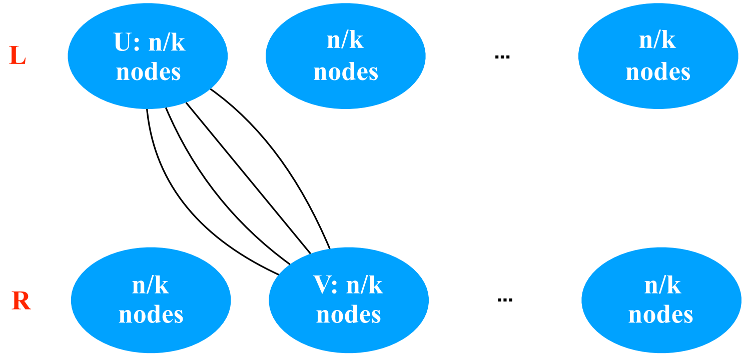
Goal: show that any 1-pass **deterministic** streaming algorithm that decides k-edge-connectivity requires $\Omega(kn \log n)$ bits.

k-EC certificate



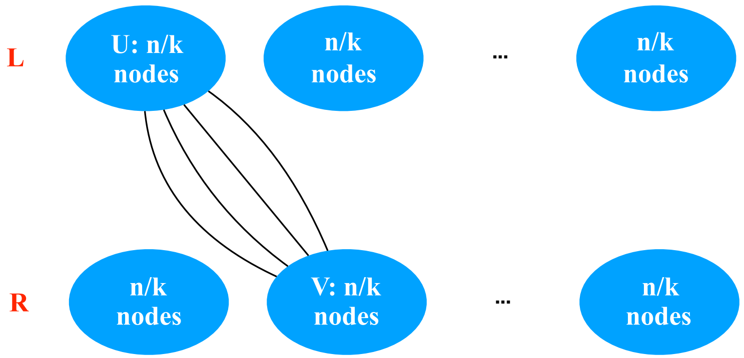
Create a random perfect matching between each pair of U and V for $U \in L$, $V \in R$. The entroy of this random graph S is $\Omega(\log (n/k)!^{(k/2)^{\wedge}2}) = \Omega(kn\log n/k)$.

k-EC certificate



If there exists a **deterministic** streaming algorithm A that can decide k-EC, then Bob can use A to reconstruct the random graph S because A doesn't **err**. (what's A?)

k-EC certificate



Consequently $D^{1\text{-way}}(k\text{-EC}) = \Omega(kn \log (n/k))$.

Permutation

Alice is given a permutation of {1, 2, ..., n}, i.e. $\sigma(1), \sigma(2), \ldots, \sigma(n)$ and represent the permutation as the concatenation of the binary representation of $\sigma(1)\sigma(2)\ldots\sigma(n)$. Hence, the representation of the permutation has n log n bits.

Bob is given an index k in [1, n log n].

Goal: to answer whether the k-th bit is 0 or 1.

$R^{1\text{-way}}(\text{Perm}) = \Omega(n \log n)$. (Is Perm related to Index?)

Exercise 1

Show that any streaming algorithm that decides connectivity requires $\Omega(n \log n)$ bits.

Give an space-optimal streaming algorithm.

Exercise 2

Show that any streaming algorithm that decides bipartiteness requires $\Omega(n \log n)$ bits.

Give an space-optimal streaming algorithm.

Exercise 3

Show that any streaming algorithm that decides cycle-freeness requires $\Omega(n \log n)$ bits.

Give an space-optimal streaming algorithm.