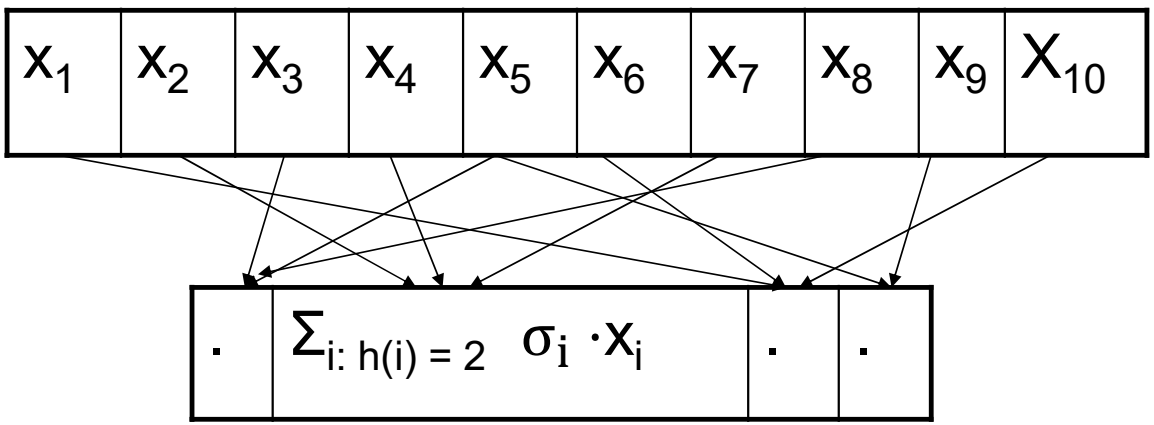
CountSketch achieves the l_2–guarantee

- Assign each coordinate i a random sign $\sigma_i \in \{-1,1\}$

- Randomly partition coordinates into B buckets, maintain $c_j = \Sigma_{i:\, h(i) = j}\, x_i \cdot \sigma_i$ in the j-th bucket



- Estimate x_i as $\sigma_i \cdot c_{h(i)}$

Why Does CountSketch Work?

- $E\left[\sigma_i c_{h(i)}\right] = E[\sigma_i \sum_{i':h(i)=h(i')} \sigma_{i'} x_{i'}] = x_i$
- Suppose we independently repeat this hashing scheme O(log n) times
- Output the median of the estimates across the log n repetitions
- "Noise" in a bucket is $\sigma_i \cdot \sum_{i' \neq i, h(i')=h(i)} \sigma_{i'} \cdot x_{i'}$
- What is the variance of the noise?
- $E\left[\left(\sigma_i \cdot \sum_{i' \neq i, h(i')=h(i)} \sigma_{i'} \cdot x_{i'}\right)^2\right] \leq \frac{|x|_2^2}{B}$
- So with constant probability, the noise in a bucket is $O(\frac{|x|_2}{\sqrt{B}})$ in magnitude
- Since the log n repetitions are independent, this ensures that our estimate $\sigma_i c_{h(i)}$ will equal $x_i \pm O(\frac{|x|_2}{\sqrt{B}})$ with probability 1-1/poly(n)
- Hence, we approximate *every* x_i simultaneously up to additive error $O(\frac{|x|_2}{\sqrt{B}})$

Tail Guarantee

- CountSketch approximates every x_i simultaneously up to additive error $O(\frac{|x|_2}{\sqrt{B}})$
- But what if x_1 is a super large poly(n), and $x_2 = n$ and $x_3 = \ldots = x_n = 1$?
- We get a pretty bad approximation to x_2
- Tail Guarantee: CountSketch approximates *every* x_i simultaneously up to additive error $O(\frac{|x_{-B/4}|_2}{\sqrt{B}})$, where $x_{-B/4}$ is x after zero-ing out its top B/4 coordinates in magnitude
- Proof: with probability at least $3/4$, in each repetition the top B/4 coordinates of x in magnitude do not land in the same hash bucket as x_i
  - Do we need a lot of independence for this?
- What happens if x is B/4-sparse?

Why Care About the ℓ_1-Guarantee?

- l_1 – guarantee
  - output a set containing all items j for which $|x_j| \geq \phi|x|_1$
  - the set should not contain any j with $|x_j| \leq (\phi\text{-}\varepsilon)\ |x|_1$

- l_2 – guarantee
  - output a set containing all items j for which $x_j{}^2 \geq \phi|x|_2^2$
  - the set should not contain any j with $x_j^2 \leq (\phi - \epsilon)|x|_2^2$

- l_2 – guarantee implies the l_1 – guarantee

- So why care about the l_1 – guarantee?

- A nice thing about the l_1-guarantee is that it can be solved deterministically!

Deterministic ℓ_1 Heavy Hitters

- An s x n matrix S is ϵ-incoherent if
  - for all columns S_i, $|S_i|_2 = 1$
  - for all pairs of columns S_i and S_j, $\left|\langle S_i, S_j\rangle\right| \leq \epsilon$
  - entries can be specified with O(log n) bits of space

- Compute $S \cdot x$ in a stream using O(s log n) bits of space

- Estimate $\widehat{x_i} = S_i^T S x$
  - $\widehat{x_i} = \sum_{j=1,\ldots,n} \langle S_i, S_j\rangle x_j = |S_i|_2^2 x_i \pm \max_{i,j} \left|\langle S_i, S_j\rangle\right| |x|_1 = x_i \pm \epsilon |x|_1$
  - Can figure out which $|x_i| \geq \phi |x|_1$ and which $|x_i| \leq (\phi - \epsilon)|x|_1$

- But do ϵ-incoherent matrices exist?

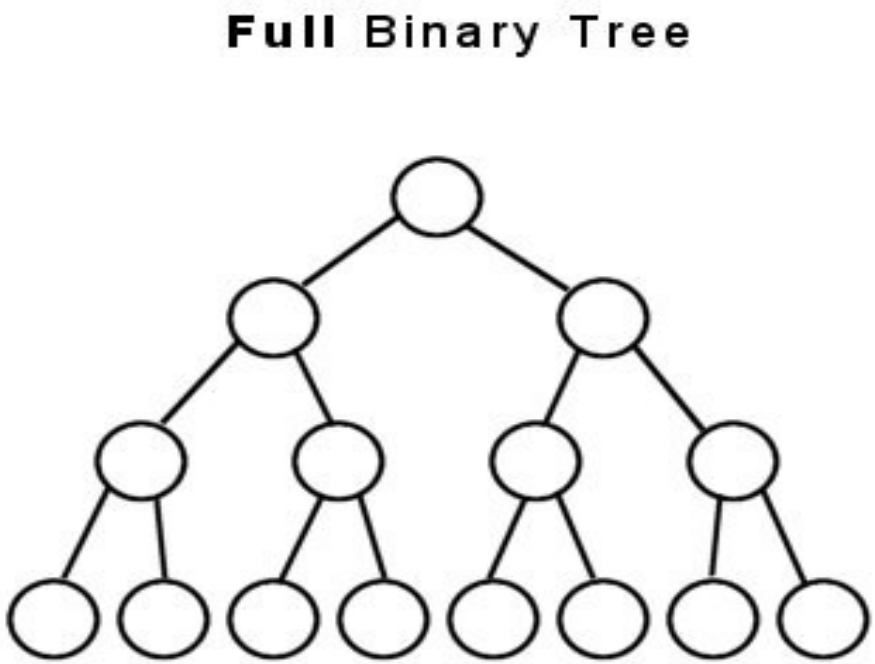
ϵ-Incoherent Matrices

- Consider a prime q = $\Theta((\log n)/\epsilon)$. Let d = $\epsilon \cdot q = \Theta(\log n)$
- Consider n distinct non-zero polynomials $p_1, \ldots, p_n$ each of degree less than d.
  - $q^d - 1 > n$
- Associate p_i with i-th column of S
- Let s = q^2 and group the rows of S into q groups of size q
  - In j-th group, the i-th column has a single non-zero on the $p_i(j)$-th entry
  - $p_{i(j)}$-th entry is equal to $1/q^{1/2}$
- Each column S_i has $|S_i|_2 = 1$
- S_i and S_j each have the same non-zero in the k-th group iff $p_i(k) = p_j(k)$
- Number of such groups k is at most $d \leq \epsilon q$, so $\left|\langle S_i, S_j \rangle\right| \leq \epsilon$

How to Find the Top k Heavy Hitters Quickly

- There are 2^i nodes in i-th level of tree
  - Start at the level with 2k nodes
- Each node corresponds to a subset of [n] of size $n/2^i$ with the same i-bit prefix
- In i-th level, for each i, hash to O(k) buckets repeat O(log n) times. Like CountSketch, but in each bucket we run an approximation algorithm to the 2-norm
- In top level our universe has only 2k nodes, so we find top k just by computing estimate for all of them

Main idea: in next level, we only need to consider the left and right child of each of the k nodes we found at the previous level. So only $2k \ll n$ nodes to consider.



Outline

- Quick recap of ℓ_1-regression, and how to speed it up
- Introduction to the Streaming Model
- Estimating Norms in the Streaming Model
- Heavy Hitters in a Stream
- Estimating Number of Non-Zero Entries (ℓ_0)

Estimating the Number of Non-Zero Entries

- $|x|_0 = |\{i \text{ such that } x_i \neq 0\}|$
- How can we output a number Z with $(1-\epsilon)Z \leq |x|_0 \leq (1+\epsilon)Z$ with prob. 9/10?
  - Want $O((\log n)/\epsilon^2)$ bits of space
- Suppose $|x|_0 = O(\frac{1}{\epsilon^2})$. What can we do in this case?
- Use our algorithm for recovering a k-sparse vector from last time, $k = O\left(\frac{1}{\epsilon^2}\right)$
  - What is another way?
- But what if $|x|_0 \gg \frac{1}{\epsilon^2}$?

Estimating the Number of Non-Zero Entries

- Suppose we somehow had an estimate Z with $Z \leq |x|_0 \leq 2Z$, what could we do?
- Independently sample each coordinate i with probability $p = 100/(Z\,\epsilon^2)$
- Let Y_i be an indicator random variable if coordinate i is sampled
- Let y be the vector restricted to coordinates i for which $Y_i = 1$
- $E[|y|_0] = \sum_{i \text{ such that } x_i \neq 0} E[Y_i] = p|x|_0 \geq \frac{100}{\epsilon^2}$
- $\mathrm{Var}[|y|_0] = \sum_{i \text{ such that } x_i \neq 0} \mathrm{Var}[Y_i] \leq \frac{200}{\epsilon^2}$
- $\Pr\left[\left||y|_0 - E[|y|_0]\right| > \frac{100}{\epsilon}\right] \leq \frac{\mathrm{Var}[|y|_0]\epsilon^2}{100^2} \leq \frac{1}{50}$
- Use sparse recovery or CountSketch to compute $|y|_0$ exactly
- Output $\frac{|y|_0}{p}$

But we don't know Z…

Estimating the Number of Non-Zero Entries

- Guess Z in powers of 2
- Since $0 \leq |x|_0 \leq n$, there are $O(\log n)$ guesses
- The i-th guess Z = 2^i corresponds to sampling each coordinate with probability $p = \min(1, \frac{100}{2^i \epsilon^2})$
- Sample the coordinates as nested subsets $[n] = S_0 \supseteq S_1 \supseteq S_2 \supseteq \cdots S_{\log n}$
- Run previous algorithm for each guess
- One of our guesses Z satisfies $Z \leq |x|_0 \leq 2Z$ and we should use that guess

- *But how do we know which one?*

Estimating the Number of Non-Zero Entries

- Use the largest guess $Z = 2^i$ for which $\frac{400}{\epsilon^2} \leq |y|_0 \leq \frac{3200}{\epsilon^2}$
- If $\frac{800}{\epsilon^2} \leq E[|y|_0] \leq \frac{1600}{\epsilon^2}$, then $\frac{400}{\epsilon^2} \leq |y|_0 \leq \frac{3200}{\epsilon^2}$ with probability $1 - O(\epsilon^2)$
- If $\frac{100}{\epsilon^2} \leq E[|y|_0] \leq \frac{200}{\epsilon^2}$, then $|y|_0 < \frac{400}{\epsilon^2}$ with probability at least $1 - O(\epsilon^2)$
  - Use nested subset property to conclude this also holds for larger i
- So with probability $1 - O(\epsilon^2)$, we choose an i for which $\frac{200}{\epsilon^2} \leq E[|y|_0] \leq \frac{1600}{\epsilon^2}$
- At most 4 such indices i, and all 4 of them satisfy $|y|_0 = (1 \pm \epsilon)E[|y|_0]$ simultaneously with probability 1-4/50. So doesn't matter which i we choose

- Overall, our success probability is 1 - $O(\epsilon^2)$ -4/50 > 4/5

What is Our Overall Space Complexity?

- If we use our k-sparse recovery algorithm for $k = O\left(\frac{1}{\epsilon^2}\right)$, then it takes $O(\frac{\log n}{\epsilon^2})$ bits of space in each of log n levels, so $O(\frac{\log^2 n}{\epsilon^2})$ total bits of space ignoring random bits
  - How much randomness do we need?
  - Pairwise independence is enough for Chebyshev's inequality
  - Implement nested sampling by choosing a hash function $h: [n] \to [n]$, checking if first i bits of h(j) = 0
  - O(log n) bits of space for the randomness

- Can improve to $O\left(\frac{\log n\,(\log\left(\frac{1}{\epsilon}\right)+\log\log n)}{\epsilon^2}\right)$ bits. How?

- Just need to know number of non-zero counters, so reduce counters from log n bits to $O(\log\left(\frac{1}{\epsilon}\right) + \log\log n)$ bits

Reducing Counter Size

- In sampling levels that we care about, we have $O\left(\frac{1}{\epsilon^2}\right)$ counters, each of O(log n) bits
- At most $O\left(\frac{\log n}{\epsilon^2}\right)$ prime numbers dividing any of these counters
- Choose a random prime q = $O\left(\frac{\log n\ (\log\log n + \log(\frac{1}{\epsilon}))}{\epsilon^2}\right)$. Unlikely that q divides any counter
- Just maintain our sparse recovery structure mod q, so $O\left(\frac{(\log\log n + \log\left(\frac{1}{\epsilon}\right)}{\epsilon^2}\right)$ bits per each of O(log n) sparse recovery instances

Outline

1. Information Theory Concepts

2. Distances Between Distributions

3. An Example Communication Lower Bound – Randomized 1-way Communication Complexity of the INDEX problem

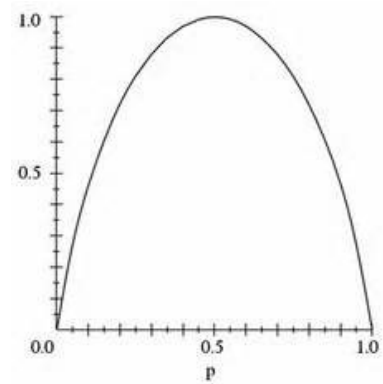
Discrete Distributions

- Consider distributions p over a finite support of size n:
  - p = $(p_1, p_2, p_3, \dots, p_n)$
  - $p_i \in [0,1]$ for all i
  - $\sum_i p_i = 1$
- X is a random variable with distribution p if $\Pr[X = i] = p_i$

Entropy

- Let X be a random variable with distribution p on n items
- (Entropy) $H(X) = \sum_i p_i \log_2 (1/p_i)$
  - If $p_i = 0$ then $p_i \log_2 \left(\frac{1}{p_i}\right) = 0$
  - $H(X) \leq \log_2 n$. Equality holds when $p_i = \frac{1}{n}$ for all i.
  - Entropy measures "uncertainty" of X.
- (Binary Input) If B is a bit with bias p, then

$$\text{H(B)} = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$$



(symmetric)

Conditional and Joint Entropy

- Let X and Y be random variables

- (Conditional Entropy)

$$H(X \mid Y) = \sum_y \mathrm{H}(\mathrm{X} \mid \mathrm{Y} = \mathrm{y})\ \mathrm{Pr}[Y = y]$$

- (Joint Entropy)

$$\mathrm{H}(\mathrm{X}, \mathrm{Y}) = \sum_{x,y} Pr[(X,Y) = (x,y)] \log(1/Pr[(X,Y) = (x,y)])$$

Chain Rule for Entropy

- (Chain Rule) H(X,Y) = H(X) + H(Y | X)

- Proof:

H(X,Y) = $\sum_{x,y} \Pr[(X,Y) = (x,y)] \log\left(\frac{1}{\Pr((X,Y)=(x,y))}\right)$

= $\sum_{x,y} \Pr[X = x\,] \Pr\,[Y = y\,|X = x] \log\left(\frac{1}{\Pr(X=x)\,\Pr(Y=y\,|X=x)}\right)$

= $\sum_{x,y} \Pr[X = x\,] \Pr\,[Y = y\,|X = x](\log\left(\frac{1}{\Pr(X=x)}\right) + \log(\frac{1}{\Pr[\mathrm{Y=y}\,|\mathrm{X=x}]}))$

= H(X) + H(Y | X)

Conditioning Cannot Increase Entropy

- Let X and Y be random variables. Then $\mathrm{H}(\mathrm{X}|\mathrm{Y}) \leq \mathrm{H}(\mathrm{X})$.

- To prove this, we need Jensen's inequality:

  Let f be a continuous, concave function, and let $\mathrm{p}_1, \ldots, \mathrm{p}_\mathrm{n}$ be non-negative reals that sum to 1. For any $\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n}$,

$$\sum_{\mathrm{i}=1,\ldots,\mathrm{n}} \mathrm{p}_\mathrm{i} \mathrm{f}(\mathrm{x}_\mathrm{i}) \leq \mathrm{f}\left(\sum_{\mathrm{i}=1,\ldots,\mathrm{n}} \mathrm{p}_\mathrm{i} \mathrm{x}_\mathrm{i}\right)$$

- Recall that f is concave if $\mathrm{f}\left(\frac{\mathrm{a}+\mathrm{b}}{2}\right) \geq \frac{\mathrm{f}(\mathrm{a})}{2} + \frac{\mathrm{f}(\mathrm{b})}{2}$ and f(x) = log x is concave

Conditioning Cannot Increase Entropy

- Proof:

$$H(X\,|Y) - H(X) = \sum_{xy} \Pr[Y=y]\,\Pr[X=x\,|\,Y=y]\,\log\left(\frac{1}{\Pr[X=x\,|Y=y]}\right)$$

$$- \sum_{x} \Pr[X=x]\,\log\left(\frac{1}{\Pr[X=x]}\right) \sum_{y} \Pr[Y=y\,|\,X=x]$$

$$= \sum_{x,y} \Pr[X=x, Y=y]\,\log\left(\frac{\Pr[X=x]}{\Pr[X=x\,|Y=y]}\right)$$

$$= \sum_{x,y} \Pr[X=x, Y=y]\,\log\left(\frac{\Pr[X=x]\,\Pr[Y=y]}{\Pr[(X,Y)=(x,y)]}\right)$$

$$\leq \log\left(\sum_{x,y} \Pr[X=x, Y=y] \cdot \frac{\Pr[X=x]\,\Pr[Y=y]}{\Pr[(X,Y)=(x,y)]}\right)$$

$$= 0$$

where the inequality follows by Jensen's inequality.

If X and Y are independent H(X | Y) = H(X).

Mutual Information

- (Mutual Information) $I(X ; Y) = H(X) - H(X \mid Y)$

$$= H(Y) - H(Y \mid X)$$

$$= I(Y ; X)$$

Note: $I(X ; X) = H(X) - H(X \mid X) = H(X)$

- (Conditional Mutual Information)

$$I(X ; Y \mid Z) = H(X \mid Z) - H(X \mid Y, Z)$$

Is $I(X ; Y \mid Z) \geq I(X ; Y)$? Or is $I(X ; Y \mid Z) \leq I(X; Y)$? Neither!

Mutual Information

- Claim: For certain X, Y, Z, we can have I(X ; Y | Z) $\leq$ I(X ; Y)

- Consider X = Y = Z

- Then,
  - $\mathrm{I(X\,;Y \mid Z) = H(X \mid Z)\ - H(X \mid Y, Z) = 0\ - 0 = 0}$
  - $\mathrm{I(X\,;Y) = H(X) - H(X \mid Y) = H(X) - 0 = H(X)}$

- Intuitively, Y only reveals information that Z has already revealed, and we are conditioning on Z

Mutual Information

- Claim: For certain X, Y, Z, we can have I(X ; Y | Z) $\geq$ I(X ; Y)

- Consider $\mathrm{X} = \mathrm{Y} + \mathrm{Z} \bmod 2$, where X and Y are uniform in {0,1}

- Then,
  - $\mathrm{I}(\mathrm{X}\,;\mathrm{Y} \mid \mathrm{Z}) = \mathrm{H}(\mathrm{X} \mid \mathrm{Z}) - \mathrm{H}(\mathrm{X} \mid \mathrm{Y}, \mathrm{Z}) = 1 - 0 = 1$
  - $\mathrm{I}(\mathrm{X}\,;\mathrm{Y}) = \mathrm{H}(\mathrm{X}) - \mathrm{H}(\mathrm{X} \mid \mathrm{Y}) = 1 - 1 = 0$

- Intuitively, Y only reveals useful information about X after also conditioning on Z

Chain Rule for Mutual Information

- I(X, Y ; Z) = I(X ; Z) + I(Y ; Z | X)

- Proof: I(X, Y ; Z) = H(X, Y) – H(X, Y | Z)

  = H(X) + H(Y | X) – H(X | Z) – H(Y | X, Z)

  = I(X ; Z) + I(Y; Z | X)

By induction, $I(X_1, \dots, X_n; Z) = \sum_i I(X_i; Z \mid X_1, \dots, X_{\{i-1\}})$

Fano's Inequality

- For any estimator X': X -> Y -> X' with $P_e = \Pr[X' \neq X]$, we have

$$H(X \mid Y) \leq H(P_e) + P_e \cdot \log(|X| - 1)$$

Here X -> Y -> X' is a Markov Chain, meaning X' and X are independent given Y.

"Past and future are conditionally independent given the present"

To prove Fano's Inequality, we need the data processing inequality

Data Processing Inequality

- Suppose X -> Y -> Z is a Markov Chain. Then,

$$I(X\,;Y) \geq I(X;Z)$$

- That is, no clever combination of the data can improve estimation

- I(X ; Y, Z) = I(X ; Z) + I(X ; Y | Z) = I(X ; Y) + I(X ; Z | Y)
- So, it suffices to show I(X ; Z | Y) = 0
- I(X ; Z | Y) = H(X | Y) – H(X | Y, Z)
- But given Y, then X and Z are independent, so H(X | Y, Z) = H(X | Y).

- Data Processing Inequality implies H(X | Y) $\leq H(X \mid Z)$

Proof of Fano's Inequality

- For any estimator X' such that X-> Y -> X' with $P_e = \Pr[X \neq X']$,

we have $H(X\,|Y) \leq H(P_e) + P_e(\log_2|X| - 1)$.

Proof: Let E = 1 if X' is not equal to X, and E = 0 otherwise.

H(E, X | X') = H(X | X') + H(E | X, X') = H(X | X')

H(E, X | X') = H(E | X') + H(X | E, X') $\leq H(P_e)$ + H(X | E, X')

But H(X | E, X') = Pr(E = 0)H(X | X', E = 0) + Pr(E = 1)H(X | X', E = 1)

$$\leq (1 - P_e) \cdot 0 + P_e \cdot \log_2(|X| - 1)$$

Combining the above, H(X | X') $\leq H(P_e) + P_e \cdot \log_2(|X| - 1)$

By Data Processing, H(X | Y) $\leq H(X\,|X') \leq H(P_e) + P_e \cdot \log_2(|X| - 1)$

Tightness of Fano's Inequality

- Suppose the distribution p of X satisfies $p_1 \geq p_2 \geq \ \ldots \geq p_n$
- Suppose Y is a constant, so I(X ; Y) = H(X) – H(X | Y) = 0.
- Best predictor X' of X is X = 1.
- $P_e = \Pr[X' \neq X]$ = 1 – p_1
- H(X | Y) $\leq H(p_1)$ + $(1 - p_1)\log_2(\mathrm{n} - 1)$ predicted by Fano's inequality
- But H(X) = H(X | Y) and if $p_2 = p_3 = \ \ldots = p_n = \frac{1-p_1}{n-1}$ the inequality is tight

Tightness of Fano's Inequality

- For X from distribution $(p_1, \frac{1-p_1}{n-1}, \dots, \frac{1-p_1}{n-1})$
- $H(X) = \sum_i p_i \log\left(\frac{1}{p_i}\right)$

$$= p_1 \log\left(\frac{1}{p_1}\right) + \sum_{i>1} \frac{1-p_1}{n-1} \log\left(\frac{n-1}{1-p_1}\right)$$

$$= p_1 \log\left(\frac{1}{p_1}\right) + (1-p_1) \log\left(\frac{1}{1-p_1}\right) + (1-p_1)\log(n-1)$$

$$= H(p_1) + (1-p_1)\log(n-1)$$