CS 229r Information Theory in Computer Science Jan 28, 2016

Lecture 2

Lecturer: Madhu Sudan *Scribe: Angela Fan*

Overview

Today: formal definition of entropy, compression
Future Lectures: (email to sign up for topics)

1. entropy and counting
2. communication complexity
3. parallel repetition
4. algorithms with max entropy
5. differential privacy

Differential privacy note: Piazza polls are not actually private.

1 Probability Notation

Let:

- Ω represent the support of a probability distribution. In this course, Ω will be a finite set, for example $\{1, 2, 3, ..., k\}$.
- probabilities p over $\Omega = (p_{(1)}, ..., p_{(k)})$ where $\sum_i p_i = 1$
- the notation $p_{(i)} \equiv pr(x = i)$
- We denote a random variable and its distribution like so: $X \sim p$ and refer to the distribution of X as P_x

Given a joint distribution on (X, Y), we notate the following:

- marginal distribution of $x = P_x$ so $P_x(a) = \sum_b P_{XY}(a, b)$ (i.e. summing over all possible values of Y)
- conditional distribution $P_{x|y=b} = \frac{P_{XY}(a,b)}{P_Y(b)}$ using **Bayes Rule**
- note that $P_{X|Y}$ is not a probability, but $\{P_{X|h}\}_b$ represents a collection of probability distributions

2 Entropy

Definition 1 (Entropy) *Given* $X \sim P_x$, $H(x) \triangleq P_x(x) \log_2 \frac{1}{P_x(x)}$

Remark **Where does log base 2 come from?** Entropy is motivated by sending information via bits, explaining the log base 2. Consider $X \perp\!\!\!\perp Y$, for which we would want $H(X, Y) = H(X) + H(Y)$. Then note that $f(A \bullet B) = f(A) + f(B)$, as product of sets implies logarithm.

Definition 2 (Conditional Entropy)

$$\begin{aligned} H(X \mid Y) &\triangleq \mathbb{E}_{y \sim P_y}\left[H(X \mid Y = y)\right] \\ &= \sum_y P_y(y) \sum_{x \in \Omega} P_{x|y}(x) \log \frac{1}{P_{x|y}(x)} \end{aligned}$$

Given the above definitions, we can verify the axioms presented in the previous lecture:

1. $H(x) \leq \log |\text{supp } P(x)|$ immediately following the above definition
2. $H(x) \leq \log |\text{supp } P(x)|$ if P_x is uniform over the support, immediately following the above definition
3. $H(X,Y) = H(X) + H(Y \mid X)$ immediately following the above definition
4. $H(Y \mid X) \leq H(Y)$, which we will prove later

3 Compression

We will discuss three forms of compression, the first two in this lecture and the third more later:

1. **Asymptotic, or n shot compression** (often what is meant when referring to the original Shannon paper)
2. **Single Shot compression** (Huffman coding)
3. **Asymptotic prior free compression**, also known as universal compression (Lempel-Ziv)

3.1 Asymptotic Compression

Premise: Allow $x \sim P_x$, known to both the sender and the receiver. The sender has a series of messages $x_1, \ldots, x_n \overset{iid}{\sim} P_x$ that s/he would like to send to the receiver, while minimizing the number of bits required.

We define:

- **compression function** C: $\Omega^n \to \{0,1\}^{ln}$
- **decompression function** D: $\{0,1\}^{ln} \to \Omega^n$

The compression/decompression process is slightly noisy, and the receiver cannot fully recover 100% of the message, which can be measured by the **probability of error**:

$$P_{error} \triangleq Pr\left[D(C(X_1, ..., X_n)) \neq (X_1, ..., X_n)\right]$$

Goal: given a fixed error $\varepsilon > 0$ that the compression process is allowed to make, design an encoding/decoding scheme that works $\forall n$

Definition 3 (Rate of Asymptotic Compression) $\lim\limits_{\epsilon \to 0} \lim\limits_{n \to \infty} \left\{\frac{ln}{n}\right\}$ *such that E, D function with $P_{error} \leq \varepsilon$*

Theorem 4 (Rate of Asymptotic Compression) *Optimal rate of asymptotic compression $= H(X)$*

Proof First, let us make a simplifying assumption. While we would want to prove Thm 4 for all distributions, let us work with the following:

- $\Omega = \{0, 1\}$
- $P_x = (1 - q, q)$

In this case, $H(X) = H(q) = q \log \frac{1}{q} + (1-q) \log \frac{1}{1-q}$
Thm 4 implies:

1. $\exists$

$$\begin{aligned} E : \{0,1\}^n \to &\{0,1\}^{(H(q)+\delta)n} \\ D : \{0,1\}^l \to &\{0,1\}^n \end{aligned}$$

   such that $P_{error} \leq \varepsilon$. We denote the exponent $(H(q) + \delta)n$ as l for convenience.

2. $H(x)$ is the best rate of asymptotic compression, i.e. $\forall E, D,\ l < (H(q) - \delta)n$ then $P_{error} \geq 1 - \epsilon$

Consider what we can say about $X_1, \ldots, X_n$ globally. By the Law of Large Numbers, or Chernoff bounds, depending on how strict one would like to be:

$$\#\{i \mid X_i = 1\} \leq t \in [(q-\varepsilon')n, (q+\varepsilon')n] \text{ with probability} \geq 1 - \tfrac{\varepsilon}{2}$$

We first prove 1:

In our special case, the sender must send t number of 1's to the receiver, and thus can use the following inefficient algorithm. Generate a table of all possible combinations of indices containing t 1's, and communicate to the receiver the correct row of the table. In this case, the sender must send indices $i \in \binom{n}{t}$.

Thus, the asymptotic length of compression

$$\begin{aligned} &= \frac{\log \binom{n}{t} + \log n}{n} \\ &\approx \frac{\log \binom{n}{qn}}{n} \text{ as log n does not contribute to the leading order term} \end{aligned}$$

We now apply Stirling's approximation: $n! \approx (\frac{n}{e})^n$, so:

$$\begin{aligned} \binom{n}{qn} &\approx \left(\frac{1}{q^2(1-q)^{1-q}} \right)^n \\ \text{taking the log } &= \frac{\log \binom{n}{qn}}{n} \\ &= \frac{nH(q)}{n} \\ &= H(q) \end{aligned}$$

We now consider the second implication of Thm 4, or why we cannot do any better.

If E, D use l such that $2^l \ll \frac{\varepsilon}{2}\binom{n}{t}$, where $t = (q - \varepsilon')n$, then the number of messages decoded correctly is $\leq 2^l$, and all other messages are decoded incorrectly. For every message with $t = \#1's$, $P\left(\text{message is chosen} \leq \frac{1}{\binom{n}{t}}\right)$ by the symmetry of the uniform distribution. Then, $P(\text{correct decoding} \mid t) \leq \frac{\epsilon}{2}$.

Remark Note that with some changes of assumptions, t can be allowed to encompass the entire Chernoff bounded range we established earlier.

■

Remark **What about a general distribution, rather than our special case?** We can use the theorem below.

Theorem 5 (Asymptotic Equipartition Property (AEP)) ***Rough:*** *if $(X_1, ..., X_n) \overset{iid}{\sim} P_x$ then most of the time $(X_1, ..., X_n)$ is chosen almost uniformly from a set of size $2^{H(x)n}$, where the exponent represents the size of the best compression.*

Precisely: *$\exists S \subseteq \Omega^n$ then $\mid S \mid \leq 2^{(1+o(1))H(x)n}$ such that:*

1. *$P(X_1, ..., X_n \in S) \leq 1 - o(1)$*
2. *$\forall (a_1, ..., a_n) \in S, P((X_1, ..., X_n) = (a_1, ..., a_n) \leq \frac{2^{O(n)}}{|S|}$*

Given the asymptotic equipartition property, the compression follows.

Remark **Where does this come from?** We expect to see q fraction of 1's, and S represents the set of all orderings that contain that fraction. The size of S is roughly $\binom{n}{p_1 n\ p_2 n\ ...\ p_k n}$, which is the multinomial coefficient. By Stirling's approximation, this is equal to $2^{H(X)n}$.

3.2 Single Shot Compression

Summary of differences between n shot and single shot compression:

1. single shot compression can use variable length strings
2. error free
3. objective function is to minimize expected length of string
4. prefix-free

Definition 6 (Prefix-free) *C is prefix-free if $\forall x \neq y \in \Omega$, $C(X)$ is not a prefix of $C(y)$*

We define:

- **compression function** C: $\Omega^n \to \{0,1\}^*$ but Ω can have variable length
- **decompression function** D: $\{0,1\}^* \to \Omega^n$

Single shot compression should have no errors, so we want $\forall x, D(C(x)) = x$.

Goal: Since length is now variable, we want to minimize instead the expected length of the compression, $\mathbb{E}_{x \sim P_x}[|C(x)|]$.

Definition 7 (Pattern of C) *Pattern of $C = \{lx\}$ for $x \in \Omega$ such that $C(x) \in \{0,1\}^{lx}$*

Using the pattern, we can tell if a message was from prefix-free code or not.

Remark **Why does single shot compression want 0 error and want to be prefix free?** Prefix-free means that compression can be strung together, and 0 error implies that when we compress m things, we still have an overall error probability of 0.

Remark The length of the message is bounded by the expected value of the length, not the length itself, but we can show using Taylor expansion that the length of the message that is received is very close to the expected length of the message theoretically.

Single shot compression can be solved using:

Theorem 8 (Kraft's Inequality) $\{lx\}$ *is pattern of prefix-free code iff* $\sum_{x\in\Omega} 2^{-lx} \leq 1$

Proof

Let $l = \max_x lx$. We'll think of all possible binary sequences of bits as a balanced binary tree. Pick $C(1), C(2), \ldots$ greedily and eliminate all parts of the binary tree underneath. We claim that each choice rules out 2^{-lx} portion of the length l strings beneath. If choices are made until one leaf is left, then only one path is still alive, indicating that only one node at each level exists. As long as $\sum_{x\in\Omega} \frac{1}{2^{lx}} < 1$, there exists one string of each length uneliminated.

To prove the converse:

Randomly choose a leaf. Let ε_x be the event that $C(X)$ is the prefix of a randomly chosen leaf. Note that ε_x and ε_y are mutually exclusive by the definition of prefix free. $P(\varepsilon_x) = 2^{-lx}$ since the entire path must be correct, and $\sum_n \frac{1}{2^{ln}} = \sum_x P(\varepsilon_x) \leq 1$.

From this, we can derive an encoding scheme. Define $lx = \left\lceil \log \frac{1}{P_x(x)} \right\rceil$, using the ceiling function as we need integer based lengths. This satisfies Kraft's inequality since lengths are admissible.

$$\mathbb{E}_x(lx) = \mathbb{E}_x \left(\left\lceil \log \frac{1}{p_x(x)} \right\rceil \right)$$

which is at most

$$\begin{aligned} &\leq \mathbb{E}_x \left(1 + \log \frac{1}{P_x(x)} \right) \\ &= 1 + \mathbb{E}_x \left(\log \frac{1}{P_x(x)} \right) \\ &= 1 + H(X) \end{aligned}$$

Due to the presence of the 1, the expected length may be off by one bit, which is the subject of much discussion.

Remark **Natural question: Is there a lower bound for 1 shot compression?** Answer is Yes! Can use n shot lower bound to prove that you cannot do better in a single shot!

■