COMPSCI 650 Applied Information Theory Jan 26, 2016

Lecture 1

Instructor: Arya Mazumdar *Scribe:* [redacted]

1 Review and formulas

- Entropy: $H(X) = -\sum_{x \in X} p(x) \log p(x)$
- Joint entropy: $H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log p(x,y)$
- Conditional entropy: $H(X|Y) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log p(x|y)$
- Chain rule: $H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$
- Mutual information: $I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y,X)$
- Relative entropy: $D(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$ (recall that $D(p||q) \neq D(q||p)$)

Question: What is $I(X,X)$?
Answer: $I(X,X) = H(X) - H(X|X)$. $H(X|X)$ is the uncertainty remaining in X if we know X, so $H(X|X) = 0$. Thus, $I(X,X) = H(X)$. This makes sense, because X doesn't give us any new information about itself, so the mutual information between X and itself is just the uncertainty that was already in X.

Question: How is $H(X)$ related to $H(X|Y)$?
Answer: Since relative entropy is a distance,

$$D(p||q) \geq 0$$

We know

$$I(X,Y) = D(p(x,y)||p(x)p(y))$$

and since D is a distance,

$$D(p(x,y)||p(x)p(y)) = I(X,Y) = H(X) - H(X|Y) \geq 0$$

$$H(X) \geq H(X|Y)$$

This also makes sense - knowing other things can't make us know less about X. Either X and Y are independent, and thus $H(X) = H(X|Y)$, or they're dependent, and knowing Y gives us some information about X.

2 Data compression

We have some intuitions about data compression:

- We can't compress a file into nothing. The file must be recoverable from the compressed version.
- We'd like to compress a file to be as small as possible, which means we are interested in lower bounds on compression size.
- We can only compress a file if the probability distribution over its characters isn't uniform. If the distribution is uniform, the codes for each characters would all be the same length, so we wouldn't be doing any compressing.
- To save characters, we want to assign the shortest codes to the most frequently occurring characters.

A code consists of an **alphabet** X of things to encode, a probability distribution over X (i.e. the likelihood of each character to occur), and a **code** C where $c(x)$ is the compressed version of $x \in X$. $\ell(x)$ is the length of $c(x)$. $L(C) = \sum_{x \in X} p(x)\ell(x)$, which is the expected length of $c(x)$.

Note: If the probability distribution over X is uniform, then $\forall x \in X, \ell(x) = \lceil \log |X| \rceil$.

We want to compress a file to the smallest possible size. Thus, compression is an optimization problem where we want to minimize $L(C)$.

2.1 Properties of a code

As mentioned previously, one of our intuitions is that we must be able to recover the file. To ensure this, any code we make must have two properties. First, it must be **non-singular**, meaning that $\forall x, x' \in X, x \neq x' \rightarrow c(x) \neq c(x')$. Second, our code must be **uniquely decodable**, which means any encoded file must have only one valid decoding.

Example: A non-singular code that is not uniquely decodable:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned} c(1) &= 0 \qquad c(2) = 1 \\ c(3) &= 01 \qquad c(4) = 101 \end{aligned}$$

When we attempt to decode 0101, we find that $3, 3$; $1, 4$; $1, 2, 3$; and $1, 2, 1, 2$ are all possible decodings.

Example: A non-singular code that is uniquely decodable:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned} c(1) &= 00 \qquad c(2) = 10 \\ c(3) &= 11 \qquad c(4) = 110 \end{aligned}$$

When we attempt to decode 1011010001110, there is only one possible decoding: $2, 4, 2, 1, 3, 4$.

Even if a code is both non-singular and uniquely decodable, decoding may take exponential time. This is because an encoded string could have substrings that map to multiple decodings, so we'd have to explore all of them to find the valid decoding (for an example, try decoding the uniquely decodable string yourself). We will see an additional code property to ensure instantaneous decoding in the next lecture.

3 Optimizing compression

Theorem 1: Given a uniquely decodable code C with expected length $L(C)$ and alphabet X with entropy $H(X)$,

$$L(C) \geq H(X)$$

To prove this theorem, we'll need to use Kraft's inequality.

3.1 Kraft's inequality

Kraft's inequality states that for any uniquely decodable code C for an alphabet X,

$$\sum_{x \in X} 2^{-\ell(x)} \leq 1$$

Proof: For some integer k,

$$\begin{aligned}
\left(\sum_{x \in X} 2^{-\ell(x)} \right)^k &= \sum_{x_1 \in X} 2^{-\ell(x_1)} \cdot \ldots \cdot \sum_{x_k \in X} 2^{-\ell(x_k)} \\
&= \sum_{x_i \in X} \ldots \sum_{x_k \in X} 2^{-\ell(x_i) - \ldots - \ell(x_k)} \\
&= \sum_{x_i \in X} \ldots \sum_{x_k \in X} 2^{-(\ell(x_i) + \ldots + \ell(x_k))} \\
&= \sum_{x_1^k \in X^k} 2^{-\ell(x_1^k)}
\end{aligned}$$

where x_1^k is a vector of k-length words. Let $a(m)$ represent the number of k-length words in x_1^k such that $\ell(x_1^k) = m$. A property of uniquely decodable codes is that $a(m) \leq 2^m$, giving

$$\begin{aligned}
\left(\sum_{x \in X} 2^{-\ell(x)} \right)^k &= \sum_{m=1}^{k \max_i \ell(x_i)} a(m) \\
&\leq \sum_{m=1}^{k \max_i \ell(x_i)} 2^m 2^{-m} \\
&\leq k \max_i \ell(x_i)
\end{aligned}$$

If Kraft's inequality is **not satisfied**,

$$\left(\sum_{x \in X} 2^{-\ell(x)} \right)^k > k \max_i \ell(x_i)$$

As $\left(\sum_{x \in X} 2^{-\ell(x)} \right)^k$ grows exponentially, $k \max_i \ell(x_i)$ grows linearly. For larger k, the right hand side simply can't grow as fast. Thus,

$$\sum_{x \in X} 2^{-\ell(x)} \leq 1$$

Now, let's use Kraft's inequality to prove **Theorem 1**:

$$\begin{aligned}
L(C) \geq & H(X) \\
\sum_{x \in X} p(x)\ell(x) \geq & -\sum_{x \in X} p(x) \log p(x) \\
L(C) - H(X) = & \sum_{x \in X} p(x)\ell(x) + \sum_{x \in X} p(x) \log p(x) \\
= & -\sum_{x \in X} p(x) \log(2^{-\ell(x)}) + \sum_{x \in X} p(x) \log p(x) \\
= & -\sum_{x \in X} p(x) \log \left(\frac{2^{-\ell(x)}}{\sum_{x \in X} 2^{-\ell(y)}} \right) + \sum_{x \in X} p(x) \log p(x) \\
& -\sum_{x \in X} p(x) \log \left(\sum_{y \in X} 2^{-\ell(y)} \right)
\end{aligned}$$

Let's consider these last three summations:

$$\begin{aligned}
-\sum_{x \in X} p(x) \log \left(\frac{2^{-\ell(x)}}{\sum_{y \in X} 2^{-\ell(y)}} \right) + \sum_{x \in X} p(x) \log p(x) \geq & \sum_{x \in X} p(x) \log \left(\frac{p(x)}{2^{-\ell(x)}} \cdot \frac{1}{\sum_{y \in X} 2^{-\ell(y)}} \right) \\
= & D(p || \frac{2^{-\ell(x)}}{\sum_{y \in X} 2^{-\ell(y)}}) \\
\geq & 0
\end{aligned}$$

We know $\log \left(\sum_{y \in X} 2^{-\ell(y)} \right) \geq \log(q) \geq 0$. So,

$$-\sum_{x \in X} p(x) \log \left(\sum_{y \in X} 2^{-\ell(y)} \right) \geq 0$$

Finally, we have:

$$\begin{aligned}
L(C) - H(X) \geq & 0 \\
L(C) \geq & H(X)
\end{aligned}$$

3.2 Shannon Code

In our consideration of an optimal code which minimizes $L(C) = \sum_{x \in X} p(x)\ell(x)$ and satisfies Kraft's inequality $\sum_{x \in X} 2^{-\ell(x)} \leq 1$, let's use Theorem 1 to get an approximation of $\ell(x)$:

$$\begin{aligned}
\sum_{x \in X} p(x)\ell(x) \geq & -\sum_{x \in X} p(x) \log p(x) \\
\sum_{x \in X} p(x)\ell(x) \geq & \sum_{x \in X} p(x) \log \left(\frac{1}{p(x)} \right)
\end{aligned}$$

Simply eyeballing this inequality tells us that getting $\ell(x)$ as close as possible to $\log\left(\frac{1}{p(x)}\right)$ would result in a smaller code. If for some code

$$\ell(x) = \log\left(\frac{1}{p(x)}\right) \quad \text{and} \quad \sum_{x \in X} 2^{-\ell(x)} = 1$$

then we have a **Shannon Code:**

$$\ell(x) = \lceil log(\frac{1}{p(x)}) \rceil$$