Lecture 13 - Nov 12th, 2013

Class Road Map - IT-I

- L1 (9/26): Overview, Communications, Information, Entropy
- L2 (10/1): Props. Entropy, Mutual Information,
- L5 (10/10): AEP, Compression
- L6 (10/15): Compression, Method of Types,
- L7 (10/17): Types, U. Coding., Stoc. Processes, Entropy rates,
- L8 (10/22): Entropy rates, HMMs, Coding, Kraft,
- L9 (10/24): Kraft, Shannon Codes, Huffman, Shannon/Fano/Elias
- L10 (10/28): Huffman, Shannon/Fano/Elias
- L11 (10/29): Shannon Games,
- LXX (10/31): Midterm, in class.
- L12 (11/7): Arith. Coding, Channel Capacity
- L13 (11/12): Channel Capacity
- L14 (11/14): Channel Capacity
- L15
- L16
- L17
- L18
- L19

Finals Week: December 12th–16th.
Cumulative Outstanding Reading

- Read chapters 1 and 2 in our book (Cover & Thomas, “Information Theory”) (including Fano’s inequality).
- Chapter 3 in our book (Cover & Thomas, “Information Theory”).
- Section 11.1 (method of types).
- Chapter 4 and 5 in our book (Cover & Thomas, “Information Theory”).
- Read stream code chapter 6 in “Information Theory, Inference, and Learning Algorithms” by David J.C. MacKay (available online http://www.inference.phy.cam.ac.uk/mackay/itila/)
- Read Chapter 7 in our book (Cover & Thomas, “Information Theory”).

Homework

- Homework 5 will be out on our web page (http://j.ee.washington.edu/~bilmes/classes/ee514a_fall_2013/), will be due Thursday, Nov 14th, at 11:45pm.
Announcements

- Office hours, every week, now Thursdays 4:30-5:30pm. Can also reach me at that time via a canvas conference.
- Midterm is graded electronically on canvas. Please see your grades and assignments there. Histogram (max score 115) is as follows:

- We'll be doing the final the same way, please make sure, when you get your scanned final, to ASAP upload it to the canvas final assignment.

Radio Communications

- Place yourself back in the 1930s.
- Analog communication model of the 1930s.

Q: Can we achieve perfect communication with an imperfect communication channel?
Q: Is there an upper bound on the information capable of being sent under different noise conditions?
If we increase the transmission rate over a noisy channel will the error rate increase?
Radio Communications

- Key: If we increase the transmission rate over a noisy channel will the error rate increase?
- Perhaps the only way to achieve error free communication is to have a rate of zero.
- The error profile we might expect to see is the following:

\[ P_e \rightarrow R \]

Here, probability of error \( P_e \) goes up linearly with the rate \( R \), with an intercept at zero.
- This was the prevailing wisdom at the time. Shannon was critical in changing that.

Simple Example

- Consider representing a signal by a sequence of numbers.
- We now know that any signal (either inherently discrete or continuous, under the right conditions) can be perfectly represented (or at least arbitrarily well) by a sequence of discrete numbers, and they can even be binary digits.
- Now consider speaking such a sequence over a noisy AM channel.
- Very possible one number will be masked by noise.
- In such case, each number we repeat \( k \) times, where \( k \) is sufficiently large to ensure we can “decode” the original sequence with very small probability of error.
- Rate of this code decreases but we can communicate reliably even if the channel is very noisy.
- Compare this idea to the figure on the following page.
Redundancy added to reduce errors

- Example: speaking numbers in AM radio, say “4,3,5,1,9” very likely could hear “4,static,5,1,static,” part of the message irreparably lost.
- Simple repetition code: say “4,4,3,3,5,5,1,1,9,9” helps, as if we receive “4,static,3,3,5,static,static,1,9,static” we can still recover the message.
- Can repeat > 2× if we wish to decrease the chance of error.
- Also, don’t send the signal, send a representation (or a description, or encoding) of the signal to well fit the channel, and in order to be able to recover (decode) the signal.
- Send information about a signal rather than the exact signal itself. Redundancy added for the sake of the channel, not for the sake of signal source or its contents. Also, rate need not go to zero as $P_e \to 0$! This was revolutionary in the 1930s/1940s!
- Is Morse code an example of this from the 1800s? Not really, Morse is symbol code, directly for sending text, not for other types of signals. Moreover, no coding redundancy, the encoder (telegraph machine) had tonal redundancy but not to the degree done in 1940s.

A key idea

- If we choose the messages carefully at the sender, then with very high probability, they will be uniquely identifiable at the receiver.
- The idea is that we choose the source messages that (tend to) not have any ambiguity (or have any overlap) at the receiver end. I.e.,

  ![Diagram of source messages and received messages](image)

  - This might restrict our possible set of source messages (in some cases severely, and thereby decrease our rate $R$), but if any message received in a region corresponds to only one source message, “perfect” communication can be achieved.
Discrete Channels

Definition 13.3.1 (discrete channel)
A discrete channel is one where there is an input alphabet $\mathcal{X}$, an output alphabet $\mathcal{Y}$, and a distribution $p(y|x)$ which is the probability of observing output $y$ after seeing $x$ as input.

Definition 13.3.2 (memoryless channel)
A discrete channel is memoryless if $y_t$, the output at time $t$, is independent of all previous inputs, given $x_t$. I.e., $y_t \perp \perp x_{1:t-1} | x_t$.

- We will see many instances of discrete memoryless channels (or just DMC).
- Recall back from lecture 1 our general model of communications:

Model of Communication

- **Source** message $W$, one of $M$ messages.
- **Encoder** transforms this into a length-$n$ string of source symbols $X^n$ (we might call them “channel input symbols”).
- **Noisy channel** distorts this message into a length-$n$ string of receiver symbols $Y^n$ (we might call them “channel output symbols”).
- **Decoder** attempts to reconstruct original message as best as possible, comes up with $\hat{W}$, one of $M$ possible sent messages.
- **Note**, $p(x, y) = p(x)p(y|x)$. $p(y|x)$ will model our channel and is fixed in most cases (we can’t control the channel). $p(x)$ is our source distribution, and we get to determine (and optimize over) it.
So we have a source $X$ governed by $p(x)$ and channel that transforms $X$ symbols to $Y$ symbols and which is governed by the conditional distribution $p(y|x)$.

These two items $p(x)$ and $p(y|x)$ is sufficient to compute the mutual information between $X$ and $Y$. That is, we compute

$$I(X; Y) = I_{p(x)}(X, Y) = \sum_{x, y} p(x) p(y|x) \log \frac{p(y|x)}{p(y)}$$

$$= \sum_{x, y} p(x) p(y|x) \log \frac{p(y|x)}{\sum_{x'} p(y|x') p(x')}$$

We write this as $I(X; Y) = I_{p(x)}(X, Y)$, meaning implicitly the MI quantity is a function of the entire distribution $p(x)$, for a given fixed channel $p(y|x)$. Recall from L3, concave in $p(x)$ for fixed $p(y|x)$.

We will often be optimizing over the input distribution $p(x)$ for a given fixed channel $p(y|x)$.

**Definition 13.3.3 (information flow)**

The rate of information flow through a channel is given by $I(X; Y)$, the mutual information between $X$ and $Y$, in units of bits per channel use.

**Definition 13.3.4 (capacity)**

The information capacity of a channel is the maximum information flow.

$$C \triangleq \max_{p(x) \in \Delta} I(X; Y)$$

where $\Delta$ is the set of all possible probability distributions over source alphabet $\mathcal{X}$. Thus, $C$ is the maximum number of bits sent over the channel per channel use.

**Definition 13.3.5 (rate)**

The rate $R$ of a code is measured in the number of bits per channel use.
Fundamental Limits of Compression

- For compression, if error exponent is positive, then error $\to 0$ exponentially fast as block length $\to \infty$. Note, $P_e \propto e^{-nE(R)}$.
- That is,

![Error Exponent Graph]

- Only hope of reducing error was if $R > H$. Something “funny” happens at the entropy rate of the source distribution. Can’t compress below this without incurring error.

Radio Communications

- Key: If we increase the transmission rate over a noisy channel will the error rate increase?
- Perhaps the only way to achieve error free communication is to have a rate of zero.
- The error profile we might expect to see is the following:

![Error Probability Graph]

- Here, probability of error $P_e$ goes up linearly with the rate $R$, with an intercept at zero.
- This was the prevailing wisdom at the time. Shannon was critical in changing that.
Fundamental Limits of Data Transmission/Communication

- For communication, lower bound on probability of error becomes bounded away from 0 as the rate of the code $R$ goes above a fundamental quantity $C$. Note, $P_e \propto e^{-nE(R)}$.
- That is, we have a “dual” situation to entropy compression, i.e.,

\[ \log P_e \]

\[ C \]

\[ R \rightarrow \]

\[ E(R) \]

- We will show: only way to get low error is $R < C$. Something funny happens at the point $C$, the channel capacity.
- Note that $C$ is not 0, so can still achieve “perfect” communication over a noisy channel as long as $R < C$.

For the moment, all we known about $C \triangleq \max_{p(x) \in \Delta} I(X; Y)$ is its definition, which is the result of an optimization problem.
- not yet connected it to communications, and to communicating about or below the rate $C$.
- We will do this. For now, think of $C$ has being measured in units of “bits per channel use”
Examples of discrete memoryless channels (BSC)

- Noiseless binary channel, diagram shows \( p(y|x) \)

\[
X = \{0, 1\} \quad X \quad Y \quad Y = \{0, 1\}
\]

\[
\begin{align*}
&0 \xrightarrow{1/2} 0 \\
&1 \xrightarrow{1/2} 1
\end{align*}
\]

- So, \( p(y = 0|x = 0) = 1 = 1 - p(y = 1|x = 0) \) and \( p(y = 1|x = 1) = 1 = 1 - p(y = 0|x = 1) \), so channel is just an input copy.

- One bit sent at a time is received without error, so capacity should be 1 bit (intuitively, we can reliably send one bit per channel usage).

- \( I(X;Y) = H(X) - H(X|Y) = H(X) \) in this case, so \( C = \max_{p(x)} I(X;Y) = \max_{p(x)} H(X) = 1 \).

- Clearly, \( p(0) = p(1) = 1/2 \) achieves this capacity.

- Also, \( p(0) = 1 = 1 - p(1) \) has \( I(X;Y) = 0 \), so achieves zero information flow.

Noisy Channel with non-overlapping outputs

Consider the channel

\[
\begin{align*}
&0 \xrightarrow{1/2} 0 \\
&0 \xleftarrow{1/2} 1 \\
&1 \xrightarrow{1/2} 2 \\
&1 \xleftarrow{1/2} 3
\end{align*}
\]

- Here, \( p(Y = 0|X = 0) = p(Y = 1|X = 0) = 1/2 \) and \( p(Y = 2|X = 1) = p(Y = 3|X = 1) = 1/2 \).

- If we receive a 0 or 1, we know 0 was sent. If we receive a 2 or 3, a 1 was sent.

- Thus, \( C = 1 \) since only two possible error free messages.

- Same argument applies

\[
I(X;Y) = H(X) - H(X|Y) = H(X).
\]

- Again uniform distribution \( p(0) = p(1) = 1/2 \) achieves the capacity.
Permutation Channel

Consider the channel

\[
\begin{array}{cc}
0 & 0 \\
1 & 1 \\
\end{array}
\]

- Here, \(p(Y = 1|X = 0) = p(Y = 0|X = 1) = 1\).
- So output is a binary permutation (swap) of input.
- Thus, \(C = 1\) no information lost.
- In general, for alphabet of size \(k = |\mathcal{X}| = |\mathcal{Y}|\), let \(\sigma\) be a permutation, so that \(Y = \sigma(X)\).
- Then \(C = \log k\).

Asside: on the optimization to compute the value \(C\)

- To maximize a given function \(f(x)\), it is sufficient to show that \(f(x) \leq \alpha\) for all \(x\), and then find an \(x^*\) such that \(f(x^*) = \alpha\).
- \(x^*\) achieves the upper bound of \(\alpha\) for \(f(\cdot)\).
- We’ll be doing this over the next few slides when we want to compute \(C = \max_{p(x)} I(X; Y)\) for fixed \(p(y|x)\).
- The solution \(p^*(x)\) that we find that achieves this maximum won’t necessarily be unique.
- Also, the solution \(p^*(x)\) that we find won’t necessarily be the one that we end up, say, using when we wish to do channel coding.
- Right now \(C\) is just the result of a given optimization.
- We’ll see that \(C\), as computed, is also the critical point for being able to channel code with vanishingly small error probability.
- The resulting \(p^*(x)\) that we obtain as part of the optimization in order to compute \(C\) won’t necessarily be the one that we use for actual coding (examples are forthcoming).
Consider the channel

- So 26 input symbols, and each symbol maps probabilistically to itself or its lexicographic neighbor.
- I.e., \( p(A \rightarrow A) = p(A \rightarrow B) = 1/2 \), etc.
- Each symbol always has chance of error, so how can we communicate without error?
- Choose subset of symbols that can be uniquely disambiguated on receiver side. Choose every other source symbol, A, C, E, ...
- Thus \( A \rightarrow \{A, B\} \), \( C \rightarrow \{C, D\} \), \( E \rightarrow \{E, F\} \), etc. so that each received symbols has only one unique source symbol.
- Capacity \( C = \log 13 \)
- Q: what happens to \( C \) when probabilities are not all \( 1/2 \)?

We can also compute the capacity more mathematically. E.g.,

\[
C = \max_{p(x)} I(X; Y) = \max_{p(x)} \left( H(Y) - H(Y|X) \right) \tag{13.4}
\]

\( \forall x \) when \( X = x \), \( \exists \) two equal prob. choices for \( Y \), hence ...

\[
= \max_{p(x)} H(Y) - 1 \tag{13.5}
\]

\[
= \log 26 - 1 = \log 13 \tag{13.6}
\]

- The \( \max_{p(x)} H(Y) = \log 26 \) can be achieved by using the uniform distribution for \( p(x) \), for which when we choose any \( x \) symbol, there is equal likelihood of two \( Y \)'s being received.
- An alternatively \( p(x) \) would put zero probability on the alternates (B, D, F, etc.). In this case, we still have \( H(Y) = \log 26 \)
- So the capacity is the same in each case (\( \exists \) two \( p(x) \) that achieved this) but only one is what we would use, say, for error free coding.
Binary Symmetric Channel (BSC)

- A bit that is sent will be flipped with probability $p$.
- $p(Y = 1|X = 0) = p = 1 - p(Y = 0|X = 0)$. $p(Y = 0|X = 1) = p = p(Y = 1|X = 1)$.

- The BSC is an important channel since it is a simple model but at the same time captures some of the difficulties of more complicated channels.

- Q: can we still achieve reliable (“guaranteed” error free) communication with this channel? A: Yes, if $p < 1/2$ and if we do not ask for too high a transmission rate (which would be $R > C$), then we can. Actually, any $p \neq 1/2$ is sufficient.
- Intuition: think about AEP and/or block coding.
- But how to compute $C$ the capacity?

BSC Capacity

- Intuition: think about AEP and/or block coding.
- But how to compute $C$ the capacity?

$$I(X; Y) = H(Y) - H(Y|X) = H(Y) - \sum_x p(x)H(Y|X = x) \quad (13.7)$$

$$= H(Y) - \sum_x p(x)H(p) = H(Y) - H(p) \leq 1 - H(p) \quad (13.8)$$

- To achieve the upper bound, need $H(Y) = 1$. Note that

$$\Pr(Y = 1) = \Pr(Y = 1|X = 1)\Pr(X = 1)$$

$$+ \Pr(Y = 1|X = 0)\Pr(X = 0) \quad (13.10)$$

$$= (1 - p)\Pr(X = 1) + p(1 - \Pr(X = 1)) \quad (13.11)$$

$$= p + (1 - 2p)\Pr(X = 1) \quad (13.12)$$

- So $H(Y) = 1$ if $H(X) = 1$ (i.e., if $\Pr(X = 1) = 1/2$).
- Thus, we get that $C = 1 - H(p)$ which happens when $X$ is uniform.
Thus, we get that $C = 1 - H(p)$ which happens when $X$ is uniform.

- If $p = 1/2$ then $C = 0$, so if it randomly flips bits, then no information can be sent.
- If $p \neq 1/2$, then we can communicate, albeit potentially slowly. E.g., if $p = 0.499$ then $C = 2.8854 \times 10^{-6}$ bits per channel use. So to send one bit, need to use the channel quite a bit.
- If $p = 0$ or $p = 1$, then $C = 1$ and we can get maximum possible rate (i.e., the capacity is one bit per channel use).

Let's temporarily look ahead to address this problem.

We can “decode” the source using the received string, source distribution, and the channel model $p(y|x)$ via Bayes rule. I.e.

$$
\Pr(x|y) = \frac{\Pr(y|x) \Pr(x)}{\Pr(y)} = \frac{\Pr(y|x)\Pr(x)}{\sum_{x'} \Pr(y|x')\Pr(x')} \quad (13.13)
$$

- Error if $x \neq \hat{x}$, and $\Pr(\text{error}) = \Pr(x \neq \hat{x})$.
- This is optimal decoding in that it minimizes the error, $\text{Error}(\hat{x}) = 1 - p(\hat{x}|y(x))$ when $y$ is received for sent $x$.
- This is minimal if we chose $\arg\max_x p(x|y)$ since the error becomes $1 - \Pr(\hat{x}|y)$ which is minimal.
Minimum Error Decoding

- Note: Computing quantities such as $\Pr(x|y)$ is a task of probabilistic inference.
- Often this problem is difficult (NP-hard, see Cooper and Herskovitz, 1990). This means that doing minimum error decoding might very well be exponentially expensive (unless P = NP).
- Many real world codes are such that computing the exact computation must be approximated (i.e., no known fast algorithm for minimum error or maximum likelihood decoding).
- Instead we do approximate inference algorithms (e.g., loopy belief propagation, message passing, etc.). These algorithms tend still to work very well in practice (achieve close to the capacity $C$).
- But before doing that, we need first to study more channels and the theoretical properties of the capacity $C$.

Binary Erasure Channel

- $e$ is an erasure symbol, if that happens we don’t have access to the transmitted bit.
- The probability of dropping a bit is then $\alpha$.
- We want to compute capacity. Obviously, $C = 1$ if $\alpha = 0$.

$$C = \max_{p(x)} I(X;Y) = \max_{p(x)} (H(Y) - H(Y|X))$$

$$= \max_{p(x)} H(Y) - H(\alpha)$$

So while $H(Y) \leq \log 3$, we want actual value of the capacity.
Binary Erasure Channel

- Let $E = \{Y = e\}$. Then

\[ H(Y) = H(Y, E) = H(E) + H(Y | E) \]

- Let $\pi = \Pr(X = 1)$. Then

\[ H(Y) = H((1 - \pi)(1 - \alpha), \frac{\alpha}{\pi}(1 - \alpha)) \]

\[ = H(\alpha) + (1 - \alpha)H(\pi) \quad (13.16) \]

\[ = H(\alpha) + (1 - \alpha)H(\pi) \quad (13.17) \]

This last equality follows since $H(E) = H(\alpha)$, and

\[ H(Y | E) = \alpha H(Y | e) + (1 - \alpha)H(Y | Y \neq e) = \alpha \cdot 0 + (1 - \alpha)H(\pi) \]

Then we get

\[ C = \max_{p(x)} H(Y) - H(\alpha) \quad (13.18) \]

\[ = \max_{\pi} (1 - \alpha)H(\pi) + H(\alpha) - H(\alpha) \quad (13.19) \]

\[ = \max_{\pi} (1 - \alpha)H(\pi) = 1 - \alpha \quad (13.20) \]

- Best capacity when $\pi = 1/2 = \Pr(X = 1) = \Pr(X = 0)$.
- This makes sense, loose $\alpha\%$ of the bits of original capacity.