18.413: Error-Correcting Codes Lab February 12, 2004

Lecture 4

Lecturer: Daniel A. Spielman

4.1 Prior, Extrinsic and Posterior Probabilities, II

I should have given the classic example relating these probabilities: that of testing for a rare but deadly dissease. Let's pretend that we have a test for Ebola that always says "yes" 95% of the time if you do have Ebola, but also has a 5% chance of saying "yes" even if you do not have it. Feeling sick, you go into the medical center, the resident in charge decides to test you for Ebola, and the test comes back "yes". Should you presume that you have Ebola?

In our language, we say that the *extrinsic probability* that you have Ebola given that the test says "yes' is 0.95. On the other hand to figure out the actually probability that you have Ebola, the *posterior probability*, we would need to know the prior probability that you have Ebola. Assuming that you are a randomly chosen U.S. resident, and that only 1/2,000,000 U.S. residents contract Ebola each year, then the prior probability that you have Ebola is 1/2,000,000. So, the posterior probability that you have Ebola is

$$\frac{1/2000000 \times .95}{1/2000000 \times .95 + 1999999/2000000 \times .05} = .00000949.$$

You are quite releived to observed that there is approximately a 1/100000 chance that you have Ebola, Of course, if you are not a randomly chosen U.S. resident, and just happened to return from a month of living in a jungle with monkeys, the prior probability will not be as good.

4.2 Normalizing constants

We will soon tire of complicated derivations involving multiple applications of the law of conditional probability. To simplify our calculations and notation, we make the following observation: if x is uniformly chosen from an alphabet A and y is a random variable depending on x, then we have

$$\mathrm{P}[x = a | y = b] = \frac{\mathrm{P}[y = b | x = a]\, \mathrm{P}[x = a]}{\mathrm{P}[y = b]}.$$

While we don't know the probability that $y = b$, it does only depend on b. Since x is chosen uniformly, $\mathrm{P}[x = a]$ does not depend on a. So, we will set

$$c_b = \frac{\mathrm{P}[x = a]}{\mathrm{P}[y = b]},$$

and write

$$\mathrm{P}\left[x = a | y = b\right] = c_b \mathrm{P}\left[y = b | x = a\right].$$

This will be fine if all the c_b terms cancel out in our calculations (which they always will). In fact, if it is clear that these terms will cancel, then we will just write

$$\mathrm{P}\left[x = a | y = b\right] \sim \mathrm{P}\left[y = b | x = a\right],$$

and skip the c_b altogether.

4.3 Example: symmetric channels

For example, let C be a symmetric channel that has input alphabet $\{0, 1\}$ and some output alphabet B. Recall that the definition of a symmetric channel implies that for each $b \in B$ there is another symbol $b' \in B$ such that

$$\begin{aligned}\mathrm{P}\left[\text{rec } b | \text{sent } 0\right] &= \mathrm{P}\left[\text{rec } b' | \text{sent } 1\right] \qquad \text{and} \\ \mathrm{P}\left[\text{rec } b | \text{sent } 1\right] &= \mathrm{P}\left[\text{rec } b' | \text{sent } 0\right].\end{aligned}$$

Let p denote $\mathrm{P}\left[\text{sent } 1 | \text{rec } b\right]$. Then

$$\mathrm{P}\left[\text{sent } 1 | \text{rec } b\right] = 1 - p,$$

and we can write

$$\begin{aligned}\mathrm{P}\left[\text{rec } b | \text{sent } 1\right] &= c_b \cdot p, \\ \mathrm{P}\left[\text{rec } b | \text{sent } 0\right] &= c_b \cdot (1 - p).\end{aligned}$$

This constant c_b has an interpretation: any symmetric two-input channel can be described as a distribution over binary symmetric channels. That is, it is equivalent to choosing a set of crossover probabilities, $p_1, \ldots, p_k$, and then assigning to each induced channel BSC_{p_i} a probability q_i, where $\sum_{i=1}^{k} q_i = 1$. When an input is fed to the channel, the channel first chooses an i according to the distribution $\vec{q}$, and then passes the bit through BSC_{p_i}. If we are being nice to the detector (and we will be), then the channel will output both a 0 or 1 and the chosen value of p_i, which provides a reliability measure for the output. This constant c_b is exactly q_i for this crossover probability.

4.4 Decoding Codes

We now turn to the problem of decoding general error-correcting codes. We will see that there are two different ideal ways of doing this.

Let $\mathcal{C} \subseteq \{0, 1\}^n$ be a code. Assume that we choose a random codeword $(a_1, \ldots, a_n) \in \mathcal{C}$, send it over a symmetric channel, and receive a vector $(b_1, \ldots, b_n)$. Moreover, know

$$p_i = \mathrm{P}\left[x_i = 1 | \text{rec } b_i\right].$$

Thus,

$$\mathrm{P}\left[\text{rec } \vec{b} | \text{send } \vec{x}\right] = \prod_{i=1}^{n} c_{b_i} (p_i)^{[x_i=1]} (1-p_i)^{[x_i=0]}.$$

So, we can compute

$$\mathrm{P}\left[\text{send } \vec{x} | \text{rec } \vec{b}\right] = \frac{\mathrm{P}\left[\text{rec } \vec{b} | \text{send } \vec{x}\right]}{\sum_{\vec{y} \in \mathcal{C}} \mathrm{P}\left[\text{rec } \vec{b} | \text{send } \vec{y}\right]}.$$

Where does this last equality come from? The answer is our formula for how to interpret the output of a channel. We can view the whole process as a meta-channel in which the input symbols are elements of $\mathcal{C}$, and the output alphabet is B^n.

Given the computation the question is what task we would like our decoder to perform. One reasonable choice, called maximum likelihood decoding, is to output the codeword $\vec{x} \in \mathcal{C}$ maximizing $\mathrm{P}\left[\text{send } \vec{x} | \text{rec } \vec{b}\right]$. That is, the most likely codeword given the received vector. This is the best choice if our goal is to minimize the probability of having any errors at all. However, if our goal is just to get through as many bits as possible, this is not necessariliy the best choice.

If we want to maximize the number of bits transmitted correctly, then we should choose our estimate for each separately. For example, to maximize our chance of estimating the first bit correctly, we should compute

$$\mathrm{P}\left[x_1 = 1 | \text{rec } \vec{b}\right],$$

and guess that $x_1 = 1$ if this probability is greater than $1/2$. The leads us to ask how we compute this probability. Well,

$$\begin{aligned}\mathrm{P}\left[x_1 = 1 | \text{rec } \vec{b}\right] &= \mathrm{P}\left[\bigvee_{\vec{x} \in \mathcal{C} : x_1 = 1} \text{sent } \vec{x} | \text{rec } \vec{b}\right] \\ &= \sum_{\vec{x} \in \mathcal{C} : x_1 = 1} \mathrm{P}\left[\text{sent } \vec{x} | \text{rec } \vec{b}\right].\end{aligned}$$

In this class, we will focus on optimizing bit-by-bit estimation. The reason is largely accidental: classical coding schemes were more suited to maximum-likelihood decoding, whereas iterative coding schemes are more suited to bit-by-bit estimation. That said, we will not be able to implement a decoder that achieves either of these goals for any non-trivial code. Rather, we will implement decoders that try to come close.

4.5 Example: parity

Even though it has many terms, this computation simplifies greatly for the parity code. In the parity code, we encode $w_1, \ldots, w_k \in \{0,1\}^k$ by setting $x_i = w_i$ for $1 \leq i \leq k$ and $x_{k+1} = \sum x_i \mod 2$. (i.e. $n = k+1$). Say that we transmit, receive $b_1, \ldots, b_n$, and learn that $p_i = \mathrm{P}\left[x_i = 1 | b_i\right]$. We will find that the estimation of x_1 can be cleanly broken into two terms: one from b_1 and one from $b_2, \ldots, b_n$. As we already know how to tell what b_1 says about x_1, let's examine the contribution from $b_2, \ldots, b_n$.

We first note that it is equivalent to choose $x_2, \ldots, x_n$ uniformly at random subject to $x_1 = \sum_{i=2}^n x_i \mod 2$. Let $\mathcal{C}_1$ be the set of $(x_2, \ldots, x_n)$ with odd parity and $\mathcal{C}_0$ be the set with even parity. We then obtain

$$
\begin{aligned}
\mathrm{P}\left[x_1 = 1 | (b_2, \ldots, b_n)\right] &= \sum_{(x_2,\ldots,x_n)\in\mathcal{C}_1} \mathrm{P}\left[x_2, \ldots, x_n | b_2, \ldots, b_n\right] \\
&= \sum_{(x_2,\ldots,x_n)\in\mathcal{C}_1} \frac{\mathrm{P}\left[b_2, \ldots, b_n | x_2, \ldots, x_n\right]}{\sum_{y_2,\ldots,y_n} \in \mathcal{C}_1 \cup \mathcal{C}_0 \mathrm{P}\left[b_2, \ldots, b_n | y_2, \ldots, y_n\right]} \\
&= \frac{\sum_{(x_2,\ldots,x_n)\in\mathcal{C}_1} \mathrm{P}\left[b_2, \ldots, b_n | x_2, \ldots, x_n\right]}{\sum_{y_2,\ldots,y_n\in\mathcal{C}_1\cup\mathcal{C}_0} \mathrm{P}\left[b_2, \ldots, b_n | y_2, \ldots, y_n\right]}.
\end{aligned}
$$

Let's look at the bottom term first. We have

$$
\begin{aligned}
\sum_{y_2,\ldots,y_n\in\mathcal{C}_1\cup\mathcal{C}_0} \mathrm{P}\left[b_2, \ldots, b_n | y_2, \ldots, y_n\right] &= \sum_{y_2,\ldots,y_n\in\{0,1\}^{n-1}} \prod_{i=2}^n c_{b_i} (p_i)^{[x_i=1]} (1-p_i)^{[x_i=0]} \\
&= (p_i + (1-p_i))^{n-1} \prod_{i=2}^n c_{b_i} \\
&= \prod_{i=2}^n c_{b_i}.
\end{aligned}
$$

Similarly, we can derive that the top term satisfies

$$
\begin{aligned}
\sum_{(x_2,\ldots,x_n)\in\mathcal{C}_1} \mathrm{P}\left[b_2, \ldots, b_n | x_2, \ldots, x_n\right] &= \frac{1}{2}\left[((1-p_i) - p_i)^{n-1} - ((1-p_i) - p_i)^{n-1}\right] \prod_{i=2}^n c_{b_i} \\
&= \frac{1}{2}\left(1 - \prod_{i=2}^n (1-2p_i)\right) \prod_{i=2}^n c_{b_i}.
\end{aligned}
$$

Thus,

$$
\mathrm{P}\left[x_1 = 1 | (b_2, \ldots, b_n)\right] = \frac{1}{2}\left(1 - \prod_{i=2}^n (1-2p_i)\right).
$$

Now, to finally compute

$$
\mathrm{P}\left[x_1 = 1 | (b_1, \ldots, b_n)\right],
$$

we treat

$$
\mathrm{P}\left[x_1 = 1 | (b_2, \ldots, b_n)\right],
$$

and

$$
\mathrm{P}\left[x_1 = 1 | b_1\right],
$$

as independent estimates, and combine them according to the rule derived last class:

$$
\mathrm{P}\left[x_1 = 1 | (b_1, \ldots, b_n)\right] = \frac{\mathrm{P}\left[x_1 = 1 | (b_2, \ldots, b_n)\right] \mathrm{P}\left[x_1 = 1 | b_1\right]}{\mathrm{P}\left[x_1 = 1 | (b_2, \ldots, b_n)\right] \mathrm{P}\left[x_1 = 1 | b_1\right] + \mathrm{P}\left[x_1 = 0 | (b_2, \ldots, b_n)\right] \mathrm{P}\left[x_1 = 0 | b_1\right]}.
$$