EE 178/278A Probabilistic Systems Analysis

Spring 2014 Tse/Hussami Lecture 16

Communications Example Continued

In the previous lecture, we covered an application in communication. We will now wrap up our communication example on the binary symmetric channel (BSC), where p represents the flip probability in the channel. What is the noisiest channel? When $p = 1$ and $p = 0$, the channel is perfect because we know exactly what the input is given the output. However, when $p = 0.5$, the output does not tell us anything about the input to the channel.

Our detection rule uses p, but how do we know p in practice? We can try to estimate it by sending a sequence of 0 bits. The output will contain some 0s and 1s, from which we can estimate

$$\hat{p} = \frac{\text{number of 1s received}}{\text{number of 0s sent}},$$

which is the fraction of ones received. This process is called training in the communication language. It is called supervised learning in machine learning. Once we estimate p, then we can use the channel to communicate data.

Typically, any probabilistic model we use in an application has some parameters $\theta_1, \ldots, \theta_k$. There are two steps to using of the probability model:

1. First, we get some data and try to estimate the parameters. The first such question we tackled was in the polling example, where we estimated p, the fraction of Democrats in the population. The channel estimation example is another one. (Mathematically, they are actually identical problems. Why?)

2. Second, based on the model. we can design algorithms to solve whatever problem that is at hand, like communications, speech recognition or tracking, and analyze the performance of the algorithms.

The above channel estimate $\hat{p}$ is called the *maximum likelihood estimate* (MLE). We are actually choosing as an estimate the value of p that maximizes the probability of observing the data:

$$\hat{p}_{ML} = \text{argmax}_p \, \mathbf{P}(\text{observation}; p).$$

Why? $\mathbf{P}(\text{observation}; p) = p^k(1-p)^{n-k}$, where n is the total number of training bits sent, and k is the total number of 1s observed at the output of the channel. By simple calculus,

$$\text{argmax}_p p^k(1-p)^{n-k} = \frac{k}{n},$$

which is the intuitive estimate we came up with earlier.

Example 2: Speech Recognition

Today, we will cover another important application of probability: speech recognition. How does Siri understand what we are saying? We speak into the iPhone microphone, which samples the analog sound waveform. Now, the iPhone needs to figure out what we told it. The goal is to transform the analog waveform into the command "driving directions to Stanford" for example. There is a lot of randomness involved in this problem: the accent of the user, the different characteristics of his voice or the ambient noise etc.

We want to design a speech recognition algorithm. How do we approach this problem? We need to understand the structure of the spoken language. Each word is a natural unit into which we can decompose the sentence. We can also decompose the words into smaller units. From a phonetic point of view, a word is decomposed into phonemes, that can contain vowels and consonants. For example 'dr' in the word driving is a phoneme. There are around 30 or 40 phonemes in english. Therefore the problem becomes figuring out the sequence of phonemes in the sentence. The sequence of phonemes forms the set of random variables that we want to estimate.

Let X_i be the *ith* phoneme in the sentence that we want to recognize. We have $X_1, \cdots, X_n$ random variables representing the sentence we want to recognize. We need to relate the signal that we picked up on the microphone to this sequence of variables. Each phoneme roughly corresponds to 10ms. We chop the analog signal into 10ms intervals. Then, we take the signal in each 10ms interval and extract some key features from it. The relevant information contained in speech is most apparent in the frequency domain. So usually, a short window fourier analysis is done on the sampled signal from each 10 ms time interval, and the corresponding fourier coefficients are extracted. These coefficients serve as features in the frequency domain, or spectral information to describe the phoneme in that 10ms interval.

Typically there are multiple features for the signal in each 10ms interval, corresponding to multiple Fourier coefficients for example. But let us simplify the story by assuming there is only a single feature. Moreover, we will assume that the value of the feature is quantized so that this can be represented by a discrete random variable. (We will consider the case when Y_i is continuous later on.) Let Y_i be the value of the feature in the *ith* interval. The system diagram can be represented as:

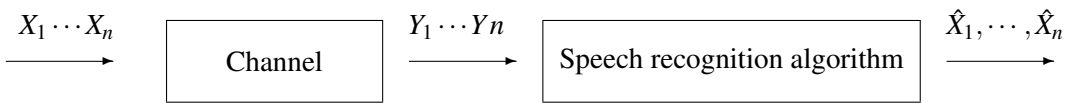


Figure 1: *A system diagram for the speech recognition problem.*

We need to specify two things to be able to derive our speech recognition algorithm:

(1) A prior probability on the input

(2) A relation between the input and the output, which , by analogy with the communication application, can be considered as a *channel*.

Let us start by characterizing our channel. We assume that each Y_i only depends on its corresponding input X_i. That is, conditional on X_i, Y_i is independent of the other inputs. Let us define $Q(b|a) := \mathbf{P}(Y_i = b|X_i = a)$.

Using the conditional independence assumption, we write:

$$\begin{aligned}\mathbf{P}(Y_1 = b_1 \cdots Y_n = b_n | X_1 = a_1 \cdots X_n = a_n) &= \mathbf{P}(Y_1 = b_1 | X_1 = a_1) \cdots \mathbf{P}(Y_n = b_n | X_n = a_n) \\ &= Q(b_1|a_1) \cdots Q(b_n|a_n).\end{aligned}$$

This is similar to our communication channel where the output at time i only depended on the input bit at time i. We will assume for today that we are given these Q function values. In the next lecture, we will explain how to estimate these values from the data. But note that by making the conditional independence assumption, we are drastically reducing the number of parameters that needs to be estimated. If each X_i and Y_i can take on say 40 values and $n = 500$ (correspond to 5 seconds of speech), then the number of parameters to specify the full conditional distribution $\mathbf{P}(Y_1 = b_1 \cdots Y_n = b_n | X_1 = a_1 \cdots X_n = a_n)$ is $(40^2)^5 00$, while the number of parameters to specify under the conditional independence assumption is 40^2, the number of parameters $Q(b|a)$!

How do we model the input sequence $X_1 \cdots X_n$? One natural way of doing it is assuming that these random variables are independent. We tried this trick before many times, like with independent coin flips for example. However, this is not a good idea in this application. Some phonemes are more likely to follow other phonemes. For example the phoneme *th* is more likely to be followed by an "*e*" rather than a "*s*". So assuming the random variables to be independent seems like a very poor model. How are we going to capture the dependency, while still having a small number of parameters in our model?

Consider $\mathbf{P}(X_n = a_n | X_1 = a_1 \cdots X_{n-1} = a_{n-1})$. This is the probability conditioning the present upon the past. We are going to assume that the dependence of X_n on the past is entirely through the random variable X_{n-1} that immediately precedes it. So the simplification we make is to suppose

$$\mathbf{P}(X_n = a_n | X_1 = a_1, \cdots, X_{n-1} = a_{n-1}) = \mathbf{P}(X_n = a_n | X_{n-1} = a_{n-1}) \qquad \text{for all } a_1, \cdots, a_n.$$

For $n = 3$ for example, it reduces to

$$\mathbf{P}(X_3 = a_3 | X_1 = a_1, X_2 = a_2) = \mathbf{P}(X_3 = a_3 | X_2 = a_2) \qquad \text{for all } a_1, a_2, a_3.$$

This is equivalent to saying that X_3 is independent of X_1 conditional on X_2. This is called the *Markov property*, and the corresponding figure that illustrates the relation between the variables is

$$X_1 - X_2 - X_3$$

This is called a *graphical model*, with the random variables as nodes of the graph. The interpretation of the graph is that when if one disconnects the graph into two subgraphs G_1 and G_2 by removing a node X_i, then the random variables in G_1 and G_2 are independent conditional on X_i.

Going back to the speech recognition problem, recall we are assuming X_n is independent of all the past given X_{n-1}. More generally, we will assume that X_i is independent of all the past given X_{i-1} for all i. We can now write the joint probability distribution as:

$$\begin{aligned} & \mathbf{P}(X_1 = a_1, X_2 = a_2 \cdots X_n = a_n) \\ = \; & \mathbf{P}(X_n = a_n | X_1 = a_1, \cdots, X_{n-1} = a_{n-1}) \mathbf{P}(X_1 = a_1, \cdots, X_{n-1} = a_{n-1}) \\ = \; & \mathbf{P}(X_n = a_n | X_{n-1} = a_{n-1}) \mathbf{P}(X_{n-1} = a_{n-1} | X_{n-2} = a_{n-2}) \cdots \mathbf{P}(X_2 = a_2 | X_1 = a_1) \mathbf{P}(X_1 = a_1) \end{aligned}$$

In other words, the relation between all our random variables is represented by the graphical model:

$$X_1 - X_2 - X_3 \cdots - X_n,$$

where X_{i+1} is independent of $X_{i-1}, \cdots, X_1$ given X_i. This is called a *Markov chain*. To specify a Markov chain, we need to specify:

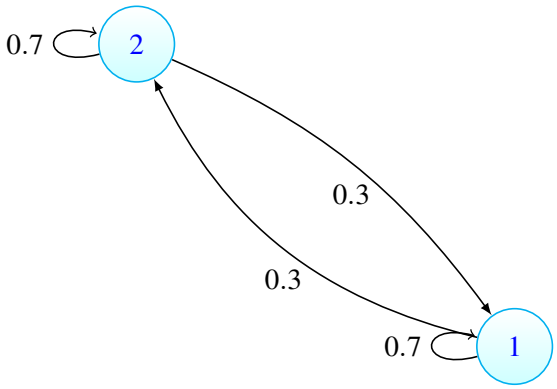
(1) Transition probabilities

$$P(a'|a) := \mathbf{P}(X_i = a'|X_{i-1} = a) \qquad \text{for all } a, a'.$$

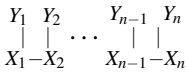
(2) Initial distribution

$$\pi(a) := \mathbf{P}(X_1 = a) \qquad \text{for all } a.$$

Consider an example of a Markov chain with $X_i \in \{1,2\}$, shown in the figure below. Suppose it has the transition probabilities: $\mathbf{P}(2|1) = \mathbf{P}(1|2) = 0.3$, and $\mathbf{P}(1|1) = \mathbf{P}(2|2) = 0.7$. The value X_i takes on is called the *state* of the system at time i. If we want to predict how the Markov chain will go forward, we only need to know the current value and not the past. The figure below is called the *state transition diagram* of the Markov chain.



Finally, we can put all the random variables of the problem into an overall graphical model for the speech recognition problem:



Note that if we remove the node X_i, the node Y_i will be disconnected from the rest of the graph. This reflects the fact that Y_i depends on other random variables only through X_i.

Now the speech recognition problem is from the sequence of Y_i's , we want to estimate the sequence of X_is. We do not want the complexity of our algorithm to increase exponentially with the number of phonemes. Ideally the complexity should grow linearly.

One last question about the model before we talk about the speech recognition algorithm: is the observation sequence $Y_1 \cdots Y_n$ a Markov chain? No (why?). But the underlying sequence that we want to figure out is a Markov chain. That is the reason for calling this a *Hidden Markov Model* (HMM).