ECE 830 Fall 2011 Statistical Signal Processing

instructor: R. Nowak

Lecture 1: Elements of Statistical Signal Processing

Throughout the course we will denote a signal of interest by x. The signal could be a digitized waveform or image, a time-series from a discrete process like the Dow-Jones index, or multi-dimensional signals such as images or video sequences. Usually, x will denote a vector of scalar samples or measurements of the signal under consideration, and we will refer to it as the *data* or *observation* of the signal.

Signals are often contaminated with errors or may themselves be partially random or unpredictable. There are many ways to model these sorts of uncertainties. In this course we will model them probabilistically. Let $p(x|\theta)$ denote a probability distribution parameterized by θ. The parameter θ could represent characteristics of errors or noise in the measurement process or govern inherent variability in the signal itself. For example, if x is a scalar measurement then we could have $p(x|\theta) = \frac{1}{\sqrt{2\pi}}\exp(-(x-\theta)^2/2)$, a model which says that typically x is close to the value of θ and rarely is very different. We will look at a concrete applications in the next section.

Statistical signal processing involves three processes: measurement, modeling, and inference (depicted in Figure **??** below).

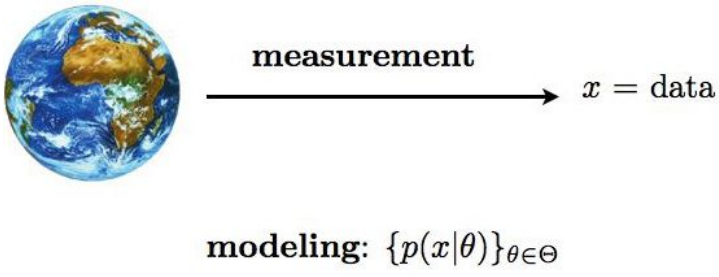

Figure 1: Statistical Signal Processing: Measurement, modeling and inference.

There are four fundamental inference problems in statistical signal processing that will be the focus of this course.

Detection: Suppose that θ takes one of two possible values, so that either $p(x|\theta_1)$ or $p(x|\theta_2)$ fit the data x the best. Then we need to "decide" whether $p(x|\theta_1)$ is a better model than $p(x|\theta_2)$. More generally, θ may be one of a finite number of values $\{\theta_1, \ldots, \theta_M\}$ and we must decide among the M models.

Parameter Estimation: Suppose that θ belongs to an infinite set. Then we must decide or choose among an infinite number of models. In this sense, estimation may be viewed as an extension of detection to infinite model classes. This extension presents many new challenges and issues and so it is given its own name.

Signal Estimation/Prediction: In many problems we wish to predict the value of a signal y given an observation of another related signal x. We can model the relationship between x and y using a *joint* prob-

ability distribution, $p(x, y)$. The *conditional* distribution of y given x, denoted by $p(y|x)$, can be derived from the joint distribution and the prediction problem can then be viewed as determining a value of y that is highly probable given x.

Learning: Sometimes we don't know a good model the relationship between x and y, but we do have a number of "training examples", say $\{x_i, y_i\}_{i=1}^n$, that give us some indication of the relationship. The goal of learning is to design a good prediction rule for y given x using these examples, instead of $p(y|x)$. One approach, called the "plug-in", is to use the training data to form an estimate of $p(y|x)$ and then use it derive a predictor. This is usually suboptimal, since estimating $p(y|x)$ is at least as difficult as the prediction problem itself.

1 Why Probabilistic Models?

The observations or measurements we make are seldom perfect; often they are impure and contaminated by effects unknown to us. We call these effects *noise*. Our models are seldom perfect. Even the best choice of θ may not perfectly predict new observations. We call these mismodelling errors *bias*.

How do we model noise and bias, these uncertain errors? We need a calculus for uncertainty, and among many that have been proposed and used, the probabilistic framework appears to be the most successful, and in many situations it is physically plausible as well.

2 A Detection Example

Consider a binary communication system. Let $s = [s_1, \ldots, s_n]$ denote a digitized waveform. A transmitter communicates a bit of information by sending s or $-s$ (for 1 or 0, respectively). The receiver measures a noisy version of the transmitted signal which we will model as

$$x_i = \theta s_i + \epsilon_i \ , \quad i = 1, \ldots, n$$

The parameter θ is either $+1$ or -1, depending on which bit the transmitter is sending. The $\{\epsilon_i\}$ represent errors incurred during the transmission process. So we have two models, or *hypotheses*, for the data:

$$\begin{aligned} H_0 : \ x_i &= +s_i + \epsilon_i \ , \quad i = 1, \ldots, n \\ H_1 : \ x_i &= -s_i + \epsilon_i \ , \quad i = 1, \ldots, n \end{aligned}$$

How well does $\{s_i\}$ match $\{x_i\}$? How well does $\{-s_i\}$ match $\{x_i\}$? This comparison can be made by computing a function of the data. Functions of data are called *statistics*. A natural statistic in this problem is the correlation statistic:

$$\begin{aligned} t &= \sum_{i=1}^n s_i x_i \\ &= \theta \sum_{i=1}^n s_i^2 \ + \ \sum_{i=1}^n s_i \epsilon_i \end{aligned}$$

If the errors are noise-like and don't resemble the signal $\{s_i\}$, then $\sum_{i=1}^n s_i \epsilon_i \approx 0$. So a reasonable way to decide which value of the bit was sent is to decide that 0 was sent if $t < 0$ and that 1 was sent if $t > 0$. To quantify the performance of this test we need a mathematical model for the errors $\{\epsilon_i\}$.

2.1 A Parameter Estimation Example

Suppose that we are sensing a sinusoidal signal in noise, modeled as follows.

$$x_k \ = \ A \sin(\omega\, k + \phi) \ + \ w_k \ , \quad k = 1, \ldots, n$$

where the amplitude, frequency and phase are unknown parameters to be estimated from $\{x_k\}$ and $\{w_k\}$ are noise/errors in the measurements. Note that here we assume that the signal has been sampled, which is most often the case in modern systems, but one could also pose a continuous-time version of this problem. For the most part we will focus on the discrete-time, sampled-data models in this course.

In this case we have $\theta = [A, \omega, \phi]$. Specifying a probability distribution for the noises (say Gaussian), would yield a probability distribution for our data $p(x|\theta)$. Given the data $x = [x_1, x_2, \ldots, x_n]$, how would you estimate the values of the parameters?

2.2 A Signal Estimation Example

Imagine that you are collaborating with biologists who are interested in imaging biological systems using a new type of microscopy. The imaging system doesn't produce perfect images: the data collected is distorted and noisy. As a signal processing expert, you are asked to develop an image processing algorithm to "restore" the image. Let us assume that the distortion is a linear operation. Then we can model the collected data by the following equation.

$$y \quad = \quad Hx + w$$

where x is the ideal image we wish to recover (represented as a vector, each element of which is a pixel), H is a known model of the distortion (represented as a matrix), and w is a vector of noise. It is tempting to ignore w and simply attempt to solve the system of equations $y = Hx$ for x. There are two problems with this approach. First, the system of equations may not admit a unique solution, depending on the physics of the imaging system. If a unique solution exists, the problem is said to be *well-posed*, otherwise it is called *ill-posed*. Second, even if the system of equations is invertible, it may be *ill-conditioned* which means that small perturbations due to noise and numerical methods can lead to large errors in the restoration of x.

2.3 A Learning Example

Now imagine you are working with geneticists to develop a diagnostic tool to predict whether patients have a certain disease. The tool is to be based on genomic data from the patient. For example, suppose that a microarray experiment is used to measure the levels of gene expression in the patient. For each of m genes we have an expression level (which reflects the amount of protein that gene is producing). Let x denote an $m \times 1$ vector of the expression levels and let y denote a binary variable indicating whether or not the patient has the disease.

The goal is to design a tool to predict the value of y from x. If we had a joint probability model $p(x, y)$, then we could use it to design the predictor. But suppose that instead of a model we have only training data in the form of pairs $\{(x_i, y_i)\}_{i=1}^n$ from a set of randomly selected human subjects. Our tool will be a function that combines a new test vector x and the training data $\{(x_i, y_i)\}_{i=1}^n$ in order to predict the value of y corresponding to x. A simple approach to this problem is to find the x_i that is "closest" in some sense to x, and then use the corresponding y_i as our prediction. This is called a *nearest-neighbor* method.