EECS 16A Designing Information Devices and Systems I
Spring 2015 Homework 7

This homework is due April 2, 2015 at 5PM.

1. Recipe Reconnaissance

Engineering Edibles has been growing in size and popularity and has introduced two new cookies: Decadent Dwight and Heavenly Hearst. As a result of their popularity there is an increased interest in understanding their secret recipes. The bakery produces 40 Decadent Dwight and 50 Heavenly Hearst cookies each day.

Each cookie costs \$1. The bakery business is way overpriced, and everyone knows about 80 cents of the cost comes from profits (and labor); ingredients are only worth *about* 20 cents. The team from Berkeley wants to figure out the recipes for the two new cookies, and they know the recipes are *different*. For the purpose of this problem, each cookie only contains eggs, sugar and butter (of course, you would use flour, water, etc. in real life!).

(a) How many unknowns are there in the problem? How many equations would you need to be able to solve for these unknowns?

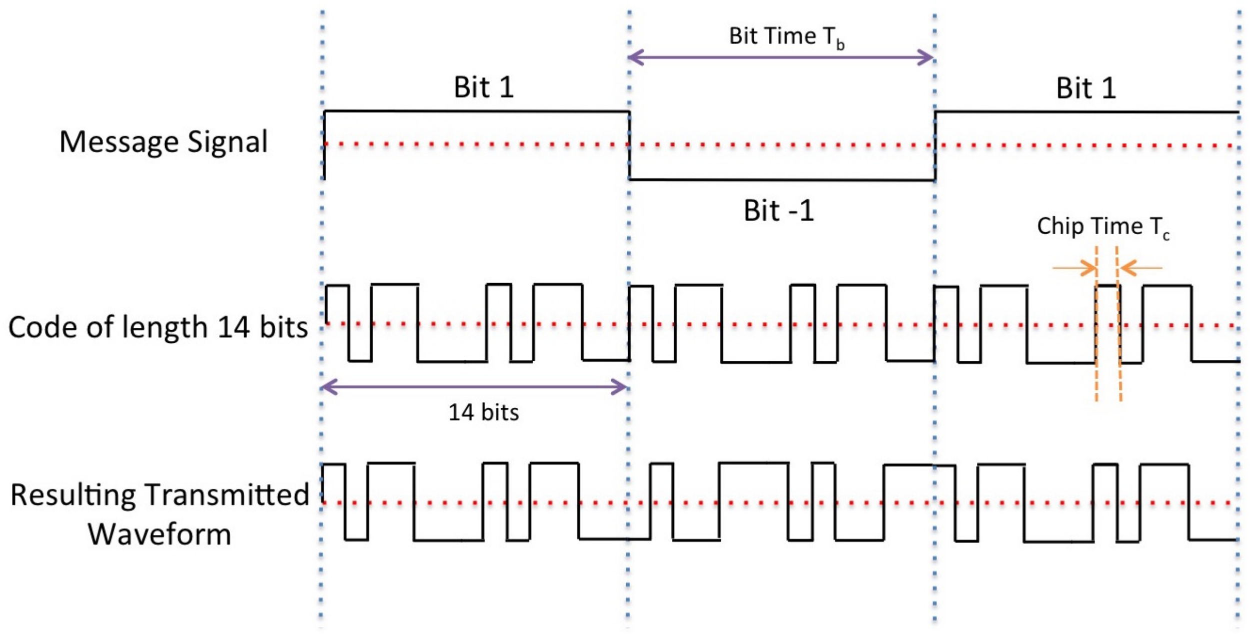
(b) This time unfortunately the team is not able to find precise information about the ingredients used in the cookie making process. They know from the supermarket that 1 kg of sugar costs 5 dollars, and 100 grams of butter costs 1 dollar. They stake-out Engineering Edibles, and they see Bob the Baker buy a dozen eggs for \$2 each day. They hold the stake-out, and they see Bob buying a 5 kg bag of sugar every week. They hire a master-taster, who tells them there is *exactly* 10 grams of butter in each of the cookies (for both Decadent Dwight and Heavenly Hearst). All this is not enough for the team to unlock the secret recipe! (Why?) One day, however, the team gets lucky. A new employee of EE likes to gossip, and the team hears through the grapevine that Heavenly Hearst contains *about* 10 grams of sugar.

Using the information above, set up the problem as a least-squares problem and find best estimate of the secret recipe. Identify what variables you are estimating. You are welcome to use a computer to solve this.

(c) The Berkeley team decides to do scientific experiments to better their estimates of the recipe. They obtain a scale, and weigh the new cookies. Their (cheap and second-hand) scale is only accurate to the gram and might not be tared correctly, so they get noisy observations: Decadent Dwight is *about* 25 grams, and Heavenly Hearst is about 24 grams. It's common knowledge that 1 egg = 50 grams. Redo the least squares problem with this new information, and see how the values change. Do they improve or not? Why?

2. GPS Receivers

The Global Positioning System (GPS) is a space-based satellite navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. Gold All satellites use the same carrier frequency to transmit the signals. To permit this without undue interference between the users, the satellites employ



"spread-spectrum" technology (very similar to CDMA) and a special coding scheme where each transmitter is assigned a code that serves as a "signature". An example is depicted in the figure below. The *message signal* to be transmitted changes at a much slower timescale than the timescale of the signature code, which is very fast.

In the example figure, we are showing a message signal that is a stream of $+1$ or -1 values. The signature code is also a stream of $+1$ or -1 values of length 14 bits. The signature code is multiplied by the appropriate *message signal* to get the final transmitted waveform. Let T_b be the "bit time", i.e. time for each message bit and and T_c be the "chip time" which is the time for each new symbol of the code to be generated. In the figure, $T_b = 14T_c$.

The GPS satellites use "Gold codes" which are 1023 bits long. So, for our problem, $T_b = 1023T_c$. (In reality, $T_b = 20 \times 1023 \times T_c$.) The Gold codes have special properties:

- Their auto-correlation of a Gold code (correlation with itself) is very high.
- The cross-correlation between different codes is very low.

These codes are generated using a linear feedback shift register (LFSR).You can read more about this if you are interested but for the problem you don't need to know how this works. The take-away is that the Gold codes are vectors of $+1$ and -1 values that are are "almost orthogonal". For the purpose of this question we only consider 24 GPS satellites. Download the file `prob7.ipynb` and the corresponding data files for the following questions:

(a) Auto-correlate the Gold code of satellite 10 with itself and plot it. Python has functions for this. What do you observe?

(b) Cross-correlate the Gold code of satellite 10 with satellite 13 and plot it. What do you observe?

(c) Now, consider a random signal, i.e. a signal that is not generated due to a specific code but is a random ± 1 sequence. Cross-correlate it with the Gold code of satellite 10. What do you observe? How does this compare to the cross-correlation of satellite 10 and satellite 13? What does this mean about our ability to identify satellites?

(d) The signals received by a receiver include signals from the satellites as well as an additional noise term. This is often modeled as a Gaussian noise term. You don't need to understand Gaussians here but we use them since they form a good model for how the transmitted signal might be perturbed (large perturbations are very unlikely, and small perturbations are more likely).

Use the Gaussian noise generator to generate a random vector of length 1023, and cross correlate this with the Gold code of satellite 10. What do you observe?

For the next subparts of this problem, the signal is corrupted by Gaussian noise. Use the observation from this subpart for solving the rest of the question.

(e) Now, assume that signals from multiple satellites are added at the receiver. So the signatures of multiple different satellites are present in the code. In addition, noise might be added to the signal. What are the satellites present in `data1.npy`?

(f) Let's assume that you can hear only one satellite, Satellite A, at the location you are in (though this never happens in reality). Let's also assume that this satellite is transmitting a length 5 sequence of $+1$ and -1 after modulating it onto the 1023 bit Gold code corresponding to Satellite A. Find out from `data2.npy` which satellite it is and what sequence of ± 1 it is transmitting.

(g) For the purpose of this problem, we'll assume that all the satellites transmit the same unique sequence of $+1$s and -1s. These are transmitted using the procedure described in the figure (called modulation, which we will learn more about soon.)

Signals from different transmitters arrive at the receiver with different propagation delays. So effectively the signals from different satellites are superimposed on each other with different offsets at the start. This propagation delay is used to find out how far the satellite is from the receiver. To find out exactly what the offset due to propagation delay is, you want to figure out the starting point of the signal transmission, and you can do this by cross-correlating the signature codes with the received signal at different offsets. What do you expect to observe when you cross-correlate the signature for a particular satellite (say Satellite A) with the received signal at the offset corresponding to the propagation delay of Satellite A?

What satellites are you able to see in `data3.npy` and what are the relative delays assuming that the message signal that was being sent was exactly $\begin{bmatrix}1 & 1 & -1 & -1 & -1\end{bmatrix}$.

3. Image analysis

Applications in medical imaging often require an analysis of images based on the pixels of the image. For instance, we might want to count the number of cells in a given sample. One way to do this is to "take a picture" of the cells and use the pixels to determine the locations and thus the number of cells. Alternatively, automatic detection of shape is useful in image classification as well (e.g. consider a robot autonomously trying to find out where a mug is in it's field of vision).

Let us focus back on the medical imaging scenario. You are interested in finding the exact position and shape of a cell in an image, so you want to find the equation of the ellipse that bounds the cell relative to a given coordinate system that is represented by the image. Your collaborator uses edge detection techniques to find a bunch of points that are approximately along the edge of the cell. We assume that the origin is in the center of the image with standard axes and collect the following points: $(.3,-.7)$, $(.5,.91)$, $(.9,-.99)$, $(1,1.01)$, $(1.2,-.93)$, $(1.5,.8)$, $(2,0)$. Submit your code for all parts of this problem, feel free to add it to the original iPython notebook.

Recall that a quadratic equation of the form

$$ax^2 + bxy + cy^2 + dx + ey = 1. \tag{1}$$

can be used to represent an ellipse (if $b^2 - 4ac < 0$), and a quadratic equation of the form

$$a(x^2 + y^2) + dx + ey = 1 \tag{2}$$

is a circle if $d^2 + e^2 - 4a > 0$. The circle has fewer parameters.

(a) How can you find the equation of a circle that surrounds the cell? First, provide a setup and formulate a set of matrix equations to do this, i.e. an equation of the form $A \cdot \vec{x} = \vec{b} + \vec{e}$, where $\vec{b}$ represents your observations and $\vec{e}$ represents the unknown errors.

(b) How can you find the equation of an ellipse that surrounds the cell? Provide a setup and formulate a set of matrix equations to do this as above.

(c) Write a short program in iPython to fit a circle using these points. If you model your system of equations as $A\vec{x} = \vec{b} + \vec{e}$ where $\vec{e}$ is the error vector and the number of data points is N, what is $\frac{\|\vec{e}\|}{N}$? Plot your points and the best fit circle in iPython.

(d) Write a short program in iPython to fit an ellipse using these points. If you model your system of equations as $A\vec{x} = \vec{b} + \vec{e}$ where $\vec{e}$ is the error vector and the number of data points is N, what is $\frac{\|\vec{e}\|}{N}$? Plot your points and the best fit ellipse in iPython. How does this error compare to the one in the previous subpart? Which technique is better?

4. Labelling patients using gene expression data

Least-squares techniques are useful for many different kinds of prediction problems. The core ideas we learned in class have been extensively further developed. These ideas are commonly used in machine learning for finance, healthcare, advertising, image processing and many other fields. Here we'll explore how least squares can be used for classification of data in a medical context.

Gene expression data of patients, along with other factors such as height, weight, age, family history, is often used to understand the likelihood that a patient might develop certain common diseases such as diabetes. Gene expression profiles can be read using DNA microarray technology, which uses tissue samples from a patient. This data, along with the patient specific characteristics above, can be combined into a vector to get a set of features that describe each patient.

Many scientific studies look at models in mice to understand how gene expression relates to diabetes. Previous studies have shown that the expression of the tomosin2 and ts1 genes are correlated to the onset of diabetes in mice. How can we predict whether or not a mouse will develop diabetes based on data about this expression as well as other factors of the mouse? We will use some (fake) data to explore this.

We are given information about the age and weight of the mouse, and additionally have access to data about whether the genes tomosin2, ts1 and chn1 (a third gene) were expressed or not. The gene expression data is captured using vectors that are $+1$ if the gene is expressed and -1 if the gene is not expressed. Similarly, whether or not a mouse has diabetes is also captured using a $+1, -1$ vector, where $+1$ indicates that the mouse has diabetes. Using this data we would like to develop a linear model that predicts whether or not a mouse will have diabetes.

$$\alpha_1(\text{age}) + \alpha_2(\text{weight}) + \alpha_3(\text{tomosin2}) + \alpha_4(\text{ts1}) + \alpha_5(\text{chn1}) \tag{3}$$

We would like the above expression to be positive if the mouse has diabetes and negative if the mouse does not have diabetes.

(a) In problems such as this, it is common to use some *training* data to generate a model. Turns out, a good heuristic for this can be developed using a least squares technique. Set up a linear model for the problem in a format we have used for least-squares problems $A\vec{x} = \vec{b}$. Here, $\vec{b}$ will be a vector with $+1, -1$ entires. α_i's are your unknowns.

(b) Using the (fake) *training* data `diabetes_train.npy`, generate the linear model using the least squares technique, i.e. find the unknown model parameters for the given data set. Include the unknown parameter values in the writeup of your homework. Use the `Gene Data.ipynb` file.

(c) Now it is time to use the model you have developed to make some predictions! It is interesting to note here that we are not looking for a real number to model whether each mouse has diabetes or not, we are looking for a binary label. So we will use the *sign* of the expression above to assign a ± 1 value to each mouse. Predict whether each mouse with the characteristics in the *test* data set `diabetes_test.npy` will get diabetes. There are four mice in the test data set. Include the ± 1 vector that indicates whether or not they have diabetes in your writeup.

5. Projections and least-squares... some theory at the end

Consider a vector $\vec{b} \in \mathbb{R}^3$. Let $\vec{a}_1 \in \mathbb{R}^3$ and $\vec{a}_2 \in \mathbb{R}^3$ be two other vectors not equal to $\vec{b}$.

(a) Calculate the equation of the plane that contains $\vec{a}_1$ and $\vec{a}_2$.

(b) Find the point in the plane of $\vec{a}_1$ and $\vec{a}_2$ that is closest to $\vec{b}$. (Note that all points in the plane can be expressed as a linear combination of $\vec{a}_1$ and $\vec{a}_2$.) Show that this is the orthogonal projection of $\vec{b}$ onto this plane, i.e. show that the error vector is orthogonal to this plane.