EECS 16A Designing Information Devices and Systems I
Summer 2020 Homework 7A

This homework is due Wednesday August 12, 2020, at 23:59 PT.
Self-grades are due Thursday August 13, 2020, at 23:59 PT.

Submission Format

Your homework submission should consist of **one** file.

- `hw7A.pdf`: A single PDF file that contains all of your answers (any handwritten answers should be scanned) as well as your IPython notebook (if any) saved as a PDF.

Homework Learning Goals: *One of the objectives of this homework is to show how correlation can be utilized for positioning. Another aim is to familiarize you with the concepts of projection and least squares. These concepts will be used towards solving overdetermined systems of equations affected by noise.*

1. Labeling Patients Using Gene Expression Data

Least squares techniques are useful for many different kinds of prediction problems. Numerous researchers have extensively further developed the core ideas that we have learned in class. 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, and family history, are often used to predict the likelihood that a patient might develop a certain disease. This data can be combined into a vector that describes each patient. This vector is often referred to as a feature vector.

Many scientific studies examine mice to understand how gene expression relates to diabetes in humans. 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 feature vectors for each mouse as:

$$\begin{bmatrix} \text{age} \\ \text{weight} \\ \text{tomosin2} \\ \text{ts1} \\ \text{chn1} \end{bmatrix}$$

Age and weight in the vector above are represented by real numbers, while the presence or absence of the expression of the genes tomosin2, ts1, and chn1 is captured by $+1$ and -1 respectively. For example, the vector $\begin{bmatrix} 2 & 20 & 1 & -1 & -1 \end{bmatrix}^T$ means a 2 month old mouse that weighs 20 grams and expresses the genes tomosin2 but not ts1 or chn1.

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

$$f(\text{age}, \text{weight}, \text{tomoson2}, \text{ts1}, \text{chn1}) = \alpha_1(\text{age}) + \alpha_2(\text{weight}) + \alpha_3(\text{tomosin2}) + \alpha_4(\text{ts1}) + \alpha_5(\text{chn1}).$$

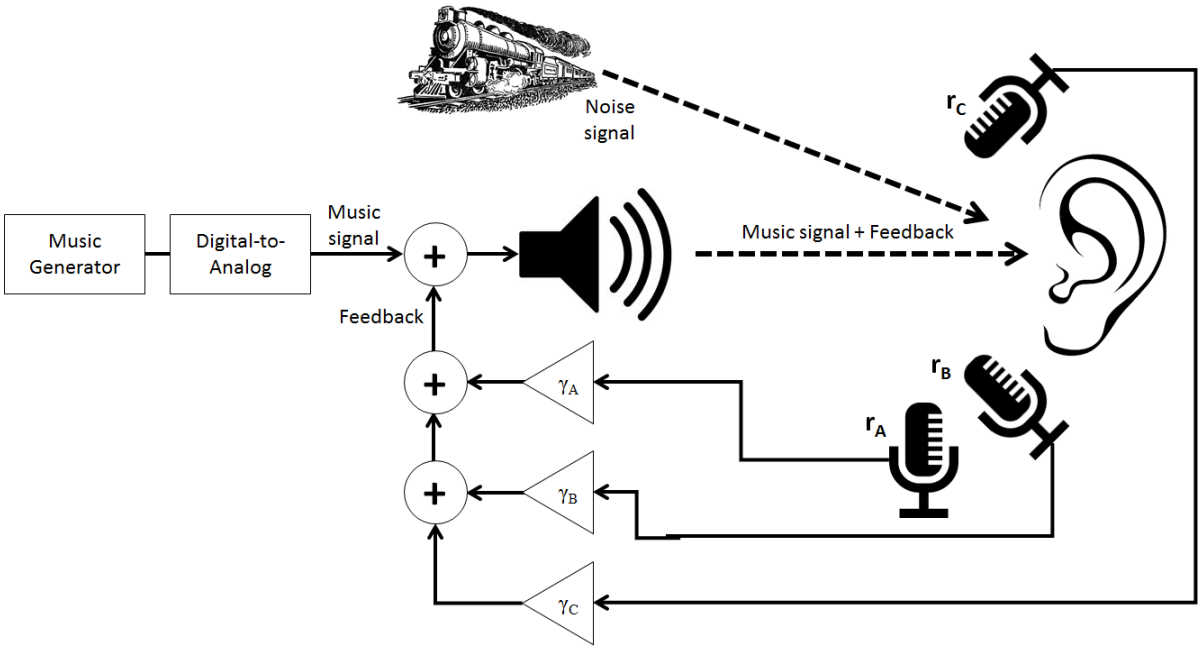
(a) We wish to set up a linear model for the problem in the format $\mathbf{A}\vec{x} = \vec{b}$. Here, $\vec{b}$ will be a vector with $+1, -1$ entries where a 1 represents that the mouse is diabetic and -1 represents that the mouse is not diabetic. The feature vectors of each mouse will be included in the rows of the matrix $\mathbf{A}$. Set up the problem by writing $\mathbf{A}$, $\vec{x}$, and $\vec{b}$ in terms of the variables in the feature vectors, α_i, and any other variables you define. What are your unknowns?

(b) Training data is data that is used to develop your model. Use the (fake) *training* data `diabetes_train.npy` to find the optimal model parameters for the given data set. What are the optimal parameter values? Use the provided IPython notebook.

(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. Therefore, 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 answer. What is the prediction accuracy (number of correct predictions divided by total number of predictions) of your model?

2. Noise Canceling Headphones

In this problem, we will explore a common design for noise cancellation using noise-canceling headphones as an example application. We will work with the model shown in the figure below.



A music signal is generated at a speaker and transmitted to the listener's ear. If there is noise in the environment (*e.g.* other people's voices, a train going by), this noise signal will be superimposed on the music signal and the listener will hear both. In order to cancel the noise, we will try to record the noise and subtract it directly from the transmitted signal with the hope that we can achieve perfect cancellation of everything but the music. Since our system is imperfect, we'll have to solve a least squares problem.

The gain blocks marked by γ (Greek "gamma") represent scalar multiplication, and we will assume that they can take on any real number, positive or negative.

(a) First, consider a noise signal noted by $\vec{n}$,

$$\vec{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{bmatrix}$$

We can use three microphones to record this signal, Mic A, Mic B, and Mic C. Each microphone records the noise, but they each have their own characteristics. This means that they do not perfectly record the noise and that they are distinct recordings. Let $\vec{r}_A$, $\vec{r}_A$, and $\vec{r}_A$ represent the noise that each microphone picks up:

$$\vec{r}_A = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix}, \vec{r}_B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix}, \vec{r}_C = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}$$

We can arrange the recordings into a matrix $\mathbf{R}$ and the microphone gains, γ, into a vector $\vec{\gamma}$

$$\mathbf{R} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ a_4 & b_4 & c_4 \\ a_5 & b_5 & c_5 \end{bmatrix}, \vec{\gamma} = \begin{bmatrix} \gamma_A \\ \gamma_B \\ \gamma_C \end{bmatrix}$$

For the system that is drawn in the figure above, write down the signal at the listener's ear using matrix notation. It should include:

- the music signal $\vec{m} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \end{bmatrix}$
- the noise signal $\vec{n}$
- the matrix of recorded noise signals $\mathbf{R}$
- the microphone gain vector $\vec{\gamma}$

You can assume that the microphones do not pick up the music signal.

(b) Ideally, we would want to have a signal at the ear that matches the original music signal perfectly. In reality, this is not possible, so we will aim to minimize the effect of the noise. What quantity would we need to minimize to make sure this happens? Write your answer in terms of the matrix $\mathbf{R}$, the vector of mic gains $\vec{\gamma}$, and the noise vector $\vec{n}$.

(c) We can solve minimization problems by the least squares method. In effect, if we have a problem, $\min_{\vec{x}} \left\| \mathbf{A}\vec{x} - \vec{b} \right\|$, then the $\vec{x}$ that solves this problem is,

$$\vec{\hat{x}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\vec{b} \tag{1}$$

.

Implement this least squares method in the IPython Notebook helper function `doLeastSquares`.

(d) For the given $\vec{n}$ and the recordings, $\vec{r}_A$, $\vec{r}_B$, $\vec{r}_C$, below, report the γ's that minimize the effect of noise.

$$\vec{n} = \begin{bmatrix} 0.10 \\ 0.37 \\ -0.45 \\ 0.068 \\ 0.036 \end{bmatrix}, \vec{r}_A = \begin{bmatrix} 0 \\ 0.11 \\ -0.31 \\ -0.012 \\ -0.018 \end{bmatrix}, \vec{r}_B = \begin{bmatrix} 0 \\ 0.22 \\ -0.20 \\ 0.080 \\ 0.056 \end{bmatrix}, \vec{r}_C = \begin{bmatrix} 0 \\ 0.37 \\ -0.44 \\ 0.065 \\ 0.038 \end{bmatrix}$$

The next few questions can be answered in the IPython notebook by running the associated cells.

(e) We can use this least squares solution to find the best γ values for our algorithm for a given number of microphones. Follow the instructions in the IPython notebook to load a music signal and some noise signals. Listen to the music signal and the two noise signals. Which ones are full of static and which ones are not?

(f) Use the IPython notebook to record the first noise signal using the `recordAmbientNoise` function and calculate a vector $\vec{\gamma}$. Create the noise cancellation signal by performing the multiplication $\mathbf{R}\vec{\gamma}$.

(g) Add the noise cancellation signal (with the correct sign) to the music signal to get the signal from the speaker and, finally, add the noise signal to the speaker signal. Play the noisy signal and the noise-cancelled signal. Can you hear a difference?

(h) Try adding the other noise signal to the music signal without re-calculating new values for $\vec{\gamma}$ (don't solve the least squares problem again). Add the noise-cancelling signal to your speaker signal and add the noise as well. Comment on the quality of the resulting noise-cancelled signal. Is it perfect or are there artifacts?

3. Homework Process and Study Group

Who else did you work with on this homework? List names and student ID's. (In case of homework party, you can also just describe the group.) How did you work on this homework?