This homework is due December 2, 2019, at 23:59.
Self-grades are due December 9, 2019, at 23:59.

Submission Format
Your homework submission should consist of one file.

- hw13.pdf: A single PDF file that contains all of your answers (any handwritten answers should be scanned) as well as your IPython notebook saved as a PDF.

If you do not attach a PDF of your IPython notebook, you will not receive credit for problems that involve coding. Make sure that your results and your plots are visible. Assign the IPython printout to the correct problem(s) on Gradescope.

Submit the file to the appropriate assignment on Gradescope.

1. Mechanical: Projections

   (a) Find the projection of \( \vec{b} = \begin{bmatrix} 3 \\ 2 \\ -1 \end{bmatrix} \) onto \( \vec{a} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \). What is the squared error between the projection and \( \vec{b} \), i.e. \( \| \vec{e} \|^2 = \| \text{proj}_{\vec{a}}(\vec{b}) - \vec{b} \|^2 \)?

   (b) Find the projection of \( \vec{b} = \begin{bmatrix} 1 \\ 4 \\ -5 \end{bmatrix} \) onto the subspace defined by the vectors \( \left\{ \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right\} \). What is the projection’s squared error with \( \vec{b} \), i.e. \( \| \vec{e} \|^2 = \| \text{proj}_{\vec{a}}(\vec{b}) - \vec{b} \|^2 \)?

2. Mechanical: Least Squares

   The goal of this problem is to use least squares to fit different models (i.e. equations) to a data set. Depending on the model’s number of parameters, the model will fit the data better or worse. A better model results in a lower squared error than a worse one. In part (a), we consider a linear model that contains a single slope parameter and intercepts the vertical axis at zero. In part (b), we consider a linear model with a possibly non-zero vertical axis intercept parameter, also known as an affine model.
(a) Consider the above data points. Find the linear model of the form
\[ \mathbf{a} \mathbf{x} = \mathbf{b} \]
that best fits the data, where \( x \) is a scalar that minimizes the squared error
\[ \| \mathbf{e} \|^2 = \left\| \begin{bmatrix} a_1 \\ \vdots \\ a_4 \end{bmatrix} x - \begin{bmatrix} b_1 \\ \vdots \\ b_4 \end{bmatrix} \right\|^2 = \| \mathbf{a} x - \mathbf{b} \|^2. \] (1)

**Note:** By using this linear model, we are implicitly forcing the line to go through the origin.

*You may use a calculator but show your work. Do not directly plug your numbers into IPython.*

Once you’ve computed the optimal solution \( \hat{\mathbf{x}} \), compute the squared error between your model’s prediction and the actual \( \mathbf{b} \) values as shown in Equation 1. Plot the best fit line along with the data points to examine the quality of the fit. You may either use the provided IPython notebook code to plot your best fit line or hand draw it.

(b) Let us consider a model with a (vertical) \( \mathbf{b} \)-intercept. That is, we can get a better fit for the data by assuming a(n) (affine) model of the form
\[ \mathbf{a} x_1 + x_2 = \mathbf{b}. \]

Set up a least squares problem to find the optimal \( x_1 \) and \( x_2 \) and compute the squared error between your model’s prediction and the actual \( \mathbf{b} \) values. Plot your affine model. Is it a better fit for the data? Support your answer both qualitatively by examining how close the best fit lines are to the data points and quantitatively by providing a numerical justification.

(c) **Prove the following theorem.**

Let \( \mathbf{A} \in \mathbb{R}^{m,n} \). If \( \hat{\mathbf{x}} \) is the solution to the least squares problem
\[ \min_{\mathbf{x}} \| \mathbf{A} \mathbf{x} - \mathbf{b} \|^2, \]
then the error vector \( \mathbf{A} \hat{\mathbf{x}} - \mathbf{b} \) is orthogonal to the columns of \( \mathbf{A} \). That is \( \langle \mathbf{a}_i, \mathbf{A} \hat{\mathbf{x}} - \mathbf{b} \rangle = 0 \) for \( i = 0, \ldots, n \), where \( \mathbf{a}_i \) is the \( i \)th column of \( \mathbf{A} \). In a more compact notation this is \( \mathbf{A}^T (\mathbf{A} \hat{\mathbf{x}} - \mathbf{b}) = \mathbf{0} \).
3. Trilateration With Noise!

In this question, we will explore how various types of noise affect the quality of triangulating a point on the 2D plane to see when trilateration works well and when it does not.

First, we will remind ourselves of the fundamental equations underlying trilateration.

(a) There are four beacons at the known coordinates \((x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\). You are located at some unknown coordinate \((x, y)\) that you want to determine. The distance between your location and each of the four beacons are \(d_1\) through \(d_4\), respectively. Write down one equation for each beacon that relates the coordinates to the distances using the Pythagorean Theorem.

(b) Unfortunately, the above system of equations is nonlinear, so we can’t use least squares or Gaussian Elimination to solve it. We will use the technique discussed in lecture to obtain a system of linear equations. In particular, we can subtract the first of the above equations from the other three to obtain three linear equations. Write down these three linear equations.

(c) Combine the three equations in the above system into a single matrix equation of the form

\[
A \begin{bmatrix} x \\ y \end{bmatrix} = \vec{b}.
\]

(d) Now, go to the IPython notebook. In the notebook we are given three possible sets of measurements for the distances of each beacon from the receiver:

i. **ideal_distances**: the ideal set of measurements, the true distances of our receiver to the beacons. \(d_1 = d_2 = d_3 = d_4 = 5\).

ii. **imperfect_distances**: imperfect measurements. \(d_1 = 5, d_2 = 4.5, d_3 = 5, d_4 = 5.5\).

iii. **one_bad_distances**: mostly perfect measurements, but \(d_1\) is a very bad measurement. \(d_1 = 6.5\) and \(d_2 = d_3 = d_4 = 5\).

Plot the graph illustrating the case when the receiver has received **ideal_distances** and visually solve for the position of the observer \((x, y)\). What is the coordinate?

(e) You will now set up the above linear system using IPython. Fill in each element of the matrix \(A\) that you found in part (c).

(f) Similarly, fill in the entries of \(\vec{b}\) from part (c) in the **make_b** function.

(g) Now, you should be able to plot the estimated position of \((x, y)\) using the supplied code for the **ideal_distances** observations. Modify the code to estimate \((x, y)\) for **imperfect_distances** and **one_bad_distances**, and comment on the results.

In particular, for **one_bad_distances** would you intuitively have chosen the same point that our trilateration solution did knowing that only one measurement was bad?

(h) We define the “cost” of a position \((x, y)\) to be the sum of the squares of the differences in distance of that position from the observation, as defined symbolically in the notebook. Study the heatmap of the cost of various positions on the plane, and make sure you see why \((0,0)\) appears to be the point with the lowest cost.

Now, compare the cost of \((0,0)\) with the cost of your estimated position obtained from the least-squares solution in all three cases. When does least squares do worse?

4. 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 \([2 \ 20 \ 1 \ -1 \ -1]^T\) 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:

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

(a) We wish to set up a linear model for the problem in the format \(A\vec{x} = \vec{b}\). 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 \(A\). What are your unknowns?

(b) Training data is data that is used to develop your model. Use the (fake) training data \(\text{diabetes_train.npy}\) 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 \(\text{diabetes_test.npy}\) 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?

5. Image Analysis

Applications in medical imaging often require an analysis of images based on the image’s pixels. 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 their locations and how many there are. Automatic detection of
shape is useful in image classification as well (e.g. consider a robot trying to find out autonomously where a mug is in its 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. You will do this by finding the equation of the circle or ellipse that bounds the cell relative to a given coordinate system in 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 \((x, y)\) and collect the following points:

\[(0.3, -0.69), (0.5, 0.87), (0.9, -0.86), (1, 0.88), (1.2, -0.82), (1.5, 0.64), (1.8, 0)\].

Recall that a quadratic equation of the form

\[ax^2 + bxy + cy^2 + dx + ey = 1\]

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

\[a(x^2 + y^2) + dx + ey = 1\]

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 minimization problem to do this, i.e. a least squares problem minimizing the squared error \(||\vec{A}\vec{x} - \vec{b}||\) where you attempt to find the unknown coefficients \(a\), \(d\), and \(e\) from your points. *Hint: The quantities \((x^2 + y^2), x, \text{ and } y\) can be thought of as variables calculated from your data points.*

(b) How can you find the equation of an ellipse that surrounds the cell? Provide a setup and formulate a minimization problem similar to that in part (a).

(c) In the IPython notebook, write a short program to fit a circle to the given points. What is \(\frac{||\vec{e}||}{N}\), where \(\vec{e} = \vec{A}\vec{x} - \vec{b}\) and \(N\) is the number of data points? Plot your points and the best fit circle in IPython.

(d) In the IPython notebook, write a short program to fit an ellipse to the given points. What is \(\frac{||\vec{e}||}{N}\), where \(\vec{e} = \vec{A}\vec{x} - \vec{b}\) and \(N\) is the number of data points? 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?

6. 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?