This homework is due May 1, 2020, at 23:59.
Self-grades are due May 4, 2020, 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: 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.

![Graph of data points and a grid with values a and b]

<table>
<thead>
<tr>
<th>a</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>
(a) Consider the above data points. Find the linear model of the form
\[ \vec{a}x = \vec{b} \]
that best fits the data, where \( x \) is a scalar that minimizes the squared error
\[
\| \vec{e} \|^2 = \left\| \begin{bmatrix} a_1 \\ \vdots \\ a_4 \end{bmatrix} x - \begin{bmatrix} b_1 \\ \vdots \\ b_4 \end{bmatrix} \right\|^2 = \| \vec{a}x - \vec{b} \|^2.
\] (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 \( \hat{x} \), compute the squared error between your model’s prediction and the actual \( b \) 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) \( b \)-intercept. That is, we can get a better fit for the data by assuming an affine model of the form
\[ \vec{a}x_1 + x_2 = \vec{b}. \]
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 \( \vec{b} \) 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 \( A \in \mathbb{R}^{m \times n} \). If \( \hat{x} \) is the solution to the least squares problem
\[
\min_{\vec{x}} \left\| A\vec{x} - \vec{b} \right\|^2,
\]
then \( A^T(A\hat{x} - \vec{b}) = \vec{0} \). This is called the normal equation; it says that the error in the least squares estimate is orthogonal to the columns of \( A \). You will often see the normal equation written in the form
\[ A^T A \hat{x} = A^T \vec{b}. \]

2. **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. \textit{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. \textit{imperfect_distances}: imperfect measurements. \(d_1 = 5.5, d_2 = 4.5, d_3 = 5, d_4 = 5\).

iii. \textit{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 \textit{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 \textit{make_b} function.

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

In particular, for \textit{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?

3. 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:

\[
\begin{bmatrix}
\text{age} \\
\text{weight} \\
tomosin2 \\
ts1 \\
chn1
\end{bmatrix}
\]
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:

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

(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, $\alpha_i$, 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 $\text{diabetes\_train.npy}$ 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 $\pm 1$ 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 $\pm 1$ 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?

4. 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:

$(0.3, -0.69), (0.5, 0.87), (0.9, -0.86), (1.08), (1.2, -0.82), (1.5, 0.64), (1.8, 0)$.

Recall that an 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 an 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 $||\mathbf{Ax} - \mathbf{b}||$ where you attempt to find the unknown coefficients $a$, $d$, and $e$ from your points. *Hint: The quantities $(x^2 + y^2)$, $x$, 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 $\|\mathbf{e}\|_N$, where $\mathbf{e} = \mathbf{Ax} - \mathbf{b}$ 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 $\|\mathbf{e}\|_N$, where $\mathbf{e} = \mathbf{Ax} - \mathbf{b}$ 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?

5. Constrained Least Squares Optimization

In this problem, we will guide you through solving the following optimization problem:

Consider a matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ where $M > N$ and all $N$ columns are linearly independent. Determine a unit vector $\hat{\mathbf{x}}$ that minimizes $\|\mathbf{Ax}\|_2$, where $\|\cdot\|$ denotes the norm—that is,

$$\|\mathbf{Ax}\|_2^2 \triangleq \langle \mathbf{Ax}, \mathbf{Ax} \rangle = \langle \mathbf{A}^T \mathbf{Ax} \rangle = \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}.$$ 

This is equivalent to solving the following optimization problem:

$$\min_{\mathbf{x}} \|\mathbf{Ax}\|_2^2 \quad \text{subject to the constraint} \quad \|\mathbf{x}\|_2^2 = 1.$$ 

This task may seem like solving a standard least squares problem $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{b} = \mathbf{0}$, but it is different. As an example, notice $\mathbf{x} = \mathbf{0}$ is not a valid solution to our problem because the norm of the zero vector does not equal one. Our optimization problem is a least squares problem with a constraint—hence the term *constrained least squares optimization*. The constraint can be visualized as limiting the vector $\mathbf{x}$ to lie on a unit circle (radius of the circle is one) if $N = 2$ and on a unit sphere if $N = 3$.

Let $(\lambda_1, \mathbf{v}_1), \ldots, (\lambda_N, \mathbf{v}_N)$ denote the eigenpairs (i.e., eigenvalue/eigenvector pairs) of $\mathbf{A}^T \mathbf{A}$. Assume that the eigenvalues are all real, distinct and indexed in a descending fashion—that is,

$$\lambda_1 > \cdots > \lambda_N.$$ 

Assume, too, that each eigenvector has been normalized to have unit length—that is, $\|\mathbf{v}_k\| = 1$ for all $k \in \{1, \ldots, N\}$.

(a) Show that $\lambda_N > 0$, i.e. all the eigenvalues are strictly positive. *Hint: Consider $\|\mathbf{Av}\|_2^2$*

(b) Consider two eigenpairs $(\lambda_k, \mathbf{v}_k)$ and $(\lambda_z, \mathbf{v}_z)$ corresponding to distinct eigenvalues of $\mathbf{A}^T \mathbf{A}$—that is, $\lambda_k \neq \lambda_z$. Prove that the corresponding eigenvectors $\mathbf{v}_k$ and $\mathbf{v}_z$ are orthogonal: $\mathbf{v}_k \perp \mathbf{v}_z$.

To help you get started, consider the two equations

$$\mathbf{A}^T \mathbf{A} \mathbf{v}_k = \lambda_k \mathbf{v}_k \quad (2)$$

and

$$\mathbf{v}_z^T \mathbf{A}^T \mathbf{A} = \lambda_z \mathbf{v}_z^T. \quad (3)$$
The second equation can be derived by taking the transpose of both sides in the eigenvalue equation \(A^T \mathbf{v} = \lambda \mathbf{v}\). Premultiply Equation 2 with \(\mathbf{v}_k^T\), postmultiply Equation 3 with \(\mathbf{v}_\ell\), compare the two, and explain how one may then infer that \(\mathbf{v}_k\) and \(\mathbf{v}_\ell\) are orthogonal, i.e. \(\langle \mathbf{v}_k, \mathbf{v}_\ell \rangle = 0\).

Premultiplication by a vector \(\mathbf{x}\) means multiplying an expression by \(\mathbf{x}\) on the left. For example, pre-multiplying the matrix \(A\) by \(\mathbf{x}\) gives \(\mathbf{x}A\). Postmultiplication means multiplying on the right. Remember that in general matrix-vector multiplication is not commutative, so those operations are not identical.

(c) The results of part (b) imply that the \(N\) eigenvectors of \(A^T A\) are mutually orthogonal. A basis formed by vectors that are both (1) mutually orthogonal and (2) have unit length is called an orthonormal basis. Since the eigenvalues of \(A^T A\) are distinct and have a norm of one, the eigenvectors form an orthonormal basis in \(\mathbb{R}^N\). This means that we can express an arbitrary vector \(\mathbf{x} \in \mathbb{R}^N\) as a linear combination of the eigenvectors \(\mathbf{v}_1, \ldots, \mathbf{v}_N\), as follows:

\[
\mathbf{x} = \sum_{n=1}^{N} \alpha_n \mathbf{v}_n.
\]

i. Determine the \(n\)th coefficient \(\alpha_n\) in terms of \(\mathbf{x}\) and one or more of the eigenvectors \(\mathbf{v}_1, \ldots, \mathbf{v}_N\).

ii. Suppose \(\mathbf{x}\) is a unit-length vector (i.e., a unit vector) in \(\mathbb{R}^N\). Show that

\[
\sum_{n=1}^{N} \alpha_n^2 = 1
\]

where the \(\alpha_n\)'s are the coefficients of \(\mathbf{x}\) in the basis defined earlier.

(d) Now express \(\|A\mathbf{x}\|^2\) in terms of \(\{\alpha_1, \alpha_2, \ldots, \alpha_N\}\), \(\{\lambda_1, \lambda_2, \ldots, \lambda_N\}\), and \(\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}\), and find an expression for \(\mathbf{\hat{x}}\) such that \(\|A\mathbf{x}\|^2\) is minimized. Do not use any tool from calculus to solve this problem, so avoid differentiation.

Hint: After expressing \(\|A\mathbf{x}\|^2\) in terms of \(\{\alpha_1, \alpha_2, \ldots, \alpha_N\}\), \(\{\lambda_1, \lambda_2, \ldots, \lambda_N\}\), and \(\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}\), which variable are you minimizing over?

6. (PRACTICE) 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 three ingredients: eggs, sugar and butter.

There are 6 unknowns: the amount of eggs, sugar, and butter used in Decadent Dwight cookies and the amount of eggs, sugar, and butter used in Heavenly Hearst cookies. We will denote these \(D_e \text{ egg }\) cookie, \(D_s \text{ grams sugar }\) cookie, \(H_e \text{ egg }\) cookie, \(H_s \text{ grams sugar }\) cookie, \(H_b \text{ grams butter }\) cookie.

(a) How many linearly independent equations would you need to be able to solve for all the unknowns (assuming a consistent system of equations)?
(b) Unfortunately, the team is not able to find precise information about the ingredients used in the cookie making process.

They stake-out Engineering Edibles, and they see Bob the Baker buy a dozen eggs for $2 daily and 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). They know from the supermarket that exactly 1 kg of sugar costs 5 dollars, and exactly 100 grams of butter costs 1 dollar. *(The team precisely knows amount of butter in each cookie and the exact price of sugar and butter. However, the amount of eggs and sugar in each cookie are imprecisely known.)*

All this is not enough for the team to unlock the secret recipe! One day, however, the team gets lucky and hears through the grapevine that Heavenly Hearst contains about 10 grams of sugar. *(This value is not precise, since it’s just gossip!)* Let’s also remember that the ingredients for each cookie of either variety would cost $0.2.

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 easily affected by air currents, so they get noisy observations: Decadent Dwight is about 25 grams, and Heavenly Hearst is about 24 grams. Assume that 1 egg weighs exactly 50 grams.

Update the least squares problem with this new information, and see how the values change. You should formulate two new equation in addition to the equations from the last part.

Are the new values more accurate? *(Hint: Find the sum of squared errors, $||e||^2$ then divide $||e||$ by the number of measurements to find the average error for data points for both (b) and (c).)*

7. 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?