Computer Science Department, Ben Gurion University of the Negev

Optimization Methods with Applications - Spring 2025

Home Assignment 1

Due on 10/4/2025.

1. **Minimizations with different norms lead to different answers**
   We are trying to approximate a vector $[x_1, x_2, x_3]$ by a constant c using ℓ_p norms. Assume $x_1 < x_2 < x_3$. Find the best approximation of the vector using a constant c in the following norms (Note: the x_i's are given, and you need to find c.):

   (a) ℓ_2 norm (Least squares): $\arg\min_{c\in\mathbb{R}}\{(c-x_1)^2 + (c-x_2)^2 + (c-x_3)^2\}$.

   (b) ℓ_∞ norm: $\arg\min_{c\in\mathbb{R}}\{\max(|c-x_1|, |c-x_2|, |c-x_3|)\}$.

   (c) ℓ_1 norm: $\arg\min_{c\in\mathbb{R}}\{|c-x_1| + |c-x_2| + |c-x_3|\}$.
   Hint for (b) and (c): the solution is obtained by logic, not by calculations as in (a).

   (d) Understand that if $\mathbf{x}$ had n variables: $x_1, ..., x_n$, then your answer to (a) would be the mean and for (c) it would be the median. Explain why.

2. **Eigenvalues and positive definite matrices**
   (not for submission - only for practice, more or less answered during class)

   (a) Show that by definition, for any matrix A, the matrix $A^\top A$ is symmetric and positive semi-definite.

   (b) Show that for any matric C, if λ is an eigenvalue of C then $1+\lambda$ is an eigenvalue of the matrix $I + C$.

   (c) Suppose that $A \in \mathbb{R}^{m\times n}$, $m \geq n$. Show that A is full rank if and only if $A^\top A$ is invertible.

   (d) Suppose that $A \in \mathbb{R}^{m\times n}$, $m \geq n$. Show that A is full rank if and only if $A^\top A$ is symmetric and positive definite (you can use the previous section).

   (e) Show that if $\alpha > 0$, then the matrix $A^\top A + \alpha I$ is always positive definite (I-the identity matrix).

3. **The existence of solution for Least Squares**

   (a) Recall the definition of the null-space: $null(A) = \{\mathbf{x} : A\mathbf{x} = 0\}$. Show that $null(A^T A) = null(A)$. Hint: $A^T A\mathbf{x} = 0 \Rightarrow \mathbf{x}^T A^T A\mathbf{x} = 0$.

   (b) Recall the definition of range: $range(A) = \{\mathbf{y} : A\mathbf{x} = \mathbf{y}\}$. Show that $range(A^T A) = range(A^T)$.
   Hint: You may have learned that for every matrix B, $range(B^T) = null(B)^\perp$ where $^\perp$ denotes the orthogonal complement of the subspace. In other words: $range(B^T)$ is the subspace of all vectors in $\mathbb{R}^n$ that are orthogonal to all the vectors in $null(B)$. Use this for $B = A^T A$ together with the previous section.

   (c) Using the previous section, conclude that the normal equation is always consistent — i.e. there is always at least one solution to the Least Squares problem.

4. **Least Squares**

   (a) Find the best approximation in a least square sense for the system $A\mathbf{x} \approx \mathbf{b}$ where

$$A = \begin{bmatrix} 2 & 1 & 2 \\ 1 & -2 & 1 \\ 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix} \quad , \mathbf{b} = \begin{bmatrix} 6 \\ 1 \\ 5 \\ 2 \end{bmatrix}. \tag{1}$$

   Write the normal equations and solve the problem using a computer. You may use built-in functions and provide the code.

   (b) Is the solution $\mathbf{x}^*$ that you found in the previous section unique? Explain. What is the minimal objective (loss) value $\|A\mathbf{x}^* - \mathbf{b}\|_2^2$?

   (c) Compute the residual of the least squares system $\mathbf{r} = A\mathbf{x}^* - \mathbf{b}$, with $\mathbf{x}^*$ that you found in the previous section. Show that $A^\top \mathbf{r} = 0$. Is that surprising?

   (d) Find the least squares solution of the system in Eq. (1), but now find a solution for which the second equation is almost exactly satisfied (let's say, such that $|r_2| < 10^{-3}$). Hint: use weighted least squares.

   (e) Find the least squares solution of the system in Eq. (1), but now add simple Tikhonov regularization term $\lambda\|\mathbf{x}\|_2^2$ with $\lambda = 0.5$.

5. **Working with real data with ordinary least squares**
In this task we will develop a linear model to help an insurance company identify the medical cost of a patient given a set of parameters. The data consists with the columns

Age	Sex	BMI	Childern	Smoker	Region (in US)	Charges in USD

We wish to find a linear model to predict the charges in USD given the 6 parameters in the table. That is,

$$\text{Charges in USD} \approx \alpha_0 + \alpha_1 \cdot \text{age} + \alpha_2 \cdot \text{Sex} + \alpha_3 \cdot \text{BMI} + \alpha_4 \cdot \text{Childern} + \alpha_5 \cdot \text{Smoker} + \alpha_6 \cdot \text{Region}$$

We will try to minimize the cost is a least squares sense, i.e., minimize the mean squared difference between the left and right sides above (a.k.a Mean Squared Error - MSE).

(a) Read the data in the CSV table into a Python program. You may use the code below:

```
import pandas  as pd #Data manipulation
import numpy as np #Data manipulation
import matplotlib.pyplot as plt # Visualization

path = '../input/'
df = pd.read_csv(path+'insurData.csv')
print('\nNumber of rows and columns in the data set: ',df.shape)
print('')

#Lets look into top few rows and columns in the dataset
df.head()
```

(b) Process the data:

i. Luckily there are no missing entries in the data, but since we have α_0 in our LS model, we'll have to add a column of 1's to have the data conveniently as a matrix.

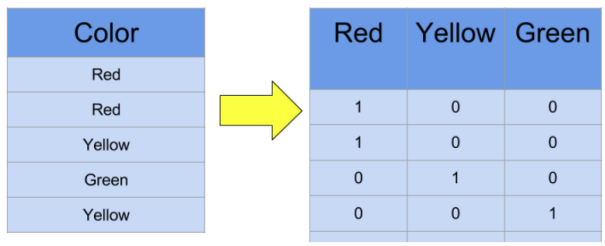
ii. Also, some entries are much larger than others, which may create numbers on very different scales which are hard to interpret. Define the charges in thousands of dollars (divide the charges data by 1000).

iii. Next - there are categorial data such as smoker (yes/no), region, sex (male/female) etc. If we have only two options (like in smoker), it makes sense to add '0' or '1' as a numeric data and it does not matter which is labeled as '0' and which is labeled as '1'. However, for three or more options, it does not make sense to add '0','1', and '2' and to multiply such entries by a variable α - since this way, unlike in the binary case, we create a non-existent order between the

data variables. See explanation here:
`https://www.youtube.com/shorts/Dz8zNQNW9RQ?feature=share`.
To fix that, replace the entries by one-hot encodings such that:

$$\begin{aligned}\text{Charges (USD)} \approx\ & \alpha_0 + \alpha_1 \cdot \text{Age} + \alpha_2 \cdot \text{Sex} + \alpha_3 \cdot \text{BMI} + \alpha_4 \cdot \text{Childern} + \alpha_5 \cdot \text{Smoker} \\ & + \alpha_6 \cdot \text{Region1} + \alpha_7 \cdot \text{Region2} + \alpha_8 \cdot \text{Region3} + \alpha_9 \cdot \text{Region4}\end{aligned}$$

so that the columns Sex and Smoker have 1's and 0's, and Region1-4 always have a single "1" and the rest are zeros. An example of such a conversion for "one-hot encoding" is illustrated here:



Note: you should not do this manually in the file. Use code. You can read more here:
`https://www.kaggle.com/code/dansbecker/using-categorical-data-with-one-hot-encoding/notebook`

iv. For the sake of a reference model we will define MSE_0 as the MSE of an optimal constant model: $MSE_0 = \min_{\alpha_0} \frac{1}{m}\|\alpha_0 - \mathbf{y}\|_2^2$ where $\mathbf{y}$ are the charges. This will measure the error if we set a constant charge for all customers. For the sake of a reference, you can compute it for the whole data without a split (details below).

(c) Now, we find the parameters α_i based on a given data, but we really wish to check wether the prediction we find holds for future customers. To verify that, we will need to measure the error of the evaluation on data (rows / customers) which were not used for evaluating the parameters. For this purpose, we will randomly split the rows into two sets: "train" and "tests", where 80% of the entries are used for finding the model parameters, and 20% are used for evaluation later. More about that will be discussed later in the course, but you can also see here, ignoring the subtle difference between validation and test which we will just neglect in our simple model here: `https://www.youtube.com/watch?v=dSCFk168vmo`).
For each experiment you should choose 80% of the rows in the data as "training data" and solve the normal equations based on them to get an estimation of $\alpha_0....\alpha_9$. Using those you should compute the MSE of the training data - that's the objective you minimized. Then, using the other 20% of the samples (test data), which you did not use to compute $\alpha_0,...,\alpha_9$, you should again compute the

MSE (note the division by the different number of samples). In both cases display the relative MSE, i.e., $\frac{MSE}{MSE_0}$ to see the improvement in percentages compared to a constant model. Repeat this 10 times with different random selections and see that the results are consistent. Write them in a table, and see if the numbers you get are similar between the testing and the training data. Does the model predict the charges well in your opinion?
Note:

- Denoting the data split for the train/test data for your least squares in $X \in \mathbb{R}^{m \times n}$ and $\mathbf{y}$, the mean squared error (MSE) is defined by $\frac{1}{m}\|X\alpha - \mathbf{y}\|_2^2$
- In some cases you get a singular matrix when trying to invert the matrix. Hence, in this question you should use a Tikhonov regularization with a small parameter λ that works well for you (for general knowledge - this λ can be thought of a hyper-parameter that was mentioned in the videos above, but we can ignore that for now).
- You may use
  `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html`
  or
  `https://numpy.org/doc/stable/reference/random/generated/numpy.random.permutation.html`
  to generate a random permuation of the row indices and then choose the first 80% and last 20% of the permuted data rows.

(d) Repeat the previous section, only now don't use the region and smoker data (i.e., compute the model only for $\alpha_0, ..., \alpha_4$). Are the results (relative MSE errors) better or worse? Is it expected? Focus on the training error in your answer. The test is a bit more subtle and we'll talk about it later in the course.