Computer Science Department, Ben Gurion University of the Negev

Introduction to Data Science - Winter 2023

Home assignment 1

Due on Dec 11th.

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**

   (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. **Least Squares**

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

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

   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 $\lambda = 0.5$.

4. **Frobenius Norm.**
   See the definition of Frobenius Norm:
   `https://mathworld.wolfram.com/FrobeniusNorm.html`
   This norm is often used to compare data sets, or apply manipulations on data sets. It is easy to see that by definition

   $$\|A\|_F^2 = \sum_{i,j}(a_{ij})^2 = \sum_j \|\mathbf{a}_j\|_2^2$$

   where $\mathbf{a}_j$ is a column of A.

   (a) Suppose that we want to solve the problem

   $$\arg\min_{X \in \mathbb{R}^{n\times n}} \|AX - B\|_F^2$$

   where $A, B, X \in \mathbb{R}^{n\times n}$. Find an expression for the solution. When is the solution unique? Hint: this is almost the same as standard least squares.

   (b) Often, we wish to make data A look similar to data B of the same size, to remove artifacts that are not related to the phenomena that we test. For example, when we measure gene expression values, often there's a bias of the particular lab's measurement. To correct that, we to make the datasets similar, and one option is to solve a minimization of the following form (this is a bit simplified)

   $$\arg\min_{D \in \mathbb{R}^{n\times n}, D \text{ is diagonal}} \|DA - B\|_F^2$$

D is a diagonal matrix. In other words, we wish to find a scale D_{ii} for each row of A (denoted by $\mathbf{a}_i$) so that $\|\mathbf{a}_i D_{ii} - \mathbf{b}_i\|_2^2$ is minimized ($\mathbf{b}_i$ is a row in B). Use a computer and find the solution D for the following matrices:

$$A = \begin{bmatrix} 5 & 6 & 7 & 8 \\ 1 & 3 & 5 & 4 \\ 1 & 0.5 & 4 & 2 \\ 3 & 4 & 3 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0.57 & 0.56 & 0.8 & 1 \\ 1.5 & 4 & 6.7 & 4.9 \\ 0.2 & 0.1 & 1 & 0.6 \\ 11 & 30 & 26 & 10 \end{bmatrix}$$

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, and minimize the squared difference between the left and right sides above.

(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: Luckily there are no missing entries in the data, but since we have α_0 in our model, we’ll have to add a column of 1’s. 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 (divide by 1000).

Next - there are some categorial data such as smoker (yes/no), region, sex (male/female) etc. It does not make sense to multiply such entries by a variable α. Replace the entries by one-hot encodings such that:

$$\begin{aligned} \text{Charges in USD} \quad \approx \quad & \alpha_0 + \alpha_1 \cdot \text{Age} + \alpha_2 \cdot \text{Male} + \alpha_3 \cdot \text{Female} + \alpha_4 \cdot \text{BMI} \\ & + \alpha_5 \cdot \text{Childern} + \alpha_6 \cdot \text{Smoker} + \alpha_7 \cdot \text{Non-smoker} \\ & + \alpha_8 \cdot \text{Region1} + \alpha_8 \cdot \text{Region2} + \alpha_8 \cdot \text{Region3} + \alpha_8 \cdot \text{Region4} \end{aligned}$$

so that the columns Male/Female and Smoker/Non-smoker always have ”1” and ”0”, and Region1-4 always have a single ”1” and the rest are zeros. Note: you should not do this manually. Use code. You can read more here: `https://www.kaggle.com/code/dansbecker/using-categorical-data-with-one-hot-encoding/notebook`

(c) We 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).
Run 5 experiments where each time you split the data randomly, solve the Least Squares problem and find α using the “train” data only, and compute the mean squared error (MSE, see below) for both the train and test data. Compare the MSE of the train and test (don’t forget to normalize). Does the model predict the charges well?
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 $\frac{1}{m}\|X\alpha - \mathbf{y}\|_2^2$
- You may use
`https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html`
or
`https://docs.scipy.org/doc/numpy-1.15.1/reference/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) Show a graph of the distribution of error values (whatever the model cannot predict) for one of the train sets (see `matplotlib.pyplot.hist`).