Introduction to Machine Learning (67577)

Exercise 2
Linear Regression

Semester B, 2020

Theoretical Questions

Let X be the design matrix of a linear regression problem with p rows (variables) and n columns (samples). Let $y \in \mathbb{R}^n$ be the response vector corresponding the samples in X. Recall that for some vector space $V \subseteq \mathbb{R}^p$ the orthogonal complement of V is: $V^{\perp} = \{x \in \mathbb{R}^p | \langle x, v \rangle = 0 \ \forall v \in V\}$

Solutions of the Normal Equations

1. Prove that: $Ker\left(X^{\top}\right) = Ker\left(XX^{\top}\right)$
2. Prove that for a square matrix A: $Im\left(A^{\top}\right) = Ker\left(A\right)^{\perp}$
3. Let $y = X^{\top}w$ be a non-homogeneous system of linear equations. Assume that $X^{\top}$ is square and not invertible. Show that the system has ∞ solutions $\Leftrightarrow$ $y \perp Ker\left(X\right)$.
4. Consider the (normal) linear system $XX^{\top}w = Xy$. Using what you have proved above prove that the normal equations can only have a unique solution (if $XX^{\top}$ is invertible) or infinitely many solutions (otherwise).

Projection Matrices

5. In this question you will prove some properties of orthogonal projection matrices seen in recitation 1. Let $V \subseteq \mathbb{R}^d$, $dim\left(V\right) = k$ and let $v_1, \ldots, v_k$ be an orthonormal basis of V. Define the orthogonal projection matrix $P = \sum_{i=1}^{k} v_i v_i^{\top}$ (Notice this is an outer product). Show that:
   (a) P is symmetric
   (b) The eigenvalues of P are 0 or 1 and that $v_1, \ldots, v_k$ are the eigenvectors corresponding the eigenvalue 1
   (c) $\forall v \in V \ \ Pv = v$
   (d) $P^2 = P$
   (e) $\left(I - P\right)P = 0$

Least Squares

Given a sample $S = ((\mathbf{x}_i, y_i))_{i=1}^m$, the ERM rule for linear regression w.r.t. the squared loss is

$$\hat{\mathbf{w}} \in \underset{\mathbf{w}\in\mathbb{R}^d}{\operatorname{argmin}} \; \|X^\top \mathbf{w} - \mathbf{y}\|^2 ,$$

where X is the design matrix of the linear regression with columns as samples and y the vector of responses. Let $X = U\Sigma V^\top$ be the SVD of X, where U is a $d \times d$ orthonormal matrix, Σ is a $d \times m$ diagonal matrix, and V is an $m \times m$ orthonormal matrix. Let $\sigma_i = \Sigma_{i,i}$ and note that only the non-zero σ_i-s are singular values of X. Recall that the pseudoinverse of X is defined by $X^\dagger = V\Sigma^\dagger U^\top$ where $\Sigma^\dagger$ is an $m \times d$ diagonal matrix, such that

$$\Sigma^\dagger_{i,i} = \begin{cases} \sigma_i^{-1} & \sigma_i \neq 0 \\ 0 & \sigma_i = 0 \end{cases}$$

Since $\left(X^\top\right)^\dagger = \left(X^\dagger\right)^\top$ (verify this), we can simplify the notation by using $X^{\top\dagger}$ for the pseudoinverse of $X^\top$. You have seen in class that if $XX^\top$ is invertible, $\hat{\mathbf{w}} = \left(XX^\top\right)^{-1} X\mathbf{y}$. In recitation we released this assumption, and showed that $\hat{\mathbf{w}} = X^{\top\dagger}\mathbf{y}$ is always a solution.

6. We will first show that if $XX^\top$ is invertible, the general solution we derived in recitation is equal to the solution you have seen in class. For this part, assume that $XX^\top$ is invertible.
   - Show that $\left(XX^\top\right)^{-1} = UD^{-1}U^\top$, where $D = \Sigma\Sigma^\top$.
   - Use this to show that $\left(XX^\top\right)^{-1} X = X^{\top\dagger}$.
7. Show that $XX^\top$ is invertible if and only if $\operatorname{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} = \mathbb{R}^d$.
8. Recall that if $XX^\top$ is not invertible then there are many solutions. Show that $\hat{\mathbf{w}} = X^{\top\dagger}\mathbf{y}$ is the solution whose L_2 norm is minimal. That is, show that for any other solution $\overline{\mathbf{w}}$, $\|\hat{\mathbf{w}}\|_2 \leqslant \|\overline{\mathbf{w}}\|_2$ (Why is it ok to do so?)

Hints

- Recall that the rank of X and the rank of $XX^\top$ are determined by the number of singular values of X. If you're not sure why this is true, go over recitation 1.
- Instead of comparing the norms of $\hat{\mathbf{w}}$ and $\overline{\mathbf{w}}$ compare those of $U^\top\hat{\mathbf{w}}$ and $U^\top\overline{\mathbf{w}}$.

Practical Questions

In the following section you will implement a linear regression model. Then you will use it, to train (fit) a model and test it over a real-world dataset of house prices. As this is a known published dataset, the instance you will be working with is different than the published ones.

9. In a file called *linear_model.py*, implement a function named 'fit_linear_regression' that receives the following parameters: a design matrix 'X' (numpy array with p rows and n columns) and a response vector 'y' (numpy array with n rows). The function returns two sets of values: the first is a numpy array of the **coefficients vector 'w'** (think what should its dimension be) and the second is a numpy array of the **singular values of X**. You are expected to implement the linear regression **yourself** (recall the normal equations seen in class and recitation). You **cannot** use any library that solves a linear regression problem (and doing so will result in a zero grade for the question). You are allowed (and encouraged) to use different function from *numpy* and *numpy.linalg*.

10. In the *linear_model.py* file, implement a function named 'predict' that receives the following parameters: a design matrix 'X' (numpy array with p rows and m columns) and coefficients vector 'w'. The function returns a numpy array with the predicted value by the model.

11. In the *linea_model.py* file, implement a function named 'mse' that receives a response vector and a prediction vector (both numpy arrays) and returns the MSE over the received samples. Reminder:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x_i^\top w - y_i\right)^2$$

12. Download the file *kc_house_data.csv* from the moodle and get familiar with the data. See what are the different features it contains and think what are valid values for each feature. Can house prices be negative? Can a living room size be so small? Check out the meta-data (including explanation of each feature) on Kaggle.
Implement a function called 'load_data' the given a path to the csv file loads the dataset and performs all the needed preprocessing so to get a valid design matrix. The function returns the dataset after the preprocessing.

13. In the given dataset you will find categorical features. These are features that have no apparent logical order to their values (is one zip-code greater/smaller than another?, is one color greater/smaller than another?). The simplest approach is to change those categorical features into One Hot encoding (or "dummy variables"). For example if you have a feature with the values "man/woman/other" you could define two binary features for "man" and "woman". So instead of one feature of t categories, we want $t-1$ binary features. For the implementation of this you may use any python package that you want (or code it yourself). Here is a nice reference for how to deal with categorical data (you can use methods 1 and 2 therein): **Dealing with categorical variables**

    - Address the categorical features in the 'load_data' function.

- In your submitted PDF explain what were the features you found to be categorical.

14. Implement a function called 'plot_singular_values' that receives a collection of singular values and plots them in descending order. That is: x-axis a running index number and y-axis the singular value's value. This kind of plot is called a scree-plot.

15. Putting it all together 1: Add to the *linear_model.py* file code that loads the dataset, performs the preprocessing and plots the singular values plot. In your answers PDF describe what can be learned from the singular values. Also is the design matrix (X) close to being singular or not?

16. Putting it all together 2: Next we will fit a model and test it over the data. Begin with writing code that splits the data into train- and test-sets randomly, such that the size of the test set is 1/4 of the total data.
Next, over the 3/4 of the data, considered as training set perform the following: For every $p \in \{1, 2, ..., 100\}$ fit a model based on the first $p\%$ of the training set. Then using the 'predict' function test the performance of the fitted model on the test-set.

    - Add to your answers PDF the plot of the MSE over the test set as a function of $p\%$.
    - Explain the results you got.

17. Basics in feature selection: In the *linear_model.py* file, implement a function named 'feature_evaluation'. This function, given the design matrix and response vector, plots for every non-categorical feature, a graph (scatter plot) of the feature values and the response values. It then also computes and shows on the graph the Pearson Correlation between the feature and the response. The graph's title should include information about what feature is tested in that graph.

$$Pearson\,Correlation\;\; \rho := \frac{COV\,(vector_1, vector_2)}{\sigma_{vector_1} \cdot \sigma_{vector_2}}$$

You are allowed to use functions that calculate the standard deviation and co-variance, but not functions that calculate the correlation itself.
Choose two features, one that seems to be beneficial for the model and one that does not. In your answers PDF add the graphs of these two chosen features and explain how do you conclude if they are beneficial or not.

Exponential Regression

In this section we'll analyze the rate of infection of COVID19.

18. Create a file named "covid19.py" and copy the function 'fit_linear_regression' from the previous section. Download the file "covid19_israel.csv" from the moodle and read it into pandas' DataFrame. You should get three columns - **day_num**, which count the days since first infection was identified in Israel, **date**, which represents the time of the event, and **detected**, which sums the number of detected cases up to this date.

19. Create a new column in the DataFrame named "log_detected" which holds the log of the "detected" column.

20. Use the function ‘fit_linear_regression‘ function from before to fit the "log_detected" cases to "day_num".

21. Plot two graphs: the first is the "log_detected" as a function of "day_num", and the second is "detected" as a function of "day_num". On each graph, add the data as single points, and add the fitted curve that you've estimated (think how to convert the linear result into an exponential result). Add those graphs to your answers PDF.

22. Given a sample (x, y) and a weight $w \in \mathbb{R}$, in this question we used the following loss function:
$$L_{exp}(f_w, (x, y)) = (\langle w, x\rangle - \log(y))^2.$$
If we want to use the least squares loss we saw in class for the exponential regression scenario (i.e., using y instead of $\log(y)$), how should the loss look like? How will you find the ERM solution in that case? (explain shortly the steps you'll preform, you do not need to provide a closed-form solution for the estimator). Answer in the submitted PDF.