Introduction to Machine Learning (67577)

Exercise 2
Linear Regression

Second Semester, 2021

Contents

1 Theoretical Questions

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

1.1 Solutions of the Normal Equations

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

1.2 Projection Matrices

5. In this question you will prove some properties of orthogonal projection matrices seen in recitation 1. Let $V \subseteq \mathbb{R}^d$, $dim(V) = k$ 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)
   (a) Show that P is symmetric.
   (b) Prove that the eigenvalues of P are 0 or 1 and that $v_1, \ldots, v_k$ are the eigenvectors corresponding the eigenvalue 1.
   (c) Show that $\forall v \in V \;\; Pv = v$.
   (d) Prove that $P^2 = P$.
   (e) Prove that $(I - P)P = 0$.

1.3 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

$$\widehat{w} \in \underset{\mathbf{w} \in \mathbb{R}^d}{argmin} \; ||\mathbf{X}\mathbf{w} - \mathbf{y}||^2$$

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

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

6. Show that if $\mathbf{X}^{\top}\mathbf{X}$ is invertible, the general solution we derived in recitation equals to the solution you have seen in class. For this part, assume that $\mathbf{X}^{\top}\mathbf{X}$ is invertible.
7. Show that $\mathbf{X}^{\top}\mathbf{X}$ is invertible if and only if $span\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} = \mathbb{R}^d$.
8. Recall that if $\mathbf{X}^{\top}\mathbf{X}$ is not invertible then there are many solutions. Show that $\widehat{\mathbf{w}} = \mathbf{X}^{\dagger}\mathbf{y}$ is the solution whose L_2 norm is minimal. That is, show that for any other solution $\overline{\mathbf{w}}$, $||\widehat{\mathbf{w}}|| \leq ||\overline{\mathbf{w}}||$.

Hints:

- Recall that the rank of $\mathbf{X}$ and the rank of $\mathbf{X}^\top\mathbf{X}$ are determined by the number of singular values of $\mathbf{X}$. If you are not sure why this is true, go over recitation 1.
- Which coordinates must satisfy $\widehat{w}_i = \overline{w}_i$? What is the value of $\widehat{w}_i$ for the other coordinates? If you are not sure, go back to the derivation of $\widehat{\mathbf{w}}$ (see recitation 4).

2 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 called 'fit_linear_regression' that receives the following parameters: a design matrix '$\mathbf{X}$' (numpy array with m rows and d columns) and a response vector '$\mathbf{y}$' (numpy array with m 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 called 'predict' that receives the following parameters: a design matrix '$\mathbf{X}$' (numpy array with m rows and d columns) and coefficients vector '$\mathbf{w}$'. The function returns a numpy array with the predicted values by the model.

11. In the *linear_model.py* file, implement a function called 'mse' that receives a response vector and a prediction vector (both numpy arrays) and returns the MSE over the received samples. Reminder:
$$MSE := \frac{1}{m}\sum_{i=1}^{m}(\widehat{y}_i - y_i)^2 = \frac{1}{m}\sum_{i=1}^{m}\left(\mathbf{x}_i^\top\mathbf{w} - y_i\right)^2$$

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 named '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 dataet after the preprocessing. It is up to you to decide if to return a tuple of the design matrix and responses separately or to return a single pandas 'DataFrame' to be split later.

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). You can also have a look at the following StackOverflow question for dealing 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 named ‘plot_singular_values‘ that receives a collection of singular values and plots them in descending order. That is: x-axis from 1 to n, where n is the number of singular values, and y-axis the singular values’ value. This kind of plot is called a **scree-plot**. If you’ve already drawn with feature names - We’ll accept that as well.

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 matrix 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. Be sure to not loose which sample is connected to which response.

    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 called ‘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.

$$\text{Pearson Correlation} \quad \rho := \frac{COV\left(vector_1, vector_2\right)}{\sigma_{vector_1} \cdot \sigma_{vector_2}}$$

    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.