Introduction to Machine Learning (67577)

Exercise 2
Linear Regression and PAC

April 2019

Contents

Linear Regression

Normal Equations

In the this question, you will prove a sequence of lemmas, showing that the normal equations can only have a unique solution or ∞ solutions.

You may use the fact that:

$$Ker(X^\top) = Ker(X^\intercal X)$$

Let $V \subseteq \mathbb{R}^d$ be a vector space. Recall the definition $V^\perp$ "the orthogonal complement" of V:

$$V^\perp = \left\{ \mathbf{x} \in \mathbb{R}^d | \, \langle \mathbf{x}, \mathbf{v} \rangle = 0 \quad \forall \mathbf{v} \in V \right\}$$

Prove the following lemmas (using slightly different notation in 1 and 2):

1. (3 points) $Im(A) = Ker\left(A^\top\right)^\perp$

2. (3 points) Let $A \in \mathbb{R}^{d \times d}$ be a square matrix and $\mathbf{x}, \mathbf{b} \in \mathbb{R}$ be vectors. Consider the nonhomegeneous system of linear equations $A\mathbf{x} = \mathbf{b}$. Assume that A is not invertible. Then the system has ∞ solutions if and only if $\mathbf{b} \perp Ker(A^\intercal)$.

3. (4 points) **Back to linear regression** Back to the notation of linear regression, we are interested in the (normal) linear system

$$XX^\top \mathbf{w} = X\mathbf{y}$$

, i.e. relating the previous items to the language of linear regression: $A = XX^\top$ and $X\mathbf{y} = \mathbf{b}$. Use the above lemmas to prove that the normal equations can only have a unique solution (if $XX^\top$ is invertible) or ∞ solutions (otherwise).

Least Squares Solution

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 $X = \begin{bmatrix} | & & | \\ \mathbf{x}_1 & \cdots & \mathbf{x}_m \\ | & & | \end{bmatrix}$ and $\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$.

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 the tirgul we released this assumption, and showed that $\hat{\mathbf{w}} = X^{\top\dagger}\mathbf{y}$ is always a solution.

4. a) (3 points) We will first show that if $XX^\top$ is invertible, the general solution we derived in the tirgul 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 $\left(XX^\top\right)^{-1} X = X^{\top\dagger}$.

   b) (3 points) Show that $XX^\top$ is invertible if and only if $\text{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} = \mathbb{R}^d$.

   c) (4 points) 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 $\overline{\mathbf{w}}$, $\|\hat{\mathbf{w}}\|_2 \leqslant \|\overline{\mathbf{w}}\|_2$.

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 Tirgul 1.
- Which coordinates must satisfy $\hat{w}_i = \overline{w}_i$? What is the value of $\hat{w}_i$ for the other coordinates? If you're not sure, go back to the derivation of $\hat{\mathbf{w}}$ (see Tirgul 3).

A practical task - Price Prediction

In the following task, you will use the Linear Regression technique to predict sale prices for houses. This is a famous data-set, used in different data case-studies and competitions. **The data you receive is similar - but not identical!**.
For this programming task we want you to practice what you have seen in class: inverse or SVD/EVD methods. you may use any python packages (e.g. linear algebra packages) **except of those solving linear regression problems**. To be more clear: do not use any package that its description includes any of these terms: least square, data fitting, optimize, least, square, regression, OLS, model.

- **Download** the file *kc_house_data.csv* from the moodle.
- **Get familiar with the data.** Always, when approaching a new data set, it is highly recommended to open the raw file, check what each field means, and take some basic descriptive statistics. Systematically look for corrupted data: missing entries, nans, and values that are far from the average. Can a house price be negative? Do all houses have ID? can the living room size be soooo small?
  Check out the meta-data (including explanation of each feature) on Kaggle.
  As this is a known data-set, you can read and see some basic stats and advanced analysis of the data online, for instance here.
  Remember that the data you received is not identical to the one you will find on-line.
- **Categorical Feature.** Categorical data is very convenient for people but very hard for most machine learning algorithms. For instance: colors, zip-code etc. On one hand, it makes no sense to look for linear function from zip-code to prices, but on the other hand, the zip-code is clearly affecting the house's value. How should we deal with such categorical features?
  The simplest approach is to change those categorical features into One Hot encoding (or "dummy variables"). For instance a field of "man/woman/other" can be changed into three binary features. Though, if there are 3 categories, we actually don't want

all three (as each one is always linearly dependent of the other two).
To conclude: 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

- Define $\mathbf{y}$ - the house prices.
  Define X the dataset, after your cleaning and pre-processing.
  We are looking for a vector $\mathbf{w}$ such that $\hat{\mathbf{y}} = X^\top \mathbf{w}$ is the best linear approximation for $\mathbf{y}$.

5. (5 points) Pre-processing: how did you deal with categorical features?

   Use the EVD/SVD technique showed in class to answer the following questions:

6. (5 points) Some features may be more relevant than others. Give an example of two features that are very relevant for predicting house prices and two features that are not so relevant. Explain your answer.

7. (5 points) Some features may be highly correlated with other features, making the matrix $XX^\top$ close to singular and thus we should be aware of numerical stability issues. Is the matrix $XX^\top$ close to singular? *Hint:* look at the singular values. If so, can you find features that are highly correlated (i.e. close to being co-linear) and who are they?

8. (15 points)

   a) For each x between $1-99$ (99 points), randomly partition the data into a training set of $x\%$ and a test set $(100-x)\%$.

   b) Train your data-set over x and return:

      – Your train-error

      – Your test-error

   c) Plot a graph of train and test errors as a function of x, and add the plot to your submission pdf.

VC Dimension

Pairity function

9. (10 points) Let $\mathcal{X} = \{0,1\}^n$ and $\mathcal{Y} = \{0,1\}$, for each $I \subseteq [n]$ define the parity function:

$$\forall x \in \mathcal{X} \quad h_I(\mathbf{x}) = \left(\sum_{i \in I} x_i\right) \bmod 2.$$

What is the VC-dimension of the class $\mathcal{H}_{parity} = \{h_I \mid I \subseteq [n]\}$? Prove your answer, you may use results we proved in class.

k intervals

10. Given an integer k, let $([a_i, b_i])_{i=1}^{k}$ be any set of k intervals on $\mathbb{R}$ and define their union $A = \cup_{i=1}^{k}[a_i, b_i]$. The hypothesis class $\mathcal{H}_{k-intervals}$ includes the functions:

$$h_A(x) = \begin{cases} 0 & x \notin A \\ 1 & x \in A \end{cases},$$

for all choices of k intervals.

a) (5 points) For a fixed k, find the VC-dimension of $\mathcal{H}_{k-intervals}$ and prove your answer.

b) (5 points) Show that if we let A be any finite union of intervals (i.e. k is unlimited), then the resulting class $\mathcal{H}_{intervals}$ has VC-dimension ∞.

Non-homogeneous half-spaces

11. (10 points) The hypothesis class of non-homogeneous half-spaces over $\mathbb{R}^d$ is defined as

$$HS_d = \left\{ h_{\mathbf{w},b} : \mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R} \right\} ,$$

where $h_{\mathbf{w},b}(\mathbf{x}) = \text{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right)$. Show that the VC-dimension of HS_d is $d+1$.

12. (optional) Let $\mathcal{H}$ be a class for binary classification over a domain $\mathcal{X}$. Show that if there is a function $\phi : \mathcal{X} \to \mathbb{R}^d$ such that for every $h \in \mathcal{H}$ there are $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ for which $h(x) = h_{\mathbf{w},b}(\phi(x))$, then $\text{VCdim}(\mathcal{H}) \leqslant d+1$.

Agnostic PAC

13. Let $\mathcal{H}$ be a hypothesis class of binary classifiers. Show that:

a) (5 points) if $\mathcal{H}$ is agnostic PAC learnable, then $\mathcal{H}$ is PAC learnable as well.

b) (5 points) if A is a successful agnostic PAC learner for $\mathcal{H}$, then A is also a successful PAC learner for $\mathcal{H}$.

14. (optional) Let $\mathcal{H}$ be a hypothesis class over a domain $Z = \mathcal{X} \times \{\pm 1\}$, and consider the 0-1 loss function. Assume that there exists a function $m_{\mathcal{H}}(\epsilon, \delta)$, for which it holds that for every distribution $\mathcal{D}$ over Z there is an algorithm A with the following property: when running A on $m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples drawn from $\mathcal{D}$, it is guaranteed to return, with probability at least $1-\delta$, a hypothesis $h_S : \mathcal{X} \to \{\pm 1\}$ with $L_{\mathcal{D}}(h_S) \leqslant \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$. Is $\mathcal{H}$ agnostic PAC learnable? Explain.

Monotonicity

Monotonicity of Sample Complexity

15. (5 points) Let $\mathcal{H}$ be a hypothesis class for a binary classification task. Suppose that $\mathcal{H}$ is PAC learnable and its sample complexity is given by $m_{\mathcal{H}}(\cdot, \cdot)$. Show that $m_{\mathcal{H}}$ is

monotonically non-increasing in each of its parameters. That is, show that given $\delta \in (0,1)$, and given $0 < \epsilon_1 \leqslant \epsilon_2 < 1$, we have that $m_{\mathcal{H}}(\epsilon_1, \delta) \geqslant m_{\mathcal{H}}(\epsilon_2, \delta)$. Similarly, show that given $\epsilon \in (0,1)$, and given $0 < \delta_1 \leqslant \delta_2 < 1$, we have that $m_{\mathcal{H}}(\epsilon, \delta_1) \geqslant m_{\mathcal{H}}(\epsilon, \delta_2)$.

Monotonicity of VC-Dimension

16. (5 points) Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two classes for binary classification, such that $\mathcal{H}_1 \subseteq \mathcal{H}_2$. Show that $\mathrm{VCdim}(\mathcal{H}_1) \leqslant \mathrm{VCdim}(\mathcal{H}_2)$.