CS 189 Introduction to Machine Learning
Fall 2019 Jennifer Listgarten & Stella Yu HW02

Due: Tuesday, Sept 17

0 Getting Started

Read through this page carefully. You may typeset your homework in latex or submit neatly handwritten/scanned solutions. Please start each question on a new page. Deliverables:

1. Submit a PDF of your writeup, **with an appendix for your code**, to assignment on Gradescope, "HW[X] Write-Up". If there are graphs, include those graphs in the correct sections. Do not simply reference your appendix.

2. If there is code, submit all code needed to reproduce your results, "HW[X] Code".

3. If there is a test set, submit your test set evaluation results, "HW[X] Test Set".

4. In all cases, replace "[X]" with the number of the assignment you are submitting.

1 Linear Regression, Projections and Pseudoinverses

We are given $\mathbf{X} \in \mathbb{R}^{n \times d}$ where $n > d$ and $\text{rank}(\mathbf{X}) = d$. We are also given a vector $\mathbf{y} \in \mathbb{R}^n$. Define the orthogonal projection of $\mathbf{y}$ onto range($\mathbf{X}$) as $P_{\mathbf{X}}(\mathbf{y})$.

(a) Prove that $P_{\mathbf{X}}(\mathbf{y}) = \underset{\mathbf{u} \in \text{range}(\mathbf{X})}{\arg\min} \|\mathbf{y} - \mathbf{u}\|^2$.

Note that in lecture, we learned how to find $\widehat{\mathbf{w}}$ that minimizes the least squares loss $L(\mathbf{w}) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$. In other words, we tried to find $\mathbf{w}$ such that $\mathbf{X}\mathbf{w}$ is the vector in the column space of $\mathbf{X}$ that is closest to our response vector $\mathbf{y}$. Hence, $P_{\mathbf{X}}(\mathbf{y}) = \mathbf{X}\mathbf{w}$.

(b) An orthogonal projection is a linear transformation. Hence, we can define $P_{\mathbf{X}}(\mathbf{y}) = \mathbf{P}\mathbf{y}$ for some projection matrix $\mathbf{P}$. Specifically, given $1 \leq d \leq n$, a matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$ is said to be a rank-d orthogonal projection matrix if $\text{rank}(\mathbf{P}) = d$, $\mathbf{P} = \mathbf{P}^\top$ and $\mathbf{P}^2 = \mathbf{P}$. Prove that $\mathbf{P}$ is a rank-d projection matrix if and only if there exists a $\mathbf{U} \in \mathbb{R}^{n \times d}$ such that $\mathbf{P} = \mathbf{U}\mathbf{U}^\top$ and $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$.

(c) Prove that if $\mathbf{P}$ is a rank d projection matrix, then $\text{tr}(\mathbf{P}) = d$.

(d) Prove that if $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\text{rank}(\mathbf{X}) = d$, then $\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$ is a rank-d orthogonal projection matrix. What is the corresponding projection matrix $\mathbf{U}$ such that $\mathbf{U}\mathbf{U}^T = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$.

For the remainder of the problem set, we no longer assume that $\mathbf{X}$ is full rank.

(e) The Singular Value Decomposition theorem states that we can write any matrix $\mathbf{X}$ as

$$\mathbf{X} = \sum_{i=1}^{\min\{n,d\}} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top = \sum_{i:\sigma_i>0} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

where $\sigma_i \geq 0$, and $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ are an orthonormal. Show that

(a) $\{\mathbf{v}_i : \sigma_i > 0\}$ are an orthonormal basis for the row space of $\mathbf{X}$

(b) $\{\mathbf{u}_i : \sigma_i > 0\}$ are an orthonormal basis for the column space of $\mathbf{X}$
Hint: consider $\mathbf{X}^\top$.

(f) Define the Moore-Penrose pseudoinverse to be the matrix:

$$\mathbf{X}^\dagger = \sum_{i:\sigma_i>0} \sigma_i^{-1} \mathbf{v}_i \mathbf{u}_i^\top,$$

How is the operator $\tilde{P}_{\mathbf{X}}(\mathbf{y}) = \mathbf{X}^\dagger \mathbf{X} \mathbf{y}$ related to $P_{\mathbf{X}}$? What is $\mathbf{X}^\dagger \mathbf{X}$ if $\operatorname{rank}(\mathbf{X}) = d$? If $\operatorname{rank}(\mathbf{X}) = d$ and $n = d$?

2 The Least Norm Solution

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$, where $n \geq d$, where $\operatorname{rank}(\mathbf{X})$ is possibly less than d. As in problem 1, we will write the SVD of $\mathbf{X}$ as a sum of rank-one terms

$$\mathbf{X} = \sum_{i:\sigma_i>0} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top,$$

In this problem, our goal will to provide an explicit expression for the *least-norm* least-squares estimator, defined as:

$$\widehat{\mathbf{w}}_{LN} := \arg\min_{\mathbf{w}} \{\|\mathbf{w}\|^2 : \mathbf{w} \text{ is a minimizer of } \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2\},$$

where $\mathbf{w} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^n$.

(a) Show without using the SVD of $\mathbf{X}$ that $\widehat{\mathbf{w}}_{LN}$ is the unique minimizer of $\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$ which lies in the rowspace of $\mathbf{X}$. Try not to use the SVD.

(b) Define $\tilde{\mathbf{w}}$ as the following:

$$\tilde{\mathbf{w}} = \sum_{i:\sigma_i>0} \frac{1}{\sigma_i} \mathbf{v}_i (\mathbf{u}_i^\top \mathbf{y}), \tag{1}$$

Solve this problem by directly checking that the above expression for $\tilde{\mathbf{w}}$ is in the rowspace of $\mathbf{X}$, and satisfies the necessary optimality condition to be a minimizer of the least-squares objective. Show that

$$\tilde{\mathbf{w}} = \widehat{\mathbf{w}}_{LN}.$$

(c) We give another solution to finding a form for $\widehat{\mathbf{w}}_{LN}$ using the pseudoinverse. Follow these steps:

(1) Show that for any $\mathbf{u} \in \mathbb{R}^n$, $(\mathbf{X}^\top\mathbf{X})^\dagger(\mathbf{X}^\top\mathbf{X}) = P_{\mathbf{X}}(\mathbf{u})$
Hint: pattern match with the last part of Problem 1, where $\mathbf{X} \leftarrow \mathbf{X}^\top\mathbf{X}$.

(2) Show that $(\mathbf{X}^\top\mathbf{X})^\dagger\mathbf{X}^\top = \mathbf{X}^\dagger$.
Hint: write everything out as a sum of rank-one terms.

(3) Show by using the least-squares optimality conditions that any minimizer of the least-squares objective satisfies $P_{\mathbf{X}}(\mathbf{w}) = \mathbf{X}^\dagger\mathbf{y}$ where $P_{\mathbf{X}}$ is the orthogonal projection onto the row space of $\mathbf{X}$.

(4) Conclude that

$$\widehat{\mathbf{w}}_{LN} = \mathbf{X}^\dagger\mathbf{y}.$$

Verify that this is consistent with your answer to the previous part of the problem.

3 The Ridge Regression Estimator

Recall the ridge estimator for $\lambda > 0$,

$$\widehat{\mathbf{w}}_\lambda := \arg\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda\|\mathbf{w}\|^2,$$

Let

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

be the SVD decomposition of $\mathbf{X}$.

(a) Show that $\mathbf{w}_\lambda$ is unique and

$$\widehat{\mathbf{w}}_\lambda = \sum_{i=1}^{d} \frac{\sigma_i}{\sigma_i^2 + \lambda} \mathbf{v}_i \langle \mathbf{u}_i, \mathbf{y} \rangle.$$

(b) Show that

$$\|\widehat{\mathbf{w}}_\lambda\|^2 = \sum_{i:\sigma_i>0} \left(\frac{\sigma_i}{\sigma_i^2 + \lambda}\right)^2 \langle \mathbf{u}_i, \mathbf{y} \rangle^2.$$

(c) Recall the least-norm least squares solution is $\widehat{\mathbf{w}}_{LN}$ from Problem 2. Show that if $\widehat{\mathbf{w}}_{LN} = 0$, then $\widehat{\mathbf{w}}_\lambda = 0$ for all $\lambda > 0$.
Hint: Recall that $\widehat{\mathbf{w}}_{LN} = \sum_{i:\sigma_i>0} \sigma_i^{-1} \langle \mathbf{u}_i, \mathbf{y} \rangle \mathbf{v}_i$.

(d) Show that if $\widehat{\mathbf{w}}_{LN} \neq 0$, then the map $\lambda \mapsto \|\widehat{\mathbf{w}}_\lambda\|^2$ is strictly decreasing and strictly positive on $(0, \infty)$.

(e) Show that

$$\lim_{\lambda \to 0} \widehat{\mathbf{w}}_\lambda \to \widehat{\mathbf{w}}_{LN}.$$

(f) In light of the above, why do you think that people describe the ridge regression as "controlling the complexity" of the solution $\widehat{\mathbf{w}}_\lambda$?

4 Patrick vs. Alvin

Make sure to submit the code you write in this problem to "HW2 Code" on Gradescope.

Patrick and Alvin are having an argument. Alvin bets Patrick can't guess his secret polynomial. Patrick manages to overhear the input, but the output is noisy due to numerous Piazza posts. Patrick overhears n inputs, $x_1, x_2, \ldots x_n$, and their corresponding noisy outputs, $y_1, y_2, \ldots, y_n$. The underlying modeling function is of the form

$$y_i \approx w_0 + w_1 x_i + w_2 x_i^2 \cdots + w_d x_i^d.$$

where their goal is to fit a polynomial of degree d to this data. Include all text responses and plots in your write-up.

(a) Show how Patrick's problem can be formulated as a linear regression problem.

(b) You are given data of the inputs $\{x_i\}_{i=1}^n$ and outputs $\{y_i\}_{i=1}^n$ in the "x_train" and "y_train" keys of the file `1D_poly.mat` with the inputs centered and normalized to lie in the range $[-1, 1]$. Write a script by completing `b.py` to do a least-squares fit (taking care to include a constant term) of a polynomial function of degree d to the data. Letting f_d denote the fitted polynomial, plot the average training error $R(d) = \frac{1}{n}\sum_{i=1}^n (y_i - f_d(x_i))^2$ against d in the range $d \in \{0, 1, 2, 3, \ldots, n-1\}$. You may not use any library other than `numpy` and `numpy.linalg` for computation.

(c) How does the average training error behave as a function of d, and why? What happens if you try to fit a polynomial of degree n with a standard matrix inversion method?

(d) Patrick has taken CS189 so decides that he needs to run another experiment before deciding that their prediction is true. He decides to camp outside of Alvin's apartment and manages to overhear Alvin again with the same inputs $\{x_i\}_{i=1}^n$, with the outputs in `1D_poly.mat`. Denoting the fresh observation for x_i by $\tilde{y}_i$, by completing `d.py`, plot the average error $\tilde{R}(d) = \frac{1}{n}\sum_{i=1}^n (\tilde{y}_i - f_d(x_i))^2$ for the same values of d as in part (b) using the polynomial approximations f_d also from the previous part. How does this plot differ from the plot in (b) and why?

(e) How do you propose using the two plots from parts (b) and (d) to "select" the right polynomial model for Patrick?

(f) Patrick decides he wants to extend his analysis to use more than one feature - so he now thinks the input to his function should include the time of day, the weather, Alvin's location, the number of seconds since the last Piazza post, and a random number of his choosing. His function will therefore be a multivariate polynomial. A degree d multivariate polynomial function looks like

$$f_d(\mathbf{x}) = \sum_j \alpha_j \prod_i x_i^{p_{ji}},$$

where $\forall j : \sum_i p_{ji} \le d$. Here α_j is the scale constant for jth term and p_{ji} is the exponent of x_i in jth term. The data in `polynomial_regression_samples.mat` (100000×5) with columns corresponding to the 5 attributes of the function. Use 4-fold cross-validation to decide which of $d \in \{0, 1, 2, 3, 4, 5, 6\}$ is the best fit for the data provided. For this part, compute the polynomial coefficients via ridge regression with penalty $\lambda = 0.1$, instead of ordinary least squares. You are not allowed to use any library other than `numpy`. Write your implementation by completing `fg.py`.

(g) Now redo the previous part, but use 4-fold cross-validation on all combinations of $d \in \{1, 2, 3, 4, 5, 6\}$ and $\lambda \in \{0.05, 0.1, 0.15, 0.2\}$ - this is referred to as a grid search. Find the best d and λ that best explains the data using ridge regression. Print the average training/-validation error per sample for all d and λ. Again, write your implementation by completing `fg.py`.