CS689: Machine Learning - Spring 2023

Homework 1

Assigned: Wed Feb 15, 2023

Getting Started: This assignment consists of two parts. Part 1 consists of written problems, derivations and coding warm-up problems, while Part 2 consists of implementation and experimentation problems. You should first complete the derivation and written problems in Part 1. You should then start on the implementation and experimentation problems in Part 2 of the assignment. The implementation and experimentation problems must be coded in Python 3.8+. Download the assignment archive from Moodle and unzip the file. The data files for this assignment are in the `data` directory. Code templates are in the `code` directory. The only modules you are allowed to use in your implementation are those already imported in the code templates.

Submission and Due Dates: Part 1 is due one week after the assignment is posted. You must submit a PDF document with your solutions to Gradescope. You are strongly encouraged to typeset your solutions using LaTeX. The source of this assignment is provided to help you get started. You may also submit a PDF containing scans of clear hand-written solutions. Part 2 is due two weeks after the assignment is posted. You must submit a PDF document containing the results of your experiments to Gradescope. You must separately submit your code to Gradescope. Solutions to Part 1 will be released immediately after Part 1 is due. Late work will only be accepted in accordance with the course's late homework policy.

Academic Honesty Reminder: Homework assignments are individual work. See syllabus for details.

Part 1: Written Problems, Derivations, and Coding Warm-Up (Due Wed Feb 22, 11:59pm)

1. (*5 points*) Consider the linear regression prediction function $f_\theta(\mathbf{x}) = \mathbf{x}\mathbf{w} + b$ with $\mathbf{x} \in \mathbb{R}^D$ and $y \in \mathbb{R}$. What is the computational complexity of computing the risk when the prediction loss function is the squared loss and the data set has N data cases? Explain your answer.

2. (*5 points*) Suppose we have a regression task where $x \in \mathbb{R}$ and $y \in \mathbb{R}$ and we expect that the optimal prediction function is a polynomial of order 3 or lower. Provide a definition for a prediction function model $\mathcal{F}$ matching these assumptions. Explain your answer.

3. (*10 points*) Suppose we have a regression task with $\mathbf{x} \in \mathbb{R}^D$. Suppose the true data generating distribution satisfies $\mathbb{E}_{p_*(\mathbf{X}=\mathbf{x})}[\mathbf{x}] = \mu_*$ for some $\mu_* \in \mathbb{R}^D$. Prove that the statistic $\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n$ is an unbiased estimator of μ_* when the inputs $\mathbf{x}_1, ..., \mathbf{x}_N$ are sampled from p_*.

4. (*10 points*) Consider a regression task where $\mathcal{X} = \mathbb{R}^D$. Suppose that for all data cases, dimension j is a scaled copy of dimension i. In other words, for some $a \neq 0$ and for all n, we have $\mathbf{x}_{ni} = a \cdot \mathbf{x}_{nj}$. Explain why the standard OLS estimator for the linear regression model $\hat{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ can not be used to find

the optimal parameters of the linear regression model under squared prediction loss in this case.

5. (*20 points*) Consider the problem of building a regression model for periodic time series data. In this problem, $y \in \mathbb{R}$ and $x \in \mathbb{R}$. One way to model such data is with a regression function built from a sum of K cosine-based components. Each component k has an amplitude w_k, a phase ϕ_k and a period ρ_k. We also include a bias parameter b. The form of the prediction function is shown below:

$$f_\theta(x) = b + \sum_{k=1}^{K} w_k \cos\left(\frac{2\pi}{\rho_k}x - \phi_k\right)$$

.

In this prediction function, the parameters are $\theta = [\mathbf{w}, \phi, b]$ where $\mathbf{w} = [w_1, ..., w_K]$ and $\phi = [\phi_1, ..., \phi_K]$. The number of periodic components K and their periods ρ_k are fixed.

a. (*5 pts*) Consider the case where $K = 2$, $\rho = [50, 25]$ and $\theta = [0.5, 0.75, -0.25, 0.75, 1.0]$. Plot $f_\theta(x)$ from 0 to 100 using these parameters.

b. (*5 pts*) Suppose we have a data set $\mathcal{D}$ containing just two observations $(20, 0)$ and $(40, 2.5)$. Compute the empirical risk for this data set using $K = 2$, $\rho = [50, 25]$ and $\theta = [0.5, 0.75, -0.25, 0.75, 1.0]$. Use the squared prediction loss.

c. (*5 pts*) Derive the gradient of the empirical risk function for this model under the squared prediction loss and assuming $K = 2$. Clearly indicate the components that correspond to $\mathbf{w}$, ϕ and b. Show your work.

d. (*5 pts*)Suppose we have a data set $\mathcal{D}$ containing just two observations $(20, 0)$ and $(40, 2.5)$. Compute the gradient of the empirical risk for this data set using $K = 2$, $\rho = [50, 25]$ and $\theta = [0.5, 0.75, -0.25, 0.75, 1.0]$. under the squared prediction loss. Clearly indicate the components that correspond to $\mathbf{w}$, ϕ and b.

Part 2: Implementation and Experimentation (Due Wed Mar 1, 11:59pm)

6. (*50 points*) In this problem, you will implement the periodic regression model introduced in Question 5 and apply it to the problem of modeling tide heights in Hawaii. In this data set, the x values correspond to time since a reference date/time measured in hours. The y values correspond to sea level height in meters. The training set contains 28 days worth of data. The test set contains the next 28 days worth of data. You will train the model on the training data set, and then use it to forecast tide heights on the test data set. As in Question 5, use the squared prediction loss.

For this problem, we will use a model with $K = 9$ components. The periods for these components are derived from properties of the Sun-Earth-Moon system that determine tide dynamics. The values are $\rho = [12.42, 12.00, 12.66, 23.93, 25.82, 6.21, 4.14, 6.00, 6.10]$.[1]

Your model implementation should be contained in the file `regression.py`. Code implementing your experiments should be contained in `experiments.py`. You are strongly encouraged to use the Matplotlib

[1] `https://www.vims.edu/research/units/labgroups/tc_tutorial/tide_analysis.php`

Python library to create graphs.

To begin, implement a Python class for the model starting from the code in `regression.py`. Your class must implement the methods indicated below. You can include any additional methods that you need. You can change the provided function signatures, so long as the required methods are consistent with the descriptions below. Please include the functions listed below first, followed by any functions you add.

- `f`: The prediction function. Takes a parameter vector θ and an array of inputs $\mathbf{X}$ of shape $N \times 1$ as input. Returns an array of predicted outputs $\hat{\mathbf{Y}}$ of shape $N \times 1$ where $\hat{\mathbf{Y}}_i = f_\theta(x_i)$. The value of ρ should be accessed from a class member variable.
- `risk`: The risk function. Takes a parameter vector θ, an array of inputs $\mathbf{X}$ of shape $N \times 1$ and an array of outputs $\mathbf{Y}$ of shape $N \times 1$ as input. Returns the risk of f_θ computed using these data. The value of ρ should be accessed from a class member variable.
- `riskGrad`: The gradient of the risk function. Takes a parameter vector θ, an array of inputs $\mathbf{X}$ of shape $N \times 1$ and an array of outputs $\mathbf{Y}$ of shape $N \times 1$ as input. Returns the gradient of the risk of f_θ computed using these data. The value of ρ should be accessed from a class member variable.
- `fit`: Takes an array of inputs $\mathbf{X}$ of shape $N \times 1$ and an array of outputs $\mathbf{Y}$ of shape $N \times 1$ as input and computes and stores the estimated model parameters $\hat{\theta}$ as a class member variable. You will need to use a numerical optimizer to implement this function. The optimizer we will use is `scipy.optimize.minimize` with the L-BFGS-B method and options={"disp":1}, tol=1e-6. The value of ρ should be accessed from a class member variable.

a. (*10 pts*) Using the provided training data in the tide height data set `data.npz`, compute and report the value of the risk when using $\theta = [1, 1..., 1]$ as the parameter vector.

b. (*10 pts*)Using the provided training data in the tide height data set `data.npz`, compute and report the gradient of the risk when using $\theta = [1, 1..., 1]$ as the parameter vector. (Hint: See `scipy.optimize.approx_fprime` for help debugging your gradient implementation).

c. (*10 pts*) Using the provided tide height data set `data.npz`, fit the model on all of the training data. Include the output produced by the optimizer in your report. As the starting value of θ_0 for the optimization, use the zero vector $\theta_0 = [0, 0..., 0]$.

d. (*10 pts*)Compute and report the average squared loss of the learned model on the training set and also on the test set.

e. (*5 pts*) Produce a single plot showing the first 48 hours of the training data along with predicted tide heights given by the fit model. Show the training data as points and the tide height predictions as a line graph.

f. (*5 pts*) Produce a single graph showing the last 48 hours of the test data along with predicted tide heights given by the fit model. Show the test data as points and the tide height predictions as a line graph.