CS 446 / ECE 449 — Homework 4

yiminc2

Instructions.

- Homework is due **Friday, Oct 31**, at 11:59 **PM** CST; you have **3** late days in total for **all Homeworks**.
- Everyone must submit individually at Gradescope under `HW4` and `HW4 - Programming Assignment`.
- The "written" submission at `HW4` **must be typed**, and submitted in any format gradescope accepts (to be safe, submit a PDF). You may use LaTeX, markdown, google docs, MS word, whatever you like; but it must be typed!
- When submitting at `HW4`, Gradescope will ask you to **mark out boxes around each of your answers**; please do this precisely!
- Please make sure your NetID is clear and large on the first page of the homework.
- Your solution **must** be written in your own words. Please see the course webpage for full **academic integrity** information. You should cite any external reference you use.
- We reserve the right to reduce the auto-graded score for `HW4 - Programming Assignment` if we detect funny business (e.g., your solution lacks any algorithm and hard-codes answers you obtained from someone else, or simply via trial-and-error with the autograder).
- When submitting to `HW4 - Programming Assignment`, only upload `hw4_q3.py` and `hw4_utils.py`. Additional files will be ignored.

1. Bias-Variance in Ridge Regression. (23 pt)

Recall from the lecture, the Expected Test Error can be decomposed as follows:

$$\mathbb{E}_{x,y,\mathcal{D}}[(h_{\mathcal{D}}(x) - y)^2] = \underbrace{\mathbb{E}_{x,\mathcal{D}}[(h_{\mathcal{D}}(x) - \bar{h}(x))^2]}_{\text{Variance}} + \underbrace{\mathbb{E}_{x}[(\bar{h}(x) - \bar{y}(x))^2]}_{\text{Bias}^2} + \underbrace{\mathbb{E}_{x,y}[(\bar{y}(x) - y)^2]}_{\text{Noise}}$$

Consider fixed (non-random) scalar features $\{x^{(i)}\}_{i=1}^N$. The labels are generated as $y^{(i)} = w^* x^{(i)} + \epsilon^{(i)}$ where w^* is fixed and $\epsilon^{(i)}$ are i.i.d noises from Gaussian distribution $N(0, \sigma^2)$. Note that w^* is unknown and $\epsilon^{(i)}$ is independent of $x^{(i)}$. Therefore, we can define the observed dataset as $\mathcal{D} = \{x^{(i)}, y^{(i)}\}_{i=1}^N$.

Ridge regression optimizes the following objective for a dataset $\mathcal{D}$ with $\lambda \geq 0$:

$$w_{\mathcal{D}} = \arg\min_{w} \frac{1}{N} \sum_{i=1}^{N} (wx^{(i)} - y^{(i)})^2 + \lambda w^2$$

For simplicity, the intercept term is omitted from this problem. The closed-form solution of ridge regression is given as:

$$w_{\mathcal{D}} = \frac{\frac{1}{N}\sum_{i=1}^{N} x^{(i)} y^{(i)}}{\lambda + \frac{1}{N}\sum_{i=1}^{N} x^{(i)2}}$$

(a) Consider the expected label $\bar{y}(x) = \mathbb{E}_{y|x}[y]$. Show that $\bar{y}(x) = w^* x$. Similarly, consider the noise term:

$$\text{Noise} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{y^{(i)}|x^{(i)}}[(\bar{y}(x^{(i)}) - y^{(i)})^2]$$

Show that Noise $= \sigma^2$. (3 pt)

(b) From the lecture, given a machine learning algorithm $\mathcal{A}$, then $h_{\mathcal{D}} = \mathcal{A}(\mathcal{D})$. For our case, $h_{\mathcal{D}}(x) = w_{\mathcal{D}} x$. Consider the expected predictor $\bar{h} = \mathbb{E}_{\mathcal{D} \sim P^N}[h_{\mathcal{D}}]$, then in our case $\bar{w} = \mathbb{E}_{\mathcal{D}}[w_{\mathcal{D}}]$. Let $s^2 = \frac{1}{N}\sum_{i=1}^{N} x^{(i)2}$, show that:

$$\bar{w} = \frac{s^2}{\lambda + s^2} w^*$$

(3 pt)

(c) Consider the squared bias term:

$$\text{Bias}^2 = \frac{1}{N} \sum_{i=1}^{N} (\bar{w} x^{(i)} - \bar{y}(x^{(i)}))^2$$

Show that:

$$\text{Bias}^2 = \left(\frac{\lambda}{\lambda + s^2}\right)^2 w^{*2} s^2$$

(3 pt)

(d) Consider the variance term:

$$\text{Variance} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{\mathcal{D}}\left[(w_{\mathcal{D}} x^{(i)} - \mathbb{E}_{\mathcal{D}}[w_{\mathcal{D}} x^{(i)}])^2\right]$$

Show that:

$$\text{Variance} = \frac{s^4 \sigma^2}{N(\lambda + s^2)^2}$$

(5 pt)

(e) What happens to the Bias^2 and Variance term when $\lambda \to 0$ and $\lambda \to \infty$. Your answer should demonstrate that the bias and variance are monotonic with respect to λ, but in different directions. Therefore, changing λ controls the trade-offs. In practice, since we don't know w^* and the true distribution of ϵ, we cannot infer the optimal value of λ. Therefore, we use model selection to determine the best value for λ. (3 pt)

(f) Alternatively, we can consider an equivalent form of ridge regression:

$$w_{\mathcal{D}} = \arg\min_{w} \frac{1}{N} \sum_{i=1}^{N} (wx^{(i)} - y^{(i)})^2 \quad \text{so that} \quad w^2 \leq R$$

The regularization constraint forces the weight w to be inside a ball around the origin with radius $\sqrt{R}$. Use the triangle inequality to show that:

$$|w_{\mathcal{D}} - \bar{w}|^2 \leq 4R$$

From there, we can see that the maximum Euclidean distance between any two points in the ball can at most be $2\sqrt{R}$. (3 pt)

(g) Show that ridge regression bounds the variance by $4Rs^2$

$$\text{Variance} \leq 4Rs^2$$

Note that this bound does not depend on w^* or ϵ, but it can be loose compared to the actual value of variance. (3 pt)

Solution.

2. Optimal Classifier under Squared Loss. (12 pt)

Let $h_D(\boldsymbol{x})$ be a predictor trained on a dataset D, which maps an input feature vector $\boldsymbol{x} \in \mathbb{R}^d$ to a predicted output. The output variable is denoted by $y \in \mathbb{R}$.

Consider the expected squared error loss, which measures the performance of our predictor. This expectation is taken over the joint distribution P of input data $\boldsymbol{x}$ and the true labels y, and distribution of dataset D samples from P^N, where D has N data points:

$$L = E_{(\boldsymbol{x},y)\sim P, D\sim P^N}\left[(h_D(\boldsymbol{x}) - y)^2\right]$$

Your task is to:

(a) **Find the Optimal Classifier**: Derive the predictor $h_{opt}(\boldsymbol{x})$ that minimizes this expected loss. Note that the optimal predictor should not be dependent on any specific dataset D. (6 pt)
Hint: One route you can take is applying the law of total expectation and minimizing the inner expectation for a fixed classifier $h_D(\boldsymbol{x})$.

(b) **Find the Optimal Error Rate**: Derive the minium achievable error, or irreducible error, after you derive the optimal classifier. (6 pt)

Solution.

3. Model Selection. (19 pt)

In this problem, you will implement a model selection pipeline using k-fold cross-validation to find the best hyper-parameters for polynomial regression with regularization. You can see more detailed instructions in the code file `hw4_q3.py`.

Submission Instruction If you want to implement any helper function of your own, please make sure you either put it directly in `hw4_q3.py` or put them into `hw4_utils.py` and submit `hw4_utils.py` with `hw4_q3.py` to Gradescope!

(a) **K-Fold Cross-Validation (8 pt)**
Implement `cross_validate_model(X, y, model, k_folds)` that

- Splits the data into k folds using `KFold` with `shuffle=True` and `random_state=42`
- For each fold, trains the model on $k-1$ folds and evaluates on the remaining fold
- Returns the mean and standard deviation of validation mean squared error across all folds

Remark 1: For `model`, you can train the `model` by calling `model.fit(X,y)` on data `(X,y)`. In addition, you can call `model.predict(X)` to obtain the prediction from `model`.
Remark 2: For each iteration during k-fold cross validation, please make sure you make a copy of `model` by `model_copy = deepcopy(model)` and then train `model_copy` instead of `model`. Otherwise, you will be training a model from previous iteration.

(b) **Model Selection (11 pt)**
Implement `select_best_model(X_train, y_train)` that sweeps through different polynomial degrees and regularization strengths (for Ridge and Lasso regression) to perform k-fold cross validation with $k = 5$. The function should return the model with lowest cross-validation error.
Remark 1: You can use `LinearRegression()` to initialize the Linear Regression model.
Remark 2: You can use `Ridge(alpha=alpha, random_state=42)` to initialize the Ridge Regression model.
Remark 3: You can use `Lasso(alpha=alpha, random_state=42, max_iter=2000)` to initialize the Lasso Regression model.

Solution.