Introduction to Machine Learning (67577)

Exercise 5
PCA, Kernels, Regularization, and Cross-Validation

Second Semester, 2022

Contents

Submission

Please make sure to follow the general submission instructions available on the course website. In addition, for the following assignment, submit a single `ex5_ID.tar` file containing:

- An `Answers.pdf` file with the answers for all theoretical and practical questions (include plotted graphs *in* the PDF file).
- The following python files (without any directories): `cross_validate.py`, `ridge_regression.py`, `perform_model_selection.py`

The `ex5_ID.tar` file must be submitted in the designated Moodle activity prior to the date specified *in the activity*.

- Late submissions will not be accepted and result in a zero mark.
- Plots included as separate files will be considered as not provided.
- Do not forget to answer the Moodle quiz of this assignment.

1 Theoretical Part

1.1 Regularization

Based on Lecture 7 and Recitation 9

1. In the following question we will show that although the Ridge estimator is biased it can achieve lower MSE compared to the LS estimator.

   Let $\mathbf{X} \in \mathbb{R}^{m \times d}$ be a **constant** design matrix, $\mathbf{y} \in \mathbb{R}^d$ a response vector, and assume that $\mathbf{X}^\top \mathbf{X}$ is invertible. Denote $\hat{\mathbf{w}}$ the LS solution and $\hat{\mathbf{w}}_\lambda$ the ridge solution for the regularization parameter $\lambda \geq 0$ (where $\hat{\mathbf{w}}_0 \equiv \hat{\mathbf{w}}$)

   - Assume the linear model is correct, namely $\mathbf{y} = X\mathbf{w} + \varepsilon$ where $\varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$.
   - Recall that in this case: $\mathbb{E}[\hat{\mathbf{w}}] = \mathbf{w}$.

   (a) Show that $\hat{\mathbf{w}}_\lambda = A_\lambda \hat{\mathbf{w}}$ where $A_\lambda := (\mathbf{X}^\top \mathbf{X} + \lambda I_d)^{-1}(\mathbf{X}^\top \mathbf{X})$

   (b) From the above, conclude that for any $\lambda > 0$ the ridge estimator is a biased estimator of $\mathbf{w}$. That is, show that for any $\lambda > 0 \quad \mathbb{E}[\hat{\mathbf{w}}_\lambda] \neq \mathbf{w}$.

   (c) Show that: $\mathrm{Var}(\hat{\mathbf{w}}_\lambda) = \sigma^2 A_\lambda (\mathbf{X}^\top \mathbf{X})^{-1} A_\lambda^\top$, for σ^2 the variance of the assumed noise.
   Hint:: Recall that for a constant matrix B and a random vector $\mathbf{z}$ it holds that$\mathrm{Var}(B\mathbf{z}) = B \cdot \mathrm{Var}(\mathbf{z}) \cdot B^\top$ and that $\mathrm{Var}(\hat{\mathbf{w}}) = \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}$.

   (d) Derive explicit expressions for the (squared) bias and variance of $\hat{\mathbf{w}}_\lambda$ as a function of λ, i.e. write a bias-variance decomposition for the mean square error of $\hat{\mathbf{w}}_\lambda$.

   Hint: recall that for the multivariate case the MSE defined to be:

   $$MSE(\hat{\mathbf{y}}) = \mathbb{E}\left[\|\hat{\mathbf{y}} - \mathbf{y}\|^2\right] = \mathbb{E}\left[(\hat{\mathbf{y}} - \mathbf{y})^\top (\hat{\mathbf{y}} - \mathbf{y})\right]$$

   where $\mathbf{y}$ is the true value, and $\hat{\mathbf{y}}$ is our estimation.

   (e) Show by differentiation that

   $$\frac{\partial}{\partial \lambda} \mathrm{MSE}(\hat{\mathbf{w}}_\lambda)|_{\lambda=0} = \frac{\partial}{\partial \lambda} \mathrm{bias}^2(\hat{\mathbf{w}}_\lambda)|_{\lambda=0} + \frac{\partial}{\partial \lambda} \mathrm{Var}(\hat{\mathbf{w}}_\lambda)|_{\lambda=0} < 0$$

   That is, calculate the derivative of the functions above with respect to λ at point $\lambda = 0$.

(f) Conclude that, if the linear model is correct, a little Ridge regularization helps to reduce the MSE.

1.2 PCA

Based on Lecture 8 and Recitation 11

2. Let $X : \Omega \to \mathbb{R}^d$ be a random variable with zero mean and covariance $\Sigma \in \mathbb{R}^{d\times d}$. Show that for any $\mathbf{v} \in \mathbb{R}^d$, where $\|\mathbf{v}\|_2 = 1$, the variance of $\langle \mathbf{v}, X\rangle$ is not larger then variance obtained by the PCA embedding of X into a one-dimension subspace (assume that the PCA uses the actual Σ).

1.3 Kernels

Based on Lecture 9 and Recitation 12

3. Let $k(\mathbf{x},\mathbf{x}')$ be a valid PSD kernel. Provide a valid PSD kernel $\tilde{k}(\mathbf{x},\mathbf{x}')$, constructed from k, which is guaranteed to be normalized. That is, for all $\mathbf{x}$ it holds that $\tilde{k}(\mathbf{x},\mathbf{x}) = 1$. Prove your answer.
4. Consider a data set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{\pm 1\}$, and a feature map $\psi : \mathbb{R}^d \to \mathcal{F}$ where $\mathcal{F}$ is some feature space. Give an example of a data set S and a feature map ψ such that S is not linearly separable in $\mathbb{R}^d$ (for $d \geq 2$) but that the transformed data set $S_\psi = \{(\psi(\mathbf{x}_i), y_i)\}_{i=1}^m$ is linearly separable in $\mathcal{F}$.
5. For each of the following functions, prove it is a valid PSD kernel or show a counter example:
   (a) $k(\mathbf{x},\mathbf{y}) = \exp(||\mathbf{x}-\mathbf{y}||^2)$
   (b) $k(\mathbf{x},\mathbf{y}) = k_1(\mathbf{x},\mathbf{y}) - k_2(\mathbf{x},\mathbf{y})$ for any two valid kernels $k_1(\cdot,\cdot)$ and $k_2(\cdot,\cdot)$
   (c) $k(\mathbf{x},\mathbf{y}) = k_a(\mathbf{x}_a,\mathbf{y}_a) + k_b(\mathbf{x}_b,\mathbf{y}_b)$ for any two valid kernels $k_a(\cdot,\cdot)$ and $k_b(\cdot,\cdot)$, where

$$\mathbf{x} := \left[\frac{\mathbf{x}_a}{\mathbf{x}_b}\right], \quad \mathbf{y} := \left[\frac{\mathbf{y}_a}{\mathbf{y}_b}\right]$$

2 Practical Part

Based on Lecture 7 and Recitations 9-10

Implement the `cross_validate` function in the `IMLearn\model_selection\cross_validate.py` according to function documentation.

- The method's signature includes the `scoring` argument. This is a callable which receives two `np.ndarrays` and evaluates them with respect to some metric. When calling the `cross_validate` function you will be passing the `mean_square_error` previously implemented.

2.1 Cross Validation For Selecting Polynomial Degree

In the following questions you will perform polynomial fitting using your *own* implementation of the class. Using Cross-Validation we will find the best fitting degree for a given data scenario.

Implement the following in the `select_polynomial_degree` function in the `perform_model_selection.py` file and answer the question below:

1. Generate a dataset of $m = 100$ samples according to the following model: $y = f(x) + \varepsilon$ where

$$f(x) := (x+3)(x+2)(x+1)(x-1)(x-2), \quad \varepsilon \sim \mathcal{N}(0,\sigma^2)$$

and x are selected uniformly in the range $x \in [-1.2, 2]$. Use a noise level (i.e. σ^2) of 5. Then, split the dataset into train- and test-sets with a `train_portion` of $\frac{2}{3}$ using your implementation of the `split_train_test` function.
Scatter plot the true (noiseless) model and the two sets using different colors for train and test samples.

2. Using only the training portion of the samples, perform 5-fold cross-validation for each of the polynomial degrees $k = 0, 1, \ldots, 10$. Plot for each value of k the average training- and validation errors. Explain your results and address the differences between the train and validation errors

3. Using k^*, the polynomial degree for which the lowest validation error was achieved, fit a polynomial model over the entire training set (that is, not only on specific folds). Report what was the value of k^* and the *test* error of the fitted model. Is the test error similar to the validation error previously achieved? Round the test error to two decimal places.

4. Repeat the questions above but using a noise level of 0. Show the figures obtained for such scenario. Compare your results to the noisy case and explain. Address the differences between the train-, validation- and test errors for the different values of k.

5. Repeat the questions above while generating $m = 1500$ samples using a noise level of 10. What can we learn about the use of Cross-Validation with respect to the sample size? How is this supported by the Law of Large Numbers and the consistency property of estimators?

2.2 Choosing Regularization Parameters Using Cross Validation

In the following questions you will compare the Lasso and Ridge regularizations over a dataset of glucose levels of diabetes patients.

- Implement the `RidgeRegression` class in the `learners.regressors.ridge_regression.py` as specified in class documentation.
- For the Lasso regression use sklearn's implementation
- You are *not allowed* to use sklearns Cross-Validation or `LassoCV` implementations.

Then, implement and answer the following:

6. Load the diabetes dataset and split it to a training and testing portion, using 50 samples for training.

7. For both Ridge and Lasso regularizations, run 5-Fold Cross-Validation using different values of the regularization parameter λ. Explore possible ranges of values of λ (say, take `n_evaluations=500` equally spaced values in a range of your choice). Which range of values is meaningful for the given data for each of the algorithms?

   Then, over these selected ranges, for each of the two algorithms, plot the train- and validation errors as a function of the tested regularization parameter value. Explain your results. Address

both the differences between the train- and validation errors as well as the differences in the plots for the two algorithms.

8. Report which regularization parameter values achieved the best validation errors for the Ridge and Lasso regularizations. Then, using these values, fit Ridge, Lasso and Least Squares regressions over the entire training set. Report the test errors of each of the fitted models.
   - Use your *own* previously implemented Least Squares algorithm.