AERSP 597: Homework 1

Out: 01/20/2025

*Any figures submitted have to be generated using a computer language, with the axis labels, titles, and legends added. Please submit only **one** PDF file, with your analytical work, code, and figures combined; failing to do so may result in reduction of points. If you worked with others for coding, please list the names of all your collaborators.*

I. Linear and Ridge Regression (40 pts)

In this problem, you will implement linear regression (using polynomial features) to reproduce curves similar to those you have seen in lecture, exploring how various choices of model-complexity parameters affect training and test error.

The training and test datasets are provided to you, named `flight_data_train.csv` and `flight_data_test.csv`. The datasets characterize the control input to an autopilot system. Every row in these csv files correspond to a single datapoint $\mathbf{x}$, $\mathbf{t}$. The first six parameters are the control inputs while the last parameter is the output, flight performance index. Your task is to predict the flight performance index from the control inputs. Also, you might need to append an additional feature that is constant for all data points, for example a feature with a value of 1, in order to model the intercept term. Finally, note that you might want to first normalize each column of the data before training the model.

1. Fit the data by applying the psuedo-inverse approach of linear regression using x_j^i, $i = 1, \ldots, m$ as features (try $m = 1, 2, \ldots, 6$), where x_j represents the jth component of vector $\mathbf{x}$. In other words, you will need to raise every element in vector $\mathbf{x}$ to the power of i, for $i = 1, 2, \ldots, m$ to get the set of features. For example, if $\mathbf{x} = [x_1, x_2]'$ and $m = 2$, then you'd have $M = 5$ features: $x_1^1, x_1^2, x_2^1, x_2^2$ along with an additional feature 1 for the bias term. Plot training error and test error as Root Mean Square Error (RMSE) against m, the order of the polynomial features.
2. Fit the data using psuedo-inverse approach of ridge regression. For this, you need to use all polynomial features (i.e. $m = 6$), and choose values of $\ln \lambda$ using a sweep as follows: $-30, -29, \ldots, 8, 9, 10$. Plot training error and test error as Root Mean Square Error (RMSE) against $\ln \lambda$, the regularization coefficient.
3. Continuing from the ridge regression: When the size of dataset is limited, it is impractical to divide the samples into the training and test datasets. Instead, one needs to carry out the model training several times over all the available data samples, and determine the optimal complexity parameters using a certain method of model selection. Using the training dataset only and the sweep of $\ln \lambda$: $-30, \ldots, 9, 10$, determine the optimal λ by:
   (a) 10-fold cross validation.
   (b) Akaike Information Criteria: $AIC = N \ln \frac{E_D(\mathbf{w})}{N} + \gamma$
   (c) Bayesian Information Criteria: $BIC = N \ln \frac{E_D(\mathbf{w})}{N} + \gamma \ln N$

   where $\gamma = \text{Trace}[(\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{I})^{-1}(\mathbf{\Phi}^T\mathbf{\Phi})]$ is the number of complexity parameters for ridge regression. Comment on the differences in the values of λ found by different model selection methods.
4. **[Optional/Challenge]** Using the training dataset only, apply the iterative procedure based on evidence approximation to determine the optimal α and β. Note that the ratio $\frac{\alpha}{\beta}$ is equivalent to the regularization λ. How is the λ found by the Bayesian approach different from those values found in the previous question?

II. Constructing Kernels (20 points)

1. Let $\mathbf{u}, \mathbf{w}$ be vectors of dimension d. What feature map $\boldsymbol{\phi}$ does the kernel

$$k(\mathbf{u}, \mathbf{w}) = (\langle \mathbf{u}, \mathbf{w} \rangle + 1)^4 \tag{1}$$

   correspond to? In other words, specify the function $\boldsymbol{\phi}(\cdot)$ so that $k(\mathbf{u}, \mathbf{w}) = \boldsymbol{\phi}(\mathbf{u})^\top \boldsymbol{\phi}(\mathbf{w})$ for all $\mathbf{u}, \mathbf{w}$. Please show the expression for $d = 3$ and describe how to extend it to arbitrary dimension d. (Hint: Use indices to simplify your expression)
2. Let k_1, k_2 be positive-definite kernel functions over $\mathbb{R}^D \times \mathbb{R}^D$, let $a \in \mathbb{R}^+$ be a positive real number, let $f : \mathbb{R}^D \to \mathbb{R}$ be a real-valued function and let $p : \mathbb{R} \to \mathbb{R}$ be a polynomial with *positive* coefficients. For each of the functions k below, state whether it is necessarily a positive-definite kernel. If you think it is, prove it; if you think it is not, give a counterexample.

(a) $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) + k_2(\mathbf{x}, \mathbf{z})$
(b) $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) - k_2(\mathbf{x}, \mathbf{z})$
(c) $k(\mathbf{x}, \mathbf{z}) = ak_1(\mathbf{x}, \mathbf{z})$
(d) $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z})k_2(\mathbf{x}, \mathbf{z})$
(e) $k(\mathbf{x}, \mathbf{z}) = f(\mathbf{x})f(\mathbf{z})$
(f) $k(\mathbf{x}, \mathbf{z}) = p(k_1(\mathbf{x}, \mathbf{z}))$

3. Prove that the Gaussian Kernel

$$k(\mathbf{x}, \mathbf{z}) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right) \tag{2}$$

can be expressed as $\boldsymbol{\phi}(\mathbf{x})^T\boldsymbol{\phi}(\mathbf{z})$, where $\boldsymbol{\phi}(\cdot)$ is an infinite-dimensional vector. (Hint: Use Taylor series)

III. Kernelized Ridge Regression (20 points)

Recall that the error function for ridge regression (linear regression with L2 regularization) is:

$$E(\mathbf{w}) = (\boldsymbol{\Phi}\mathbf{w} - \mathbf{t})^T(\boldsymbol{\Phi}\mathbf{w} - \mathbf{t}) + \lambda\mathbf{w}^T\mathbf{w} \tag{3}$$

and its closed-form solution and model are:

$$\hat{\mathbf{w}} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\mathbf{I})^{-1}\boldsymbol{\Phi}^T\mathbf{t} \tag{4}$$

$$\hat{f}(\mathbf{x}) = \hat{\mathbf{w}}^T\boldsymbol{\phi}(\mathbf{x}) = \mathbf{t}^T\boldsymbol{\Phi}(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\mathbf{I})^{-1}\boldsymbol{\phi}(\mathbf{x}) \tag{5}$$

Now we want to kernelize ridge regression and allow non-linear models.

1. Use the following matrix inverse lemma to derive the closed-form solution and model for kernelized ridge regression:

$$(\mathbf{P} + \mathbf{QRS})^{-1} = \mathbf{P}^{-1} - \mathbf{P}^{-1}\mathbf{Q}(\mathbf{R}^{-1} + \mathbf{SP}^{-1}\mathbf{Q})^{-1}\mathbf{SP}^{-1} \tag{6}$$

where $\mathbf{P} \in \mathbb{R}^{n\times n}$, $\mathbf{R} \in \mathbb{R}^{k\times k}$, $\mathbf{Q} \in \mathbb{R}^{n\times k}$, and $\mathbf{S} \in \mathbb{R}^{k\times n}$. Make sure that your kernelized model only depends on the feature vectors $\boldsymbol{\phi}(\mathbf{x})$ through inner products with other feature vectors.

2. Apply kernelized ridge regression to the steel ultimate tensile strength dataset, provided in `steel_composition_train.csv`. The last column is the output, and the rest are the inputs. As usual, you are recommended to normalize the data before applying the models. Report the RMSE (Root Mean Square Error) of the models on the training data. Try (set $\lambda = 1$)
   (a) Polynomial kernel $k(\mathbf{u}, \mathbf{w}) = (\langle\mathbf{u}, \mathbf{w}\rangle + 1)^2$
   (b) Polynomial kernel $k(\mathbf{u}, \mathbf{w}) = (\langle\mathbf{u}, \mathbf{w}\rangle + 1)^3$
   (c) Polynomial kernel $k(\mathbf{u}, \mathbf{w}) = (\langle\mathbf{u}, \mathbf{w}\rangle + 1)^4$
   (d) Gaussian kernel $k(\mathbf{u}, \mathbf{w}) = \exp\left(-\frac{\|\mathbf{u}-\mathbf{w}\|^2}{2\sigma^2}\right)$ (set $\sigma = 1$)

IV. Learning Dynamics (20+10 points)

The students with the top 3 most accurate model gets the 10 bonus points; if the model were developed by a team of N students, each student would get 10/N bonus points.

In this question, you are given a dataset of trajectories from a dynamical system, and asked to identify the dynamics model from the data.

This is an open-ended question, in the sense that you could try as many as possible methods from the linear/kernel regression modules to produce a model that is as accurate as possible.

You may use existing packages to experiment different methods/models quickly, but in the submitted work, please implement your own model.

Dataset The dataset consists of `train_dyn_hw1.npy` and `test_dyn_hw1.npy`. The dynamical system is 3-dimension. The sample trajectories are recorded with a step size of 0.04 s and 51 steps. The training and test datasets contain 40 and 10 trajectories, respectively.

Assessment First, plot and compare the predicted and true trajectories in the test dataset.

Second, the accuracy of a model should be quantified by two errors, one-step error and roll-out error, using the test dataset. For one trajectory of N_t steps, $\{(t_i, \mathbf{x}_i)\}_{i=1}^{N_t}$

1. One-step error is to compare the true i step (denoted $\mathbf{x}_i$), and the prediction of the ith step given the $(i-1)$th step (denoted $\mathbf{F}_i$).

$$\epsilon_o = \left(\frac{1}{N_t} \sum_{i=1}^{N_t} \|\mathbf{x}_i - \mathbf{F}_i\|^2 \right)^{1/2} \tag{7}$$

2. Roll-out error is to compare the entire true trajectory (denoted $\{\mathbf{x}_i\}_{i=1}^{N_t}$), and the prediction of the entire trajectory given only the initial condition $\mathbf{x}_0$ (denoted $\{\hat{\mathbf{x}}_i\}_{i=1}^{N_t}$).

$$\epsilon_r = \left(\frac{1}{N_t} \sum_{i=1}^{N_t} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 \right)^{1/2} \tag{8}$$

Report the mean and maximum of ϵ_o and ϵ_r of your model.