EE798R: Intelligent Pattern Recognition

Prof. Tushar Sandhan
sandhan@iitk.ac.in

Assignment - 1

Due date: 15 Sept, 2024 | Weight: 10%
Due time: 11:59PM | Submission: MookIT

Introduction

This assignment is based on applications of Bayesian Linear Regression and Probabilistic Classification.

1 Bayesian Linear Regression Using RBF Kernel: `BLR_RBF` [5%]

1.1 Introduction

In this problem, you are required to implement a Bayesian Linear Regression model using Radial Basis Function (RBF) kernels. This model will allow you to perform regression in a transformed feature space defined by RBF kernels, and it will enable you to update the posterior distribution of model parameters based on observed data. The task involves making predictions with associated uncertainty estimates using this posterior distribution.

Important Links :

- Colab Notebook : IPR_Assignment_01_Q1
- Dataset : features_q1.npy, labels_q1.npy, noisy_labels_q1.npy

1.2 Dataset

The dataset for this assignment is provided in separate files:

- `features_q1.npy` - Contains the input features.
- `labels_q1.npy` - Contains the true values.
- `noisy_labels_q1.npy` - Contains the labels corrupted by Gaussian noise, which will be used for training.

You need to load these files to perform model training and evaluation. The data consists of feature values uniformly sampled from a given range, true values computed from a sine function, and noisy observations generated by adding Gaussian noise.

1.3 Relevant Formulae

The following equations are relevant for implementing the Bayesian Linear Regression with RBF kernels:

1.3.1 Design Matrix with RBF Kernels

The design matrix Φ is constructed using RBF features:

$$\Phi(X) = \exp\left(-\frac{(X-\mu)^2}{2\sigma^2}\right)$$

where X is the input data, μ represents the RBF centers, and σ is the length scale of the RBF kernels.

1.3.2 Posterior Update Equations

Given the prior mean m_0 and prior covariance S_0, the posterior mean m_N and covariance S_N after observing the data are updated as:

$$S_N = \left(S_0^{-1} + \frac{1}{\sigma_{\text{noise}}^2}\Phi^\top\Phi\right)^{-1}$$

$$m_N = S_N\left(S_0^{-1}m_0 + \frac{1}{\sigma_{\text{noise}}^2}\Phi^\top y\right)$$

where y represents the observed targets, and σ_{noise}^2 is the noise variance.

1.3.3 Predictive Distribution

The predictive posterior mean and variance for a new input X_* are given by:

$$\text{Predictive Mean} = \Phi(X_*)^\top m_N$$

$$\text{Predictive Variance} = \Phi(X_*)^\top S_N \Phi(X_*) + \sigma_{\text{noise}}^2$$

1.4 Instructions

Make a copy of the provided Google Colab Notebook, complete the implementation of the Bayesian Linear Regression with RBF kernel model using the provided code outline. Specifically, you need to:

1. Initialize the model parameters including prior mean, covariance, and RBF kernel parameters.
2. Implement the posterior update equations to update the mean and covariance of the parameter posterior distribution based on the observed data.
3. Compute the design matrix using RBF kernels to transform the input features.
4. Predict the mean and variance for new data points using the predictive distribution formulas.

5. Visualize the results, including the true values, noisy observations, predicted mean, and the predictive uncertainty intervals.

2 RBF Kernel Softmax Classifier: `RBF_Classfn` [5%]

2.1 Introduction

In this problem, you are required to implement a probabilistic classifier using the Radial Basis Function (RBF) kernel and the Softmax function. The goal is to classify data points into different classes based on their features. The classifier will utilize an RBF kernel to compute the similarity between data points and a Softmax function to output class probabilities.

Important Links :

- Colab Notebook : IPR_Assignment_01_Q2
- Dataset : features_q2.npy, labels_q2.npy

2.2 Dataset

It contains 2D feature data points along with their corresponding one-hot encoded labels. The dataset for this assignment is provided in separate files:

- `features_q2.npy` - Contains the input features.
- `labels_q2.npy` - Contains the true values from class label 0 to 9, they are one-hot encoded.

2.3 Relevant Formulae

Below are the key formulae and concepts needed for the implementation:

2.3.1 RBF Kernel Function

The RBF kernel is defined as:

$$K(X_1, X_2) = \exp\left(-\gamma \cdot \text{pairwise\_sq\_dists}(X_1, X_2)\right)$$

where $\text{pairwise\_sq\_dists}(X_1, X_2)$ is the matrix of squared Euclidean distances between each pair of rows from X_1 and X_2, and γ is a parameter that controls the width of the RBF kernel.

2.3.2 Softmax Function

The Softmax function is used to convert logits into probabilities:

$$\text{Softmax}(\text{logits}) = \frac{\exp(\text{logits})}{\sum \exp(\text{logits})}$$

2.3.3 Weight Calculation

The weights for the classifier are obtained by solving the linear system:

$$\text{weights} = (K^+) \cdot y_{\text{one\_hot}}$$

where K^+ is the Moore-Penrose pseudo-inverse of the kernel matrix K.

2.4 Instructions

Complete the missing parts of the provided code. Specifically, you need to:

1. Implement the RBF kernel function.
2. Implement the softmax function.
3. Complete the `fit` method to compute the kernel matrix and solve for the weights.
4. Complete the `predict_proba` method to compute logits and predict probabilities.
5. Load the dataset and prepare the data correctly for training and testing the classifier.

Submission

Make two folders for each question with the names 'Q1' & 'Q2'. Each folder will contain the following :

1. `.ipynb` file of the completed codes, which is to be downloaded after completing the code on Google Colab. Rename the file to `<your_rollNo>.ipynb`.
2. `output.csv` that is generated for each of the files in the attached Google Colab template.

Make a parent folder with both of the above folders in it. Name of the folder should be `<your_rollNo>_A1`, please submit the zipped file of this folder with the name `<your_rollNo>_A1.zip`.

■