HW5

I. Gaussian Process

1. Code (20%)

Task1:

To calculate the covariance of the marginal likelihood, I utilize the rational quadratic kernel, with all its parameters set to 1. Subsequently, I compute the means and variances of the sample points generated by np.linspace(-60, 60, num=1000) using the prediction\_distribution() function. Finally, visualize the results by plotting the computed means and variances.

The formula for rational quadratic kernel and mean and variance of prediction distribution are shown below.

rational quadratic kernel: 一張含有 字型, 筆跡, 行, 文字 的圖片

自動產生的描述

mean and variance of prediction distribution: 一張含有 文字, 字型, 筆跡, 書法 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

一張含有 文字, 字型, 螢幕擷取畫面 的圖片

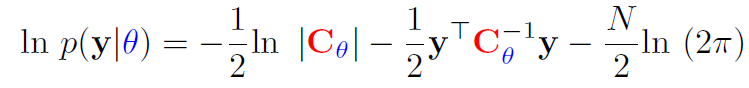
自動產生的描述

一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

Task2:

I used scipy.optimize.minimize to minimize the negative marginal log likelihood. The marginal log likelihood shows in below. The initial parametric of kernel are all set to 1. After I got the new parameters of kernel , I do the same as task1 but using new parameters for kernel. The rational\_quadratic\_kernel() and prediction\_distribution() are same as in task1.



一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

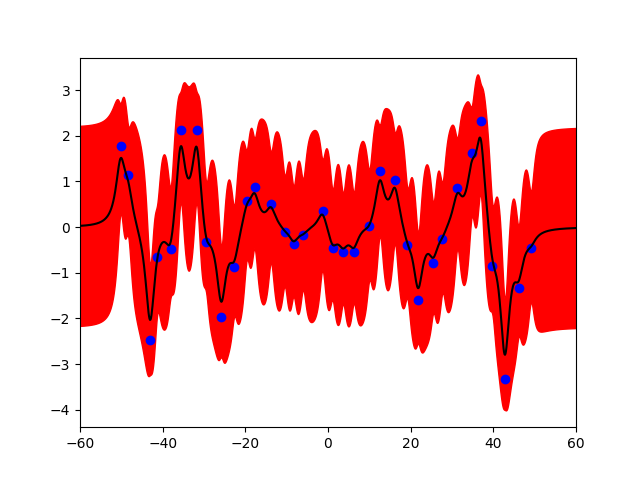
自動產生的描述

2. Experiments (20%)

Task1:

Initial hyperparameters: sigma=1.0, length\_scale=1.0, alpha=1.0

Result:

Image1:

Task2:

Initial hyperparameters: sigma=1.0, length\_scale=1.0, alpha=1.0

Optimal hyperparameters:

sigma=1.3109717036753885

length\_scale=3.3188614346512195

alpha=468.45411131705134

Image2:一張含有 文字, 字型, 繪圖, 行 的圖片

自動產生的描述

3. Observations and Discussion (10%)

After optimizing the hyperparameters, the variance of the predictive distribution appears smaller within the range of sample points compared to before optimization. This phenomenon can be observed in Image1 and Image2. The reason is that the optimized hyperparameters adjust the kernel to better fit the training data, thereby reducing uncertainty in regions with sample points. In contrast, regions far from the observed points still exhibit higher uncertainty since the model has less information.

II. SVM on MNIST

1. Code (20%)

Task1:

Since LIBSVM already provides linear, polynomial, and RBF kernels, I select the corresponding parameters for each kernel and pass them to the train\_and\_evaluate() function. This function trains the model using the specified kernel and evaluates its performance by predicting the results, ultimately calculating the accuracy of the predictions.一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

Task2:

First, I define the parameters of the corresponding kernel for the grid search. Within the grid\_search() function, I retrieve the corresponding parameters from the param\_grid and perform cross-validation for each combination of parameters. The cross-validation results are then compared, and the parameters yielding the highest accuracy are selected and output as the best parameters for each kernel. Finally, I use these parameters to train the model and calculate the accuracy of the predictions. The train\_and\_evaluate function is the same as in task1.

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

Task3:

Since the kernel specified in the specification is not included in LIBSVM, I use the get\_kernel() function to construct a compatible kernel for training. Within get\_kernel(), I construct the linear and RBF kernels based on the formulas provided below. The parameter 𝛾 is chosen as 1/features, which equals 1/784. To calculate the squared Euclidean distance, I utilize cdist.

Once the kernel is constructed, I use the train\_and\_evaluate() function (as defined in Task 1) to train the model and compute the accuracy of the predictions.



一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

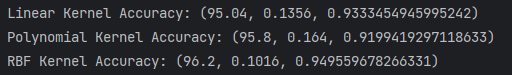
自動產生的描述

1. Experiments (20%)

Task1:

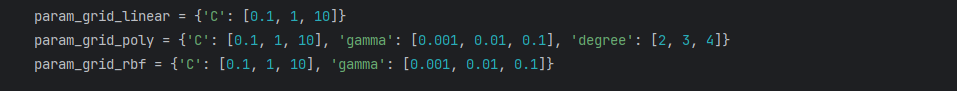
Parameters: C:1, Degree:3, Gamma:1/ num\_features = 1/784

Accuracy format: (accuracy, MSE, SCC)

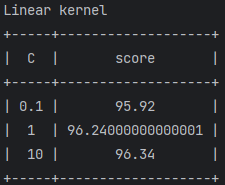


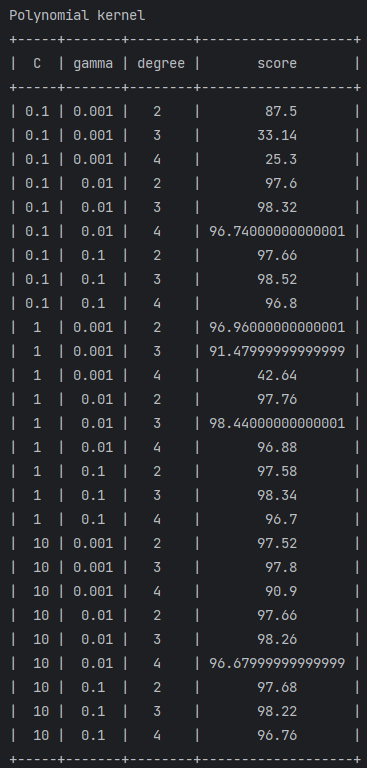
Task2:

Initial parameters: (poly = polynomial)



In grid search:



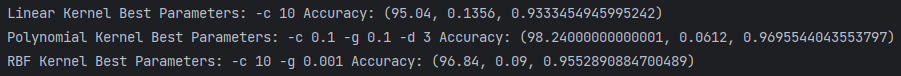
一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

After grid search, the result and its corresponding accuracy:

-c = C, -g = gamma, -d = degree

Accuracy format: (accuracy, MSE, SCC)



Task3:

Parameters: C=1, gamma = 1/784

Accuracy format: (accuracy, MSE, SCC)



1. Observations and Discussion (10%)

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

RBF kernel can have better performance when the gamma is close to 1/num\_feature, in this homework the number of features is equal to 784 which is close to 0.001, so the result that use gamma =0.001 will significant outperform than other conditions. I think the reason is that When γ is set close to 1/num\_features, The kernel scales its influence according to the dimensionality of the feature space. It balances the trade-off between capturing the overall structure of the data and being responsive to local variations, preventing them from overfitting and underfitting.

In task1, linear and RBF kernel have the accuracy of testing data for 95.04% and 96.2% respectively. However, when I sum up these two kernels to be the new kernel, the performance significantly decreases. I think the combination of non-linear and linear components may not align with the data's structure in this homework. Using grid search to tune the hyperparameter may be the solution.