AMATH 582: HOME WORK 3

RAPHAEL LIU

Department of Appliede Mathematics, University of Washington, Seattle, WA
raph651@uw.edu

Abstract. In this assignment, a dataset of the Portuguese "Vinho Verde" red wine's features is given to us. As a data scientist, our task is to develop a supervised learning algorithm to predict a wine's quality based on its 11 attributes measured in laboratory. This is accomplished through different methods including Linear Regression, Kernel Ridge Regression, and Cross-Validation.

1. Introduction and Overview

The Portuguese "Vinho Verde" red wine data set contains 1115 instances for training and 479 instances for test. The data has 11 input features of a wine: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates and alcohol. The 1 output feature is the quality of the wine on a 0-10 score scale. All the features are numeric. Our goal in this homework is to fit our wine features data using 3 different regression models: linear regression, Gaussian kernel ridge, and Laplacian kernel ridge. To optimize our models and prediction, we need to find the optimal parameters accordingly. Below is a list of tasks to complete.

(1) Use linear regression (least squares) to fit a linear model to the training set..

(2) Use kernel ridge regression to fit a nonlinear model to the training set using the Gaussian (RBF) kernel as well as the Laplacian kernel

(3) Use 10-fold CV to tune the length scale σ and the regularization parameter λ for each of the above kernels. Report your choices of the optimal values of σ & λ and provide a clear explanation of why and how you picked those values. Keep in mind that you won't be able to report the "true" optimal values here. We are looking for an informed/good choice given your computational budget.

(4) Provide a table reporting the training and test mean squared errors (MSEs) of all three models: linear regression, and Gaussian and Laplacian kernels with the optimal hyperparameters found via CV. Discuss your findings.

(5) Use your three models to predict the quality of the new batch of wines and report the output of each model on the 0-10 scale.

2. Theoretical Background

2.1. **Supervised Learning.**

Date: February 24 2022.

The function model assumes there exists a function $f^+ : X \to y$ so that $y_i = f^+(x_j) + \epsilon_j$, where ϵ_j are some noise that may be in the output or our observation of the $f^+(x_j)$. By far the most common assumption is the Gaussian noise,

$$\epsilon_j \approx \mathcal{N}(0, \sigma^2)$$

This implies $y_i|x_j \approx \mathcal{N}(f^+(x_j), \sigma^2)$, and that $\Pi(y_j|x_j) \propto exp(-\frac{1}{2\sigma^2}|f^+(x_j) - y_j|^2)$, where Π is the PDF of y for fixed x_j. At this moment, it is useless without a model, since there are many solutions.

2.2. **Linear Regression.**

One of the simplest model is **linear regression**.

$$f_{\text{MLE}} \equiv \beta_{\text{MLE}} = \underset{B\in\mathbb{R}^d}{\arg\min} \frac{1}{2\sigma^2}||A\beta - Y||^2$$

Therefore, MLE is nothing but a least square solution to the problem. Typically, the system is over-determined. Solution is given by solving the normal equations,

$$\frac{\partial}{\partial\beta}(\frac{1}{2\sigma^2}(A\beta - y)^T(A\beta - y)) = \frac{1}{\sigma^2}A^T(A\beta - y) = 0$$

$$\implies A^T(A\beta - y) = 0$$
$$\beta = (A^TA)^{-1}A^Ty$$

2.3. **Kernel Interpolation.**

Suppose we have an interpolation problem, given x data points and y data points. We wish to find $f = \sum_{j=0}^{\infty} c_jF_j$ so we need to solve $\sum_{j=0}^{\infty} c_jF_j(x_i) = y_i, i = 0, ..., N-1$. We want the solution with minimized H_k norm. Therefore, we wish to have

$$minimize \sum_{j=0}^{\infty} c_j^2$$

$$s.t \sum_{j=0}^{\infty} c_jF_j(x_i) = y_i$$

What it reduces to is a solution of the form $f(\mathbf{x}) = \sum_{j=0}^{N-1} a_jK(\mathbf{x_j}, \mathbf{x})$

The interpolation constraints then tell us that $\sum_{j=0}^{N-1} a_jK(\mathbf{x}_j, \mathbf{x}_i) = y_i$, where $\mathbf{a} = (a_0, ..., a_{N-1})$.

$$\implies \Theta\mathbf{a} = \mathbf{y}, \qquad \mathbf{y} = (y_0, ...y_{N-1})$$
$$\Theta_{ji} = K(\mathbf{x}_j, \mathbf{x}_i)$$

Thus, $\mathbf{a} = \Theta^{-1}\mathbf{y}$, now the matrix is invertible provided that it is NDS, and the x_j are distinct.

2.4. **Kernel Ridge Regression.**

The Kernel Ridge Regression is to fit the data with our choice of kernels such as Gaussian kernel and Laplacian kernel, and then model the following:

$$\min_f ||f(X) - Y||^2 + \lambda ||f||^2_{H_k}$$

Given the **Representer Theorem**:

$$\hat{f}(\mathbf{x}) = \sum_{n=0}^{N-1} \hat{a}_n K(\mathbf{x}_n, \mathbf{x}),$$

with $\hat{a} = (\hat{a}_0, ..., \hat{a}_{N-1})$ being minimizer of

$$minimize\ ||\Theta\mathbf{a} - Y||^2 + \lambda \mathbf{a}^T \Theta \mathbf{a}$$

The solution is found by differentiating it, and it is exactly $\mathbf{a} = (\Theta + \lambda I)^{-1} y$. In this assignment, the kernels we are going to use are Gaussian kernel (rbf) and Laplacian kernel, which are given in the form:

$$k_{rbf}(x, x') = exp(-\frac{||x - x'||_2^2}{2\sigma^2}), \quad k_{lap}(x, x') = exp(-\frac{||x - x'||_1}{\sigma}),\ \text{for } x, x' \in \mathbb{R}^{11}$$

It is noted that in some languages, the parameter σ is replaced by γ and the kernels are in the form:

$$k_{rbf}(x, x') = exp(-\gamma||x - x'||_2^2), \quad k_{lap}(x, x') = exp(-\gamma||x - x'||_1 \sigma),\ \text{for } x, x' \in \mathbb{R}^{11}$$

We will keep the convention with parameter σ, and slightly different than that, we seek it in the log_2 form in our implementations.

2.5. **Cross Validation.**

If λ too small, the model is basically memorizing all the train data. The test error is large (over-fitting, high-variance). If λ too large, model is too simply biased. We want the best test error which is the smallest. However, in real life, we don't know the test error.

The idea of Cross Validation is to split the train data to k parts, or k-folds. Randomly permute the data pairs– $\mathbf{x} = \{x_{10}, x_{-1}, ..., x_1 3\}$ and responding $\mathbf{y}$. Then split the data $\mathbf{x}$ & $\mathbf{y}$ into K-subsets. Iterate over $k = 0, ...k = K - 1$ and fit the model to the training data with the k-th fold removed.

Finally, calculate the CV prediction error (CV cost) with changing λ

$$CV(\hat{f}, \lambda) := \frac{1}{N} \sum_{k=0}^{K-1} ||\hat{f}(\mathbf{x}_k, \lambda) - \mathbf{y}_k||^2$$

3. Algorithm Implementation and Development

The programming language used here is Python 3. The procedure for solving the tasks are given below, separately.

(1) First normalize and center the train data set so that the 0 mean and 1 standard deviation are ensured. The normalized data is split into two parts X_{train} and Y_{train} for input features and output feature. Same normalization on test data.

(2) Use sklearn's linear regression method to fit (X_{train}, Y_{train}) data and calculate the MSEs for both training and test.

(3) To get a reasonable range of parameters and reduce computational cost, randomly select 100 instances from the normalized train data, and split them into X_{rand} and Y_{rand} parts for input features and output feature. Generate an equally spaced array for both σ and λ with 20 data points on the interval [-5,5] in the log_2 form.

(4) Build KRR_CV model using sklearn's Gaussian (rbf) kernel with 10-fold cross validation. Iterate over each pair of σ and λ and modify the parameters in sklearn's convention. The overall score is calculated and stored in the manner of negative mean squared error provided by sklearn's cross-validation method. So the maximum of score corresponds to the minimum MSE.

(5) Plot the 2d contour of the scores with respect to $log_2\lambda$ and $log_2\sigma$. Analyze the contour map and obtain a smaller range for finer parameter tuning.

(6) Provided a smaller range of parameters, we then use the complete data sets (X_{train}, Y_{train}) to build Gaussian (rbf) kernel model. Regenerate an equally spaced array for both σ and λ with 20 data points but on the smaller interval. This time we find the optimal parameters $\sigma_{opt}, \lambda_{opt}$. The computation takes about 10 minutes.

(7) Use $\sigma_{opt}, \lambda_{opt}$ to modify the KRR_CV model, fit with (X_{train}, Y_{train}), and predict it on X_{train} and X_{test}. Calculate the train and test MSEs using

$$\text{MSE}_{train} = \frac{1}{\text{length of } Y} \times ||\text{Prediction} - Y||_2^2$$

(8) Repeat the procedures for Laplacian kernel.

(9) Given the wine's new batch data, normalize and center the 11 features based on previous normalization coefficients-mean value and standard deviation. Use the three different models with optimal hyperparameters to predict on the normalized new batch data. The prediction is normalized, so to obtain true prediction we denormalize the outcomes by multiplying the standard deviation and add the mean value back. These prediction will be on the 0-10 scales.

4. Computational Results

For the first task linear regression model, the train MSE and test MSE are recorded in table 1. 1 For the next task, after normalizing and centering the train/test data, we randomly choose 100 instances from the normalized train data. This subset is used on the purpose of finding a reasonable range of parameters & reducing computational cost at the meanwhile. Build Gaussian Kernel Ridge Regression model, and start looping from $\log_2\sigma, log_2\lambda \in [-5, 5]$, we plot the contour map of MSE and MSE standard deviation with respect to each pair of the two parameters.1 From the map, we spot a possible range for σ is $log_2\sigma \in [1.5, 4]$, and a range for λ is $log_2\lambda \in [-3, 1]$.

Now we use the complete train data set, and iterate the parameters in the range just found. Again, the MSE and MSE standard deviation with respect to the combinations of parameters are recorded and plotted in the below contour map.2. The resulting optimal pair of parameters is found to be $\sigma_{opt} = 2^{2.026}, \lambda_{opt} = 2^{-2.789}$ where MSE has its minimum. Modify the model with the optimal parameters, fit the train data and predict it on the train and test set. The train and test MSEs are calculated and recorded in table 1.1

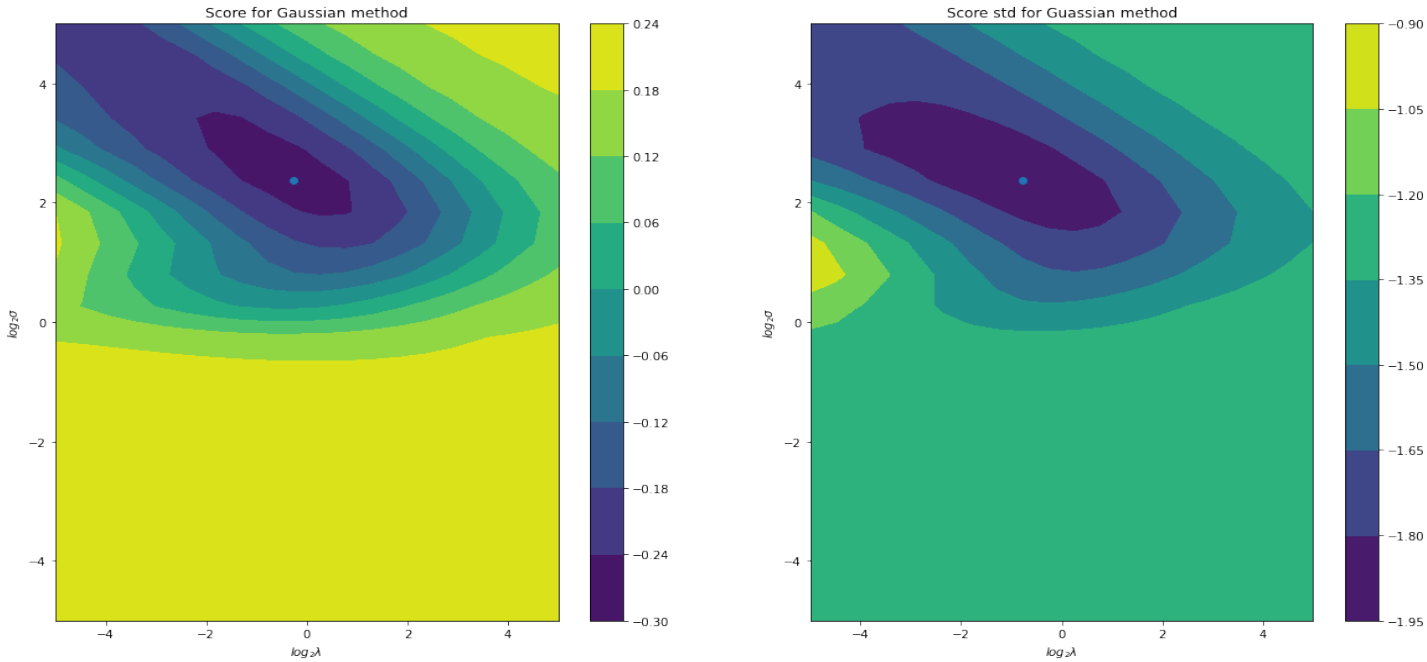

FIGURE 1. Contour map of MSE and MSE std for Gaussian (rbf) kernel, where the blue dot represents where min MSE or MSE std is found.

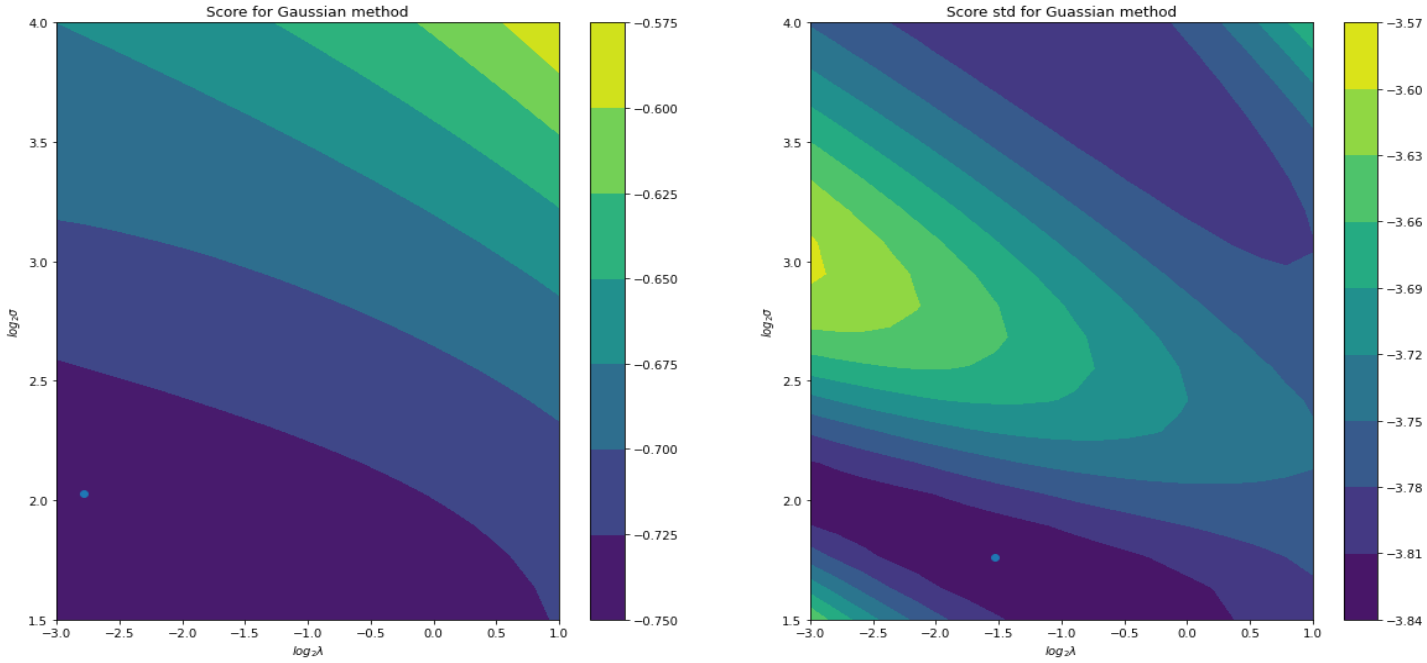

FIGURE 2. Contour map of MSE and MSE std for Gaussian (rbf) kernel, where the blue dot represents where min MSE or MSE std is found. A finer tune for the whole train data set.

The same procedures are conducted for Laplacian kernel method. The first contour map is shown below.3 From here, we choose a finer range of parameters-$log_2\sigma \in [0, 8]$, $log_2\lambda \in [-4, -1]$, and move on to find the optimal parameters fitting the complete train data set. The optimal parameters are found to be $\sigma_{opt} = 2^{2.105}$, $\lambda_{opt} = 2^{-2.105}$. 4 The train and test MSEs are calculated and recorded in table 1.1

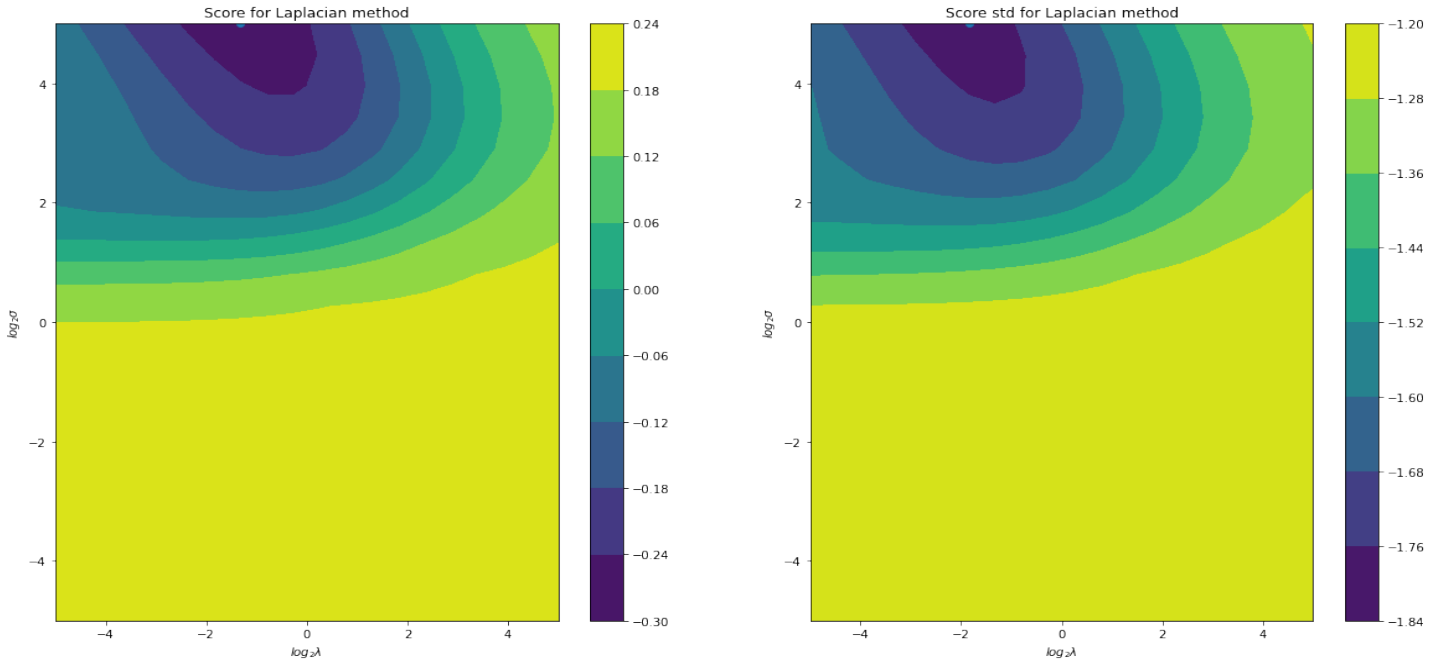

FIGURE 3. Contour map of MSE and MSE std for Laplacian kernel.

MSE	linear	Gaussian kernel	Laplacian kernel
train	0.628	0.460	0.0579
test	0.702	0.649	0.589

TABLE 1. MSE for digits 1,8 classifier with two different α values

The final task is done by first normalizing and centering the wine's new batch data with previous normalization coefficients. The prediction is madee by calling each of the three models. Then the true prediction is found by denormalizing and recentering data back. The final result is shown in table 2.2

Samples	linear	Gaussian kernel	Laplacian kernel
1	6.0047	5.976	6.0471
2	5.2877	5.4329	5.475
3	5.5636	5.3277	5.624
4	6.067	6.1057	5.9732
5	5.9425	6.032	6.0072

TABLE 2. Prediction of scores by each of the three models

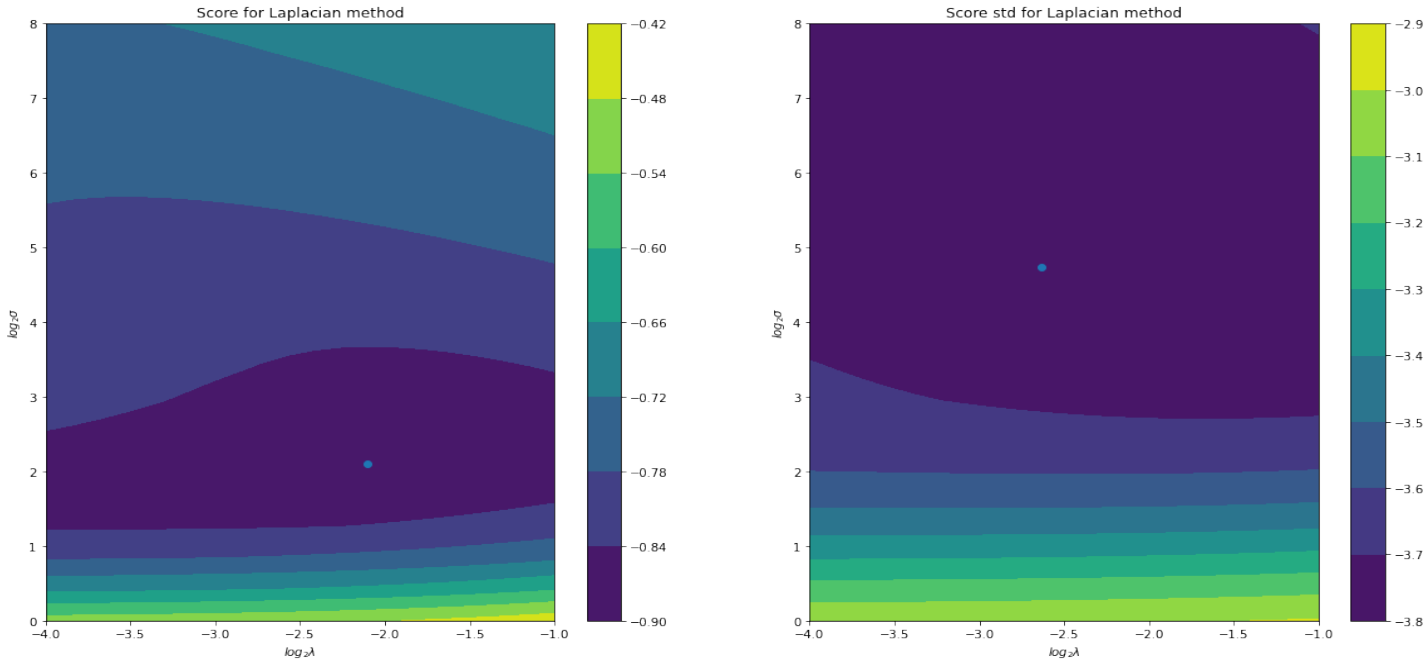

FIGURE 4. Contour map of MSE and MSE std for Laplacian kernel. A finer tune for the whole train data set.

5. Summary and Conclusions

In order to make promising prediction for our Portuguese "Vinho Verde" red wine's new batch quality, there are few steps most importantly to obtain the results. These steps include data normalization, building kernel ridge regression models, conducting cross validations, optimizing hyperparameters, and finally predicting features.

Acknowledgements

The author is thankful to Prof. Hosseini for detailed discussion about supervised learning and his example of predicting housing market in Taiwan with Kernel Ridge Regression method.