AMATH 582: HOMEWORK 3

VINSENSIUS

Applied Mathematics Department, University of Washington, Seattle, WA
vxvinsen@uw.edu

Abstract. A supervised machine learning is done on wine dataset, with 479 test sets, 1115 training sets and 5 new batch of wines [1]. The task is to predict the quality of the wine based on its chemical measurement features. The learning is done through three different methods, linear regression, gaussian kernel regression and laplacian kernel regression. Mean square errors are then obtained to evaluate each method. Based on the error, the laplacian kernel is the most appropriate method to predict the data, followed by gaussian kernel and linear regression method. The prediction of the new batch of wines is then done using the classifiers that has been trained. It is found out that due to uneven distribution of data, that the quality of the new batch is of average quality, between 5 and 6, on 0-10 scale by experts of wine. Furthermore, the hyperparameters λ, the regularization parameter and σ, the kernels parameters are found to be very sensitive for the classifier function such that the classifier does not overfit.

1. Introduction and Overview

Kernel regression is one of methods that can predict data if the data points are highly nonlinear and linear regression can not be used. Thus, kernel regression will map the data into another space such that the transformed data is linear with respect to output. There are many basis function for kernel that are available within the field, but we will only look at gaussian and laplacian kernel [2].

In this project, we will be looking at wine data set [1] and will be looking at the difference between kernel (gaussian and laplacian) and linear regression. The regression will be used to predict wine quality based on 11 chemical measurement. There are 1115 training data set, 479 test data set and 5 data set for prediction. MSE values will be calculated for each regression and will be evaluated. The hyperparameter within the kernel regression will be examined such that the regression will have optimal parameter to produce the smallest error of the training data while keeping the error value reasonable. Finally, the models will do prediction on the quality of the 5 data set based on their chemical measurement.

2. Theoretical Background

Let X denotes the input matrix where each column is the feature of the data points (row-wise), and Y denotes the output (label) vector. Linear regression is where the features of the data is assumed to be linear to the label. The function prediction $\hat{f}$ can be written as:

$$f(\hat{\underline{x}}) = \hat{\beta}_0 + \sum_{j=0}^{d-1} \hat{\beta}_0 x_j \tag{1}$$

where $\hat{\beta}$ are the coefficients of the linear model, the x_j are the features, $\underline{x}$ is the vector of each data points containing the features and d is the number of features. The $\hat{\beta}$ are found by minimizing the square difference between $\hat{f}$ and Y.

Date: February 26, 2022.

Linear regression works great if all the features have linear correlation with the output. But, when the relation becomes highly nonlinear between features and the output, linear regression becomes inaccurate. This is where kernel method comes in. The kernel method essentially introduces a mapping to another space such that the features within the new space is linear to the output. Unsurprisingly, the map is called feature maps. The feature maps let the features of the input map to a higher dimension with new set of basis features [2] [4]. Firstly we define kernel K to be function

$$K(\underline{x}, \underline{x'}) = \sum_{j=0}^{\infty} \lambda_j \psi_j \underline{x} \psi_j \underline{x'} \tag{2}$$

where K is nonnegative, definite and symmetric (like matrices), $\underline{x}$ and $\underline{x'}$ are pair of input vectors (features), λ is the eigenvalue and ψ_j are the eigenvectors that becomes the basis of the features in the new space. We define features map $F(\underline{x})$ to be a matrix of eigenvectors that defines the transformation of the input matrix to a new space where the features are linear to the output of the data. Thus, the kernel can be rewritten in terms of feature maps as

$$K(\underline{x}, \underline{x'}) = (F(\underline{x}), F(\underline{x}))_{l^2} \tag{3}$$

We define the space where the kernel can map to the output vector as reproducing kernel hilbert space (RKHS) or H_k as

$$f(\underline{x}) = \sum_{j=0}^{\infty} c_j F_j(\underline{x}) \tag{4}$$

where $f(\underline{x})$ is define as linear combination of the feature maps define in the new space and c_j are the coefficient to be found to approximate the output. Furthermore, c_j needs to be bounded to provide convergence of the output.

We can focus on the kernel ridge, we define ridge linear regression problem as :

$$\hat{\beta} = \text{argmin}_{\beta \in \mathbb{R}^d}(||\hat{f(x)} - Y||^2 + \lambda ||\beta||^2) \tag{5}$$

where $\hat{\beta}$ is the coefficient of the function prediction $\hat{f(x)}$ and λ is the regularization term. The linear ridge problem is a minimizer problem with some regularization term such that the feature matrix is invertible [9].

In kernel notation, we can define it as

$$\hat{\beta} = \text{argmin}_{f \in H_k}(||f(x) - Y||^2 + \lambda ||f||^2_{H_k}) \tag{6}$$

The kernel ridge problem is similar to linear ridge problem that it is still a minimizer problem to find a function such that it can minimize the difference between the prediction and the output with some regularization term but now the space is in RKHS.

Now we can define Gaussian/Radial Basis Function (RBF) kernel to be [5][2]:

$$K(\underline{x}, \underline{x'}) = exp(-\gamma ||\underline{x} - \underline{x'}||_2^2) \tag{7}$$

where the above is the convention of scikit and $\gamma = \frac{1}{2\sigma^2}$. Since the kernel is defined as distance formula, we can think that each data point in the space has a sphere of influence which depends on σ or γ.

Laplacian kernel is defined to be:

$$K(\underline{x}, \underline{x'}) = exp(-\gamma ||\underline{x} - \underline{x'}||_1) \tag{8}$$

where the above is the convention of scikit and $\gamma = \frac{1}{\sigma}$. Since the kernel is defined as distance formula, we can think that each data point in the space has a space of influence which depends on σ or γ. But the difference with the laplacian kernel that the shape of the space will be similar to cube since it is a 1-norm.

We define the hyperparameters as parameters of the model that are not trained by the model and need to tune manually. The example of this is the regularization parameter in the kernel ridge, λ and the width of the kernel space, σ. The tuning of hyperparameters can be time consuming because it can cause the model to overfit or underfit which can makes the whole modelling inaccurate.

The idea of cross-validation (CV) feature of regression is that to break the data into smaller pieces and remove one of the pieces to validate the model. This way the model can be tune better such as number of features. By using CV, the optimal hyperparameters can be found faster rather than testing it by trial and error in regression model.

To measure the error of the model, the mean-square error (MSE) can be calculated by :

$$\text{MSE} = \frac{1}{N} \sum \left(||y_{predict} - y_{true}||^2 \right) \tag{9}$$

The MSE only shows how much variance between the model and the value can be affected by the distribution of the data.

3. Algorithm Implementation and Development

Python packages that are used:numpy [3], matplotlib [6] and scikit-learn [8].

Training Data First normalize the data such that it has mean of 0 and standard deviation of 1. This is done by subtracting the training data both the input(X) and the output (Y) by its column-wise mean and then divided by the column-wise standard deviation. After that, do the regression, find the optimal hyperparameter (for kernel), and the MSE. For linear regression, use *LinearRegression* function from sklearn to fit. Then, predict the output using the classifier based on the input data. After that, calculate its MSE value. For kernel regression, use *GridSearchCV* function from sklearn. The function is very compact because it is automatically search through optimal hyperparameters based on the range of hyperparameters that it is given and employ crossvalidation too. To use this function, firstly create an classifier object, which is in this case the *KernelRidge* function from the sklearn with appropriate kernel type. Then, create a parameter dictionary that contains the range of hyperparameters that is used in the kernel. When using the *GridSearchCV* set the crossvalidation parameter to 10 fold, since the default is 5 fold. Furthermore, set the scoring to negative MSE such that the function treats the problem as maximizer problem. After that, get the optimal parameters from the from the function, calculate the prediction output and MSE. The method works for each kernel regression type.

Test Data. For the test data, it is similar to the training data except when normalization. The test data will be normalized with respect to training data, both input and output. After that, the procedure will be the same as the training data starting from the predicting the output using the classifier that has been trained using the training dataset.

New Batch For the new batch data, normalized the input with respect the training data set. After that predicting the output using the classifiers, the output will need to denormalize because the output is based on the normalized input. To denormalize, the output will need to multiply by the standard deviation of the training data and then added by the mean of the training data.

4. Computational Results

The section will discuss the results from analyzing the data. The table 3 shows the MSE from three different regression methods, with optimal values used for MSE of kernels. Based on the table, linear regression is overall not suitable for the data since there are 11 features and their correlation are nonlinear with respect to the labels (outputs). Kernel regression is suitable for the wine data set, because their MSEs are overall lower than linear regression. In particular, laplacian kernel regression can fit the data better than gaussian kernel. However, based on MSE, the laplacian kernel might be overfitting the data with the current optimal parameters as it can be seen from the

training data MSE. While gaussian has overall higher MSE than laplacian kernel, it has less bias because the MSE of training and test data set has reasonable difference unlike laplacian kernel.

The optimal parameters for both kernel regression is shown in table 1. The σ and λ is expressed as powers of 2 and the parameters were varied based on the exponent. The exponents are varied from -5 to 5 with 10 data points for each parameters and kernel. And then, the exponent is then substituted into the α and γ definition based on the sklearn. Thus, to get the real value σ and λ, just need to calculate the power of 2 based on the optimal value of the exponent determined by the kernel classifier. The parameters is very sensitive because changing the range or the step size might affect the fit and thus the MSE. If the parameter σ or λ is too low or the step size is too small, the classifier might be overfitting. This is because with smaller σ, the kernel function is essentially has small standard deviation and only evaluating the function value of the training data set. Similarly, with smaller λ, the kernel regression is just minimizing the training data values, and over-fits the model. The opposite is true with higher range or larger step size of parameter, which resulting underfitting model because it is too sensitive of the change.

Figure 1 shows the mean score and the score standard deviation of parameters for gaussian kernels. The plot is done by taking the logarithm of the absolute values of both mean and the standard deviation. Since the scoring is done based on negative MSE values, the lower the score the better the parameter values. By taking the absolute value of it then taking the logarithm of the absolute values, the higher the value of the scores the better. Thus, the optimal values are located in the lighter region within the contour map. The analogous explanation can be applied to laplacian kernel, shown in Figure 2 which has similar contour score plot as the gaussian kernel.

Now we can look at the new batch prediction based on the different methods shown in table 2. The first observation is that the quality prediction across three different methods is around 5 and 6. Thus, the training data might have uneven distribution with data points that are mostly having quality 5 and 6. The training data shows that the proportion of data point of having quality 5 or 6 is about 82% of the entire data. The distribution of the data has very strong effect on the learning model despite different regression models being used. Furthermore, this shows that most of the new batch features have similar features to the training data set which explains the reason why the predictions are very similar despite being applied on different regression models.

Method	λ	σ
Gauss Kernel	0.315	3.17
Laplacian Kernel	0.146	3.17

TABLE 1. The optimal parameters for the two kernel methods based on the scores result shown by the contour plot 2 and 1.

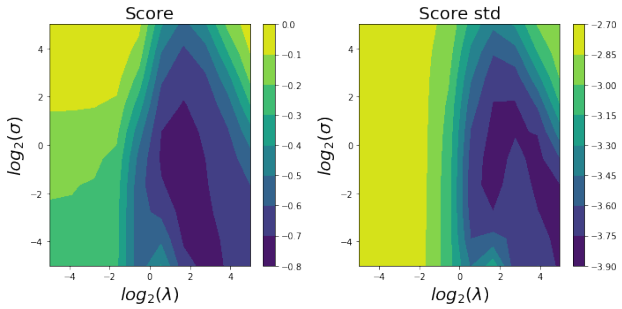

FIGURE 1. The contour plot for mean of scores and the standard deviation of scores based on powers of 2 of σ and λ using Gaussian Kernel.The contour shows that the higher the score the better the classifier, indicated by lighter colour.

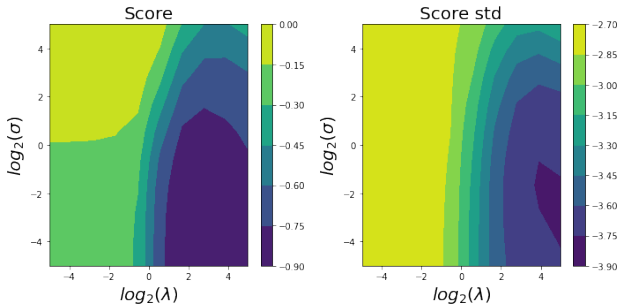

FIGURE 2. The contour plot for mean of scores and the standard deviation of scores based on powers of 2 of σ and λ using Laplacian Kernel.The contour shows that the higher the score the better the classifier, indicated by lighter colour.

New Batch	Linear Quality	Gauss Quality	Laplacian Quality
1	6.00	6.02	5.97
2	5.29	5.45	5.54
3	5.56	5.42	5.61
4	6.07	6.19	5.87
5	5.94	6.12	5.91

TABLE 2. Quality of new batch of wine based on different methods

Methods (Dataset)	MSE (%)
Linear (Training)	62.8
Linear (Test)	74.7
Gaussian Kernel (Training)	44.4
Gaussian Kernel (Test)	67.1
Laplacian Kernel (Training)	2.06
Laplacian (Test)	60.5

TABLE 3. MSE value for training and testing dataset based on the three methods. The MSE values for kernel regressions are calculated at the optimal values.

5. Summary and Conclusions

From wine data set, we can predict quality of the wine based on 11 chemical measurements. We have used three different regression models, linear regression, gaussian kernel ridge regression, and laplacian kernel ridge regression. By using 10-fold kernel regression, optimal hyperparameters were found when they are varied. Then the MSE values are calculated for each regression. Laplacian kernel regression offers better fit of the model but slightly overfit based on the optimal hyperparameters chosen. Gaussian kernel regression is worse fit to the data but less bias based on the optimal hyperparameters chosen. This shows how difficult it is to adjust the hyperparameters such that the model does not overfit or underfit. From regression model, we predict the quality of the new batch of wine and found that the quality is between 5 and 6. It is found out that the quality is largely influenced by the uneven distribution of wine training data that has quality 5 or 6. Thus, the predicted quality of wine might not be real quality because the data is very biased to quality 5 or 6.

To improve for the future, we can find better hyperparameters for both kernel regression such that they can have lower MSE values while not exhibiting bias. Another possible improvement is to have more data to reduce the uneven distribution of the quality 5 and 6 wine. Furthermore, to improve the regression fit, we can explore different kernel methods or introduce more features outside chemical measurement such as production year or features that is related to agriculture such as quality of the crop that is used for wine production [7].

Acknowledgements

The author is thankful to Prof. Hosseni in teaching the material and providing the demo that is similar to the problem. I would like thank Raymond Mead for explaining kernel methods to me. I would like to thank 482/582 discord community for introducing *GridSearchCV* function which helps streamline finding the hyperparameters. The discord community also helps me figure out other programming issue and guidance on the problem.

References

[1] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 1998.
[2] S. Dye. An intro to kernels, Dec 2019.
[3] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020.
[4] B. Hosseini. Introduction to kernel methods. University of Washington-Seattle (LOW 216), Feb 2022. AMATH 482/582.
[5] B. Hosseini. Kernel ridge regression. University of Washington (LOW 216), Feb 2022. AMATH 482/582 Lecture.
[6] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
[7] D. Nguyen. Red wine quality prediction using regression modeling and machine learning, Nov 2020.
[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
[9] Vinsensius. Amath 582: Homework 2, Feb 2022. Unpublished.