# AMATH 482/582: HOME WORK 3

YIXUAN LIU

*University of Washington, Seattle, WA*
*yixuanl@uw.edu*

Abstract. In this project, we will be given a data set that grades the quality of a set of wines by 11 features. The data set contains a training data set and a test data set. We will use three different types of regression to fit linear and nonlinear models to the training set and predict the quality of a new batch of five new wines by each of the three models.

## 1. Introduction and Overview

We have a training data set containing 1115 types of wine and a test data set with 479 types of wine. Each instance of the data has 11 features of chemical measurements. The corresponding output to each set of features is the quality of the wine on a scale of 0 to 10 provided by experts. We will use linear regression to fit a linear model to the training set, the Gaussian kernel ridge regression, and the Laplacian kernel ridge regression to fit nonlinear models to the training set. By finding out the training and test mean squared errors of all three models, we can investigate their performance. We will use the three models to predict the quality of a new batch of five new wines on the 0-10 scale.

## 2. Theoretical Background

A linear regression model [2] is given by the following:

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

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^d}{\arg\min} ||f(x) - \underline{y}||^2$$

A function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is called a kernel [3].

We say $K$ is non-negative definit & symmetric (NDS) if

- $K(\underline{x}, \underline{x}^{'}) = K(\underline{x}^{'}, \underline{x}), \forall \underline{x}, \underline{x}^{'} \in \mathbb{R}^n$
- For any set of points $(\underline{x}_0, ..., \underline{x}_m)$ in $\mathbb{R}^n$, the matrix $(K)_{ij} = K(\underline{x}_i, \underline{x}_j) \in \mathbb{R}^{M \times M}$

is NDS.

*Date*: February 25, 2022.

Given an NDS kernal $K$ we define its reproducing Kernel Hilbert space (RKHS) as the space of function $f : \mathbb{R}^n \to \mathbb{R}$ that have the form

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

that satisfy $\sum_{j=0}^{\infty} c_j^2 {<} {+\infty}$. We wirte $H_K$ to denote this RKHS. The features $F_j$ dictate what the functions inside $H_K$ look like. So in practice we can build our kernel by prescribing its features.

With the understanding of kernel, we can now solve problem of the form

$$\min_{B} ||A\beta - Y||^2 + \lambda||\beta||^2$$

as

$$\min_{f \in H_K} ||f(X) - Y||^2 + \lambda||f||^2_{H_K}$$

with $H_K$ induced by the kernel K

$$K(\underline{x}, \underline{x}^{'}) = \sum_{j=0}^{J-1} F_j(\underline{x}) F_j(\underline{x}^{'})$$

which is the abstruct formulation of Kernel Ridge (KR) Regression [4].

The K-fold Cross Validation (CV) is used to tune $\sigma$ and $\lambda$ in kernel ridge regression and train the two nonlinear models. It will iterate over $k = 0, ..., \underline{K} - 1$ and fit the model to the training data [5]

Gaussian kernel (rbf) in SKlearn is given by

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

and Laplacian kernel is given by

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

and define $\sigma = \sqrt{2\gamma}$, we can get

$$K_{rbf}(\underline{x}, \underline{x}^{'}) = exp(-\frac{||\underline{x} - \underline{x}^{'}||_2^2}{2\sigma^2})$$
$$K_{lap}(\underline{x}, \underline{x}^{'}) = exp(-\frac{||\underline{x} - \underline{x}^{'}||_1}{\sigma})$$

The training and test mean squared error (MSE) on the trained regression model $\hat{f}(\underline{x}) = \sum_{j=1}^{J} \hat{\beta}_j \psi_j(\underline{x})$ is defined as

$$\mathrm{MSE}_{\mathrm{train}}(\hat{f}, \underline{y}) = \frac{1}{N} \sum_{n=0}^{N-1} |\hat{f}(\underline{x}_n) - y_n|^2 \text{ for } \underline{x}_n \in X, y_n \in Y$$

$$\mathrm{MSE}_{\mathrm{test}}(\hat{f}, \underline{y}) = \frac{1}{N'} \sum_{n=0}^{N'-1} |\hat{f}(\underline{x}_n) - y_n|^2 \text{ for } \underline{x}_n \in X', y_n \in Y'$$

## 3. Algorithm Implementation and Development

We will use

- `pandas` [7] to load `.cvs` data set
- `sklearn` [6] to call
  - `kernel_ridge`
  - `linear_model`
    - `LinearRegression`
  - `model_selection`
  - `mean_squared_error` to calculate the training and test MSE
- `numpy` [1] for mathematical operations

We will use rank($A$) by SVD to determine the dimentionality of our train set

## 4. Computational Results

After loading the training, test, and a batch of five new wines' data, we split the training and test set into features and outputs. Normalize all the features included in the new batch by centered training features and all the outputs in the training and test set by centered training outputs. We can now use the centered data to do the following tasks.

We use `LinearRegression` to fit a linear model to the training set and get predicted value for both training set and test set for MSE calculation.

For kernel ridge regression, we use 10-fold CV to tune the length scale $\sigma$ and the regularization parameter $\lambda$ for each of the Gaussian (RBF) kernel and the Laplacian kernel. For RBF kernel, we first choose the range for $\sigma$ and $\lambda$ both to be `np.linspace(-4,4,10)`, and get $log_2\sigma = 2.2222222222222214$ and $log_2\lambda = -4.0$, so we change the range for $\sigma$ to be (1,3) and for $\lambda$ to be (-5,-3). Now we get $log_2\sigma = 2.111111111111111$ and $log_2\lambda = -3.2222222222222223$. To make the values be approximately center of the interval, tune the interval to be (1,3) and (-4,-2). The new value is 1.8888888888888888 and -2.4444444444444446. Repeat the same procedure with (1,3) and (-3,-1), the new value are 1.8888888888888888 and -2.3333333333333335. Since both of them are at about the center of the intervals, so we pick those values. Then calculate $\sigma = 3.7034988491491614$ and $\lambda = 0.19842513149602492$.

For Laplacian kernal, we do the similar steps above: first we start at intervals (-4,4) and (-4,4) and get 2.2222222222222214 and -2.2222222222222223. Tune the intervals to be narrower as (1,3) and (-3,-1), then get the values 2.111111111111111 and -2.111111111111111. Since both of them are at about the center of the intervals, so we pick those values. Then calculate $\sigma = 4.320238955569224$ and $\lambda = 0.2314686780718226$.

| | Gaussian | Laplacian |
|---|---|---|
| $\sigma$ | 3.7034988491491614 | 4.320238955569224 |
| $\lambda$ | 0.19842513149602492 | 0.2314686780718226 |

By definition

$$\alpha = \lambda$$
$$\gamma_{rbf} = \frac{1}{2\sigma_{rbf}^2}$$
$$\gamma_{lap} = \frac{1}{\sigma}$$

we get

| | Gaussian | Laplacian |
|---|---|---|
| $\alpha$ | 0.19842513149602492 | 0.2314686780718226 |
| $\gamma$ | 0.03645403248675365 | 0.23146867807182261 |

Now we can use the $\alpha$ and $\gamma$ to fit ridge regression on the training set for both Gaussian and Laplacian kernel. Get predicted value for both training set and test set to calculate MSE. Combined with linear model, we get the following table of MSE:

| | Linear | Gaussian | Laplacian |
|---|---|---|---|
| training MSE | 0.6278484956554882 | 0.4548788888959536 | 0.057890626651083514 |
| test MSE | 0.7471696905187208 | 0.6786661476640042 | 0.6077484857863533 |

We can see that both training and test MSE for the linear model is relatively large, so for this project linear model is not very useful. However, the training MSE for the Laplacian kernel nonlinear model is really small, this might due to overfitting.

Now we can use our three model to predict the quality of the new batch of wine. The score get from linear model is [6.00469789 5.28767761 5.56363072 6.067022 5.94248207], from Gaussian kernel is [5.99233072 5.44373019 5.36230769 6.14112495 6.06319855], and for Laplacian kernel is [6.0483042 5.47399545 5.62433419 5.97466709 6.00854608]. The rounded scores are [6. 5. 6. 6. 6.], [6. 5. 5. 6. 6.], and [6. 5. 6. 6. 6.] respectively. To visulize them, we have the following table with rounded scores:

| Linear | Gaussian | Laplacian |
|---|---|---|
| 6 | 6 | 6 |
| 5 | 5 | 5 |
| 6 | 5 | 6 |
| 6 | 6 | 6 |
| 6 | 6 | 6 |

## 5. Summary and Conclusions

We use three different models to predict a batch of five new wines by fitting to the training set. However, all models have relatively high test mean squared errors, and both the linear model and Gaussian kernel ridge regression model have relatively high training values. The reason that the Laplacian kernel ridge regression has a relatively low training MSE might be overfitting. So neither of them would be an optimal model to predict new wine. By checking the scores in the training and test set, most of them are 6, so this might lead to our prediction of the batch of new wine have mostly score of 6.

## Acknowledgements

The author is thankful to every thought in the AMATH 582/482 Discord server, and Professor Hosseini for the code in lecture 16 to implement data centereing and cross validation.

## References

[1] 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.

[2] B. Hosseini. Evaluating sl models. University of Washington (LOW 216), Jan 2022.

[3] B. Hosseini. Introduction to kernel methods. University of Washington (LOW 216), Jan 2022.

[4] B. Hosseini. Kernel ridge regression. University of Washington-Seattle (LOW 216), Jan 2022. AMATH 482/582.

[5] B. Hosseini. Model tuning with cross validation. University of Washington-Seattle (LOW 216), Jan 2022. AMATH 482/582.

[6] 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.

[7] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.