AMATH 482/582: HOMEWORK 3

JONATHAN BEAUBIEN

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

Abstract. We have been tasked with developing an algorithm that can predict the quality of wine (0-10) from a series of chemical measurements. By testing multiple linear regression versus kernel ridge regression with Laplacian and Gaussian (RBF) kernels, we found that we could improve our accuracy using the non-linear ridge regression models. To obtain optimal hyper-parameters for Kernel regression we used 2D cross-validation.

1. Introduction and Overview

Our analysis requires data about wine's that have already been rated for training purposes and those for testing the model after it will be created. The features of our data set are as follows: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol. The label is simply the quality of the wine from 0 to 10. The training set has 1115 instances while our test set contains 479 instances.

The goal of training a regression model is to have an algorithm to rate wines in the future. We have a data set of 5 new wines that we will then predict the quality with each model.

The models we will train will use linear regression, kernel ridge regression with a Gaussian kernel, and kernel ridge regression with a Laplacian kernel. The kernel regressions require parameter optimization so we will use cross-validation to determine optimal values. We will show a comparison of all the models in terms of their mean squared error as an measure of accuracy.

2. Theoretical Background

The first model we will train will use linear regression. This is our simplest model because it only contains linear combinations of our features. The equations are from [2]

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

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \|f(x) - \underline{y}\|^2 \tag{2}$$

For kernel ridge regression, we first need to understand normal ridge regression. The original equation for ridge regression was as follows (proceeding equations (3)-(8) are from [3]),

$$\hat{\underline{\beta}} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2\sigma^2} \|A\underline{\beta} - \underline{Y}\|^2 + \underline{\lambda}_2 \|\underline{\beta}\|^2 \tag{3}$$

Date: February 25, 2022.

where A was our feature matrix often of the form

$$A = \begin{bmatrix} F_0(\underline{x}_0) & F_1(\underline{x}_0) & \dots \\ F_0(\underline{x}_1) & F_1(\underline{x}_1) & \dots \\ & \vdots & \\ F_0(\underline{x}_{N-1}) & F_1(\underline{x}_{N-1}) & \dots \end{bmatrix}$$

This means that we could pick the features $F_j : \mathbb{R}^d \to \mathbb{R}$ lending to a model of the form

$$f(\underline{x}) = \sum_{j=0}^{J-1} \beta_j F_j(\underline{x}) \approx y(\underline{x}) \tag{4}$$

Our knowledge of kernels now reveals to us that underneath this model is the kernel

$$K(\underline{x}, \underline{x}') = \sum_{j=0}^{J-1} F_j(\underline{x}) F_j(\underline{x}') \tag{5}$$

with features $\Psi_j : \mathbb{R}^d \to \mathbb{R}$ and whose RKHS,

$$H_K := \left\{ f : \mathbb{R}^d \to \mathbb{R} | f(x) \text{ has form (4)} \right\}$$

In other words, (1) is equivalent to the problem

$$\underset{f \in H_K}{\text{minimize}} \, \|f(X) - Y\|^2 + \lambda \|f\|_{H_K}^2 \tag{6}$$

with H_K induced by the kernel K above. We will use two different kernels including the Gaussian kernel

$$K_{\text{rbf}}(\underline{x}, \underline{x}') = \exp\left(-\frac{\|\underline{x} - \underline{x}'\|_2^2}{2\sigma^2} \right) \tag{7}$$

and the Laplacian kernel

$$K_{\text{lap}}(\underline{x}, \underline{x}') = \exp\left(-\frac{\|\underline{x} - \underline{x}'\|_1}{\sigma} \right) \tag{8}$$

To find out good values for λ and σ we need to use 2D Cross Validation. We will use a range of $\log_2(\lambda) \in (-4, 4)$ and $\log_2(\sigma) \in (-4, 4)$ where $\delta \log_2(\lambda) = \delta \log_2(\sigma) = \frac{4-(-4)}{20}$ (20 equally spaced intervals) for both kernel regression models where the optimal values are as follows [4]:

$$\lambda^*, \sigma^* = \underset{\log_2(\lambda) \in (-4,4), \log_2(\sigma) \in (-4,4)}{\text{argmin}} CV(\hat{f}, \lambda, \sigma)$$

After we determine our $\hat{f}$, we can compute the Mean Standard Error (MSE) of each set of data (training and test). If $\{X, Y\} \to$ training set and $\{X', Y'\} \to$ test set then our MSE equations are as follows [2]:

$$MSE_{\text{train}} = \frac{1}{N} \sum_{n=0}^{N-1} |\hat{f}(\underline{x}_n) - y_n|^2, \quad \underline{x}_j \in \mathcal{X}, y_j \in \mathcal{Y} \tag{9}$$

$$MSE_{\text{test}} = \frac{1}{N} \sum_{n=0}^{N-1} |\hat{f}(\underline{x}_n) - y_n|^2, \quad \underline{x}_j \in \mathcal{X}', y_j \in \mathcal{Y}' \tag{10}$$

3. Algorithm Implementation and Development

The scikit-learn [5] implementations of linear regression and kernel ridge regression were used in a Python environment. The 2D cross validation was done by using some nested for loops provided by Prof. Bamdad. The MSE was retrieved from the regression classifier's attributes. NumPy [1] was also used for matrix and array manipulation.

4. Computational Results

The first important result we will share is the MSE for both the training and test data for each model. We haven't included the actual hyper-parameters here, they will be in the next section.

Regression Model	$\text{MSE}_{\text{train}}$	MSE_{test}
Linear Regression	0.6278	0.7472
Gaussian KRR	0.4563	0.6791
Laplacian KRR	0.0399	0.6057

As we can see, if our measure of accuracy is the MSE_{test} then the Laplacian KRR is the most effective model. However, we can see that the Laplacian KRR model may be over-fitted because of the extremely small $\text{MSE}_{\text{train}}$.

After running 2D cross validation of the hyper-parameters λ and σ with 20 equally spaced intervals we found the optimal values (the values where the cross validation scores are the lowest in magnitude) to be as follows,

Kernel Function	$\log_2(\lambda^*)$	$\log_2(\sigma^*)$
Gaussian	-2.4210	1.8947
Laplacian	-2.3157	1.8947

These values were determined by first doing a 10 intervals on a range of $(-4, 4)$ and then narrowing the range by graphing the cross-validation scores versus $\log_2(\lambda)$ and $\log_2(\sigma)$ and upping the intervals to 20. While they might not be perfect, it takes 10 minutes for the cross-validation to run so I could not try too many values.

Lastly, we will see what our models have predicted for the quality of the 5 wines that are in our 'new batch'.

Regression Model	Wine 1	Wine 2	Wine 3	Wine 4	Wine 5
Linear Regression	6	5	6	6	6
Gaussian KRR	6	5	5	6	6
Laplacian KRR	6	6	6	6	6

Even though the Laplacian KRR seems to be over-fit, because its MSE_{test} is also the lowest, I am inclined to believe that all of the wines are of quality 6. Of all the wines, the most variable are wines 2 and 3.

5. Summary and Conclusions

We were able to train multiple regressions including linear, Gaussian KRR, and Laplacian KRR models. By tuning the hyper-parameters of the KRR models using cross-validation, we were able to obtain a smaller MSE_{test} than the linear model which means they are capturing some non-linear aspects of the training data. While our Laplacian KRR model may be overfitted, because of its success with the testing data it should be the most accurate model. An individual in need

of a quality ranking of their red wine could simply plug-in the characteristics mentioned in the introduction and receive a quality score as we did with the 5 new wines.

Acknowledgements

The author is thankful to Katherine Owens for her advice on calculating MSE and general tips. The author is also thankful to Ava Mistry for her help regarding citations and relevant equations. Thank you to Sathvik Chinta for help with printf statements and reference to sklearn metrics functions.

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 supervised learning models. University of Washington-Seattle (LOW 216), Feb 2022. AMATH 482/582.

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

[4] B. Hosseini. Model training with cross-validation. University of Washington (LOW 216), Feb 2022.

[5] 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.