AMATH 482 HOME WORK 3
QUALIFYING RED WINE

LANG QIN

Amazing Department, University of Washington, Seattle, WA
lq25@uw.edu

Abstract. The goal of this work is to develop regression models to qualifying red wine based on 11 quantitative features of wine. This work implements three regression models: 1) least sqaures regression, 2) kernel regression using Gaussian (RBF) kernel, and 3) kernel regression using Laplacian kernel. This work consists of three sub-tasks in order to achieve the goal: 1) normalize input data to fit the model, 2) develop and evaluate each regression model, and 3) use the fitted model to predict the quality of the batch of 5 new wines.

1. Introduction and Overview

The purpose of this work is to develop an algorithm that predicts the quality of wine from a series of 11 chemical measurements: 1) fixed acidity, 2) volatile acidity, 3) citric acid, 4) residual sugar, 5) chlorides, 6) free sulfur dioxide, 7) total sulfur dioxide, 8) density, 9) pH, 10) sulphates, and 11) alcohol. The above 11 numeric measurements are the *features* of the regression model and the numeric measurement (in range 0 to 10) of the quality of the wine is the *label*. The training set contains 1115 training instances and 479 test instances. The additional information about the new batch of 5 wines contains the same 11 measurements for quality prediction.

The following three models are implemented for quality regression: 1) linear regression (least squares), 2) Gaussian (RBF) kernel ridge regression, and 3) Laplacian kernel ridge regression. The mean square error is used to evaluate the performance of the above models.

2. Theoretical Background

To achieve the goal, this work primarily relies on the following three points: 1) linear least squares regression and 2) kernel ridge regression with cross-validation.

2.1. Linear Least Squares Regression.

Let the features of instances stored in a matrix $\mathbf{X}$, where each column represents one measurement of the features. Let the labels of instances stored in a column vector $\mathbf{y}$, where each entry represents the label (quality) of an instance. The purpose of linear regression is to find the best linear combination of features that adds up to the labels. Mathematically, we are going to find a vector $\beta \in \mathbb{R}^{n+1}$, such that

$$\mathbf{y}_i = \beta_0 + \beta_1 \mathbf{X}_{i1} + \beta_2 \mathbf{X}_{i2} + \cdots + \beta_n \mathbf{X}_{in} = \mathbf{X}_i \beta, \tag{1}$$

where $\mathbf{X}_{ij}$ represents the j-th feature measurement of the i-th instance and β_j is the constants corresponding to the j-th feature measurement. Often these n instances are stacked together and written in matrix notation as

$$\mathbf{y} = \mathbf{X}\beta, \tag{2}$$

Date: February 26, 2022.

where

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} \mathbf{x}_1^\mathsf{T} \\ \mathbf{x}_2^\mathsf{T} \\ \vdots \\ \mathbf{x}_n^\mathsf{T} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}.$$

Then, we could obtain the target β by solving the least squares problem

$$\underset{\beta}{\operatorname{argmin}} ||\mathbf{X}\beta - \mathbf{y}||^2. \tag{3}$$

2.2. **Kernel Ridge Regression with Cross-Validation.** Kernel ridge regression (KRR) combines ridge regression (linear least squares with l2-norm regularization) with the kernel trick. It thus learns a linear function in the space induced by the respective kernel and the data. For non-linear kernels, this corresponds to a non-linear function in the original space.

2.2.1. *Kernel.* A function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is called a kernel. A non-negative definite and symmetric kernel satisfies

$$K(x, x') = \sum_{j=0}^{\infty} F_j(x) F_j(x'), \tag{4}$$

where $F_j : \mathbb{R}^N \to \mathbb{R}$ are the features of K and F are the basis in L^2 space, forming a Reproducing Kernel Hilbert space [1], [3]. The below are two kernels used for this work.

1. Gaussian (RBF) kernel:

$$K(x, x') = \exp\left(-\gamma ||x - x'||^2\right) \tag{5}$$

2. Laplacian kernel:

$$K(x, x') = \exp\left(-\gamma ||x - x'||\right) \tag{6}$$

2.2.2. *Kernel Ridge Regression.* The approach to ridge is to solve problems of the form

$$\min_{\beta} ||\mathbf{A}\beta - \mathbf{y}||_2^2 + \lambda ||\beta||_2^2, \tag{7}$$

where $\mathbf{A}$ is the feature matrix often of the form

$$A = \begin{pmatrix} F_0(x_{10}) & \cdots & F_p(x_{1p}) \\ F_0(x_{21}) & \cdots & F_P(x_{2p}) \\ \vdots & \ddots & \vdots \\ F_0(x_{n0}) & \cdots & F_p(x_{np}) \end{pmatrix}.$$

The complexity parameter $\lambda \geq 0$ controls the amount of shrinkage: the larger the value of λ, the greater the amount of shrinkage and thus the coefficients become more robust to collinearity. Applying kernel defined in Section 2.2.1, the ridge problem can be rewritten as

$$\underset{f \in H_K}{\operatorname{argmin}} ||f - \mathbf{y}||^2 + \lambda ||f||_{H_K}^2, \tag{8}$$

where $H_k := \{f : \mathbb{R}^d \to \mathbb{R} | f(x) = \sum_{j=0}^{J-1} \beta_j F_j(x)\}$ and $||f||_{H_K}^2 = \sum_{j=0}^{J-1} \beta_j^2 = ||\beta||^2$. By Representer Theorem, Eq. 8 can be further expressed as

$$\underset{\mathbf{a} \in \mathbb{R}^N}{\operatorname{argmin}} ||\Theta\mathbf{a} - \mathbf{y}||^2 + \lambda \mathbf{a}^T \Theta \mathbf{a}, \tag{9}$$

where $\Theta_{ij} = K(x_i, x_j)$ [2]. This is similar to the case of penalized least squares and has an exact solution

$$\hat{\mathbf{a}} = (\Theta + \lambda I)^{-1} \mathbf{y}. \tag{10}$$

2.2.3. *Cross-Validation.* The data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other $k-1$ subsets are put together to form a training set. Then the average error across all k trials is computed [4].

Define the CV prediction error (CV cost) of prediction function $\hat{f}$ with penalty parameter as

$$CV(\hat{f}, \lambda) := \frac{1}{N} \sum_{k=0}^{k-1} ||\hat{f}_{-k}(\hat{\mathbf{x}}_k, \lambda) - \hat{\mathbf{y}}_k||^2. \tag{11}$$

With above cost function, an optimal choice of λ is obtained by minimizing the CV loss, i.e.

$$\lambda^* = \underset{\lambda \in (0, +\infty)}{\operatorname{argmin}} \; CV(\hat{f}, \lambda). \tag{12}$$

3. Algorithm Implementation and Development

The implementation of this project can be divided into three parts: 1) Environment Setup, 2) Data Normalization, and 3) Model Development. This programming work is done by Google Colab. The `Python` libraries imported for this work include: `google.colab.drive`, `numpy`, `matplotlib.pyplot`, and `sklearn`.

3.1. **Environment Setup.** Import data from `wine_training.csv`, `wine_test.csv`, and `wine_new_batch.csv` as training, test, and new batch data sets respectively. The *features* are the first 11 columns of data for both training and test sets and the 12th column is the quality of wines, the *label*. New batch set contains only the *features* for quality prediction of 5 wines. Extract the *features* and *label* of each set for further steps.

3.2. **Data Normalization.** Normalize and center the *features* and *label* so that they have mean 0 and standard deviation 1. This is done by subtracting the column mean from *features* and *label* data sets and divided the subtracted data by their standard deviation.

3.3. **Model Development.** This parts develops 3 regression models: 1) linear least squares regression, 2) Gaussian kernel ridge regression, and 3) Laplacian kernel ridge regression.

3.3.1. *Linear Least Squares Regression.* Based on Section 2.1, the matrix $\mathbf{X}$ is constructed by concatenate a column of ones to the matrix of training *features* at the front. Then, use `numpy` function `linalg.lstsq` to solve the least squares problem of Eq.3 for β. Then, use the computed β to evaluate the model over training and test sets via mean sqaure error, and output the quality of the new batch predicted by the fitted model.

Attention: the new batch quality can be obtained by denormalizing the predicted data on standard deviation and center. Mathematically, this step is done by

$$\texttt{predicted\_batch} \cdot \texttt{training\_label\_std} + \texttt{tranining\_label\_mean}. \tag{13}$$

3.3.2. *Kernel Ridge Regression.* Based on Section 2.2, two kernel ridge regression are implemented and tuned to fit a nonlinear model to the training set using the Gaussian (RBF) kernel and the Laplacian kernel. The model is tuned via 2-D 10-fold cross-validation and the mean square errors of the training and test sets are used in model evaluation. The new batch is obtained by Eq.13.

The ridge regression with Gaussian (RBF) kernel defined by Eq.5 and Laplacian kernel defined by Eq.6 are implemented in Algorithm 1. By examining the score matrix and the contour plot of scores vs σ, λ, iterate Algorithm 1 with tuned σ-s and λ-s until a good-enough model is found. In particular, start with a large range of σ-s and λ-s and then manually shrink the range iteration by iteration based on the score reports and contour plots. Then, evaluate the best tuned model by mean square error, and predict the new batch quality by Eq.13.

4. Computational Results

4.1. **Kernel Ridge Parameters.** The best parameters found after numerous trials for Gaussian kernel ridge regression are $\sigma = 1.888$ and $\lambda = -2.222$; for Laplacian kernel ridge regression are $\sigma = 2.111$ and $\lambda = -2.222$.

The counter plots associated with the best parameters are shown in Figure 1.

Algorithm 1: Kernel Ridge Regression with Cross-Validation

Input : *Sigma* - List of σ-s to tune; *Lambda* - List of λ-s to tune; k - the type of kernel to use
Output: *scores* - List of model score; *std* - List of score's standard deviation

```
// Initialize output storage
Initialize scores and std
// Initialize regressor
KRR ← kernel ridge regressor using kernel k
// Cross-validation
foreach σ ∈ Sigma do
    KRR.gamma ← σ
    foreach λ ∈ Lambda do
        KRR.alpha ← λ Perform 10-fold cross-validation over KRR
        Record the model score in scores
        Record the score standard deviation in std
    end foreach
end foreach
```

Regressor	Training MSE	Test MSE
Linear least squares	0.6278484956554884	0.7471696905187203
Gaussian kernel ridge	0.45846142299952847	0.6791289111901163
Laplacian kernel ridge	0.052650423557821985	0.6079848496431081

TABLE 1. MSE of training and test sets with different regression implementation.

4.2. **Model Performance by Mean Square Error.** The mean square errors of the models by linear regression, kernel ridge regression with Gaussian kernel, and kernel ridge regerssion with Laplacian kernel are listed in Table 1.

According to Table 1, Laplacian kernel ridge regression performs the best in both training and test set with mse about 0.053 and 0.607. In general, the performance the kernel ridge regressions with either Gaussian or Laplacian kernel is better than that of linear least squares regression. Among the above three regressors, kernel ridge regression with Laplacian kernel is the best.

4.3. **Predicted New Batch Quality.** The qualities of the new batch of wines predicted by the three models over the 0-10 scale are listed in Table 2.

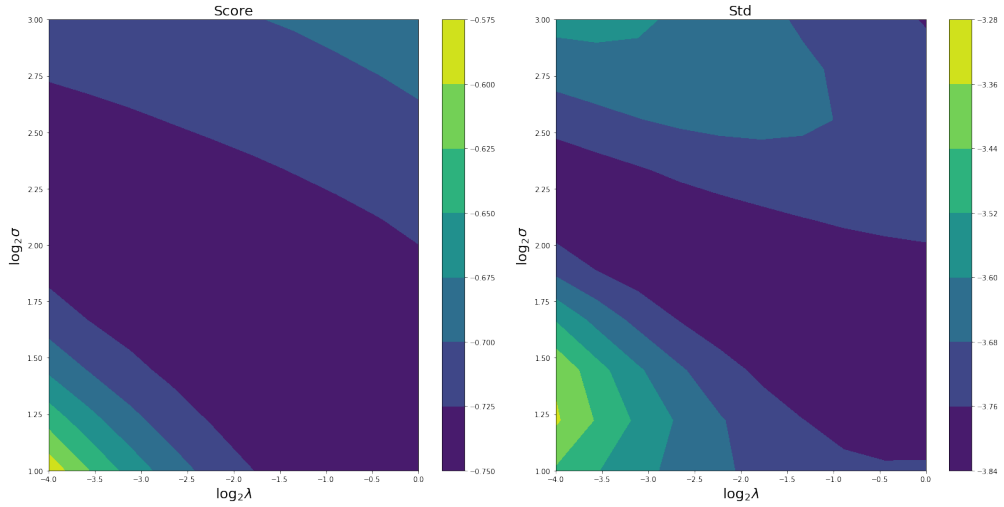
Regressor	Wine 1	Wine 2	Wine 3	Wine 4	Wine 5
Linear least squares	6.00469789	5.28767761	5.56363072	6.067022	5.94248207
Gaussian kernel ridge	5.99359168	5.43331001	5.36499355	6.1344601	6.05712892
Laplacian kernel ridge	6.05116626	5.47305444	5.6265591	5.9765956	6.01032488

TABLE 2. MSE of training and test sets with different regression implementation.

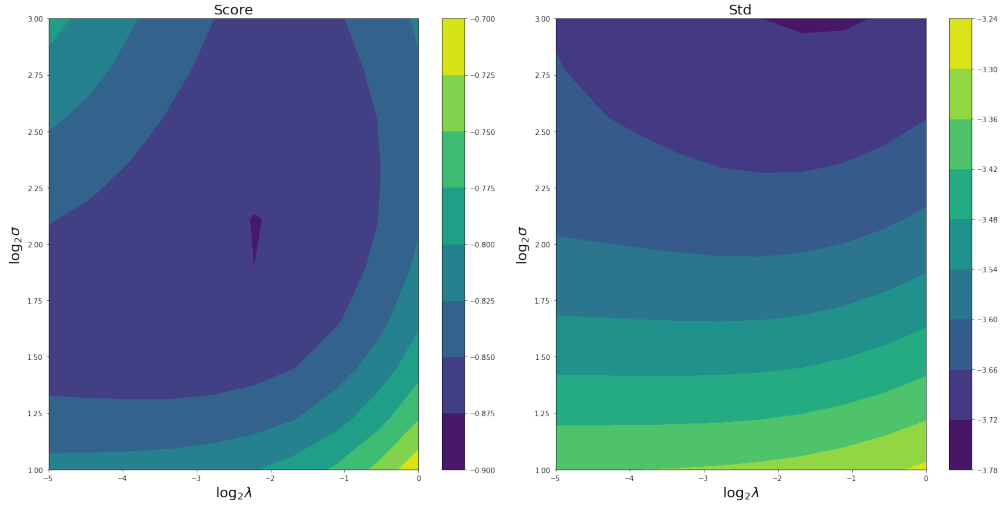
According to the data in Table 2, the qualities of the wines in the new batch are around scale 6.

5. Summary and Conclusions

Through linear least squares regression and kernel ridge regression with Gaussian kernel as well as Laplacian kernel, this work successfully develops three wine quality regressors given measurements described in Section 1, and this work uses the fitted models to predict the quality of the new batch of wines according to the measurements. Among these regressors, kernel ridge regression with Laplacian performs the best with the given training and test data sets according to their mean square errors. Thus, it is reasonable to conclude that the wine qualities from the new batch should be the same range as predicted by kernel ridge regression with Laplacian kernel shown in Table 2.



(A) Gaussian Kernel



(B) Laplacian Kernel

FIGURE 1. Final contour plots of σ, λ, and scores for Gaussian and Laplacian kernel

Although the kernel ridge regressors are trained with cross-validation for multiple trials, the manual tuning inputs for cross-validation may have missed some better parameter options. The future work could be done to improve this potential issue.

ACKNOWLEDGEMENTS

The author is thankful to Professor Bamdad Hosseini's announcement about suggested references for kernel methods.

REFERENCES

[1] K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf. Kernel mean embedding of distributions: A review and beyond. 2017.
[2] C. E. Rasmussen and W. C. K. I. *Gaussian processes for machine learning*. MIT Press, 2006.
[3] B. SCHLKOPF. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press, 2018.
[4] J. Schneider. Cross validation, Feb 1997.