AMATH 563 Homework 1: Regression, Model Selection and DMD

Gloria(Tiange) Tang

4/20/2020

Abstract

This MNIST dataset study compares the method of least squares, ridge regression and lasso regression to find a best mapping from the image space to the label space, and determines the most informative pixels for hand-written digit prediction by promoting sparsity.

1 Introduction and Overview

The MNIST database (Modified National Institute of Standards and Technology database) is a database of handwritten digits containing 60,000 training images and 10,000 testing images. This study consists of three major parts: the first part *Mapping Determination* uses Least Square Fit, Ridge Regression and Lasso Regression to find the best mapping from the image space to the label space; the second part *Feature(Pixels) Selection of Multiple Digit Prediction* determines the most important pixels for all digit prediction using Lasso with multiple empirical rules to promote sparsity; the third part *Feature(Pixels) Selection of Single Digit Prediction* redos the second analysis with each digit individually to find most important pixels for each digit.

2 Theoretical Background

2.1 Overdetermined System

An overdetermined system is a linear system of equations

$$A_{m*n}x = b$$

where $m > n$(more equations than unknowns). To solve this system, we need to find

$$\arg\max_x(||Ax - b||_2 + \lambda g(x))$$

where λ is user-defined magnitude of regularization and $g(x)$ is the regularization term. The method is called Least Square Fit when we choose $\lambda = 0$ to

approximate the solution of overdetermined systems by minimizing the sum of the squares of the residuals made in the results of every single equation. We can add regularization to solve the system by making λ nonzero: if $g(x) = ||x||_1$,the L1-norm of the parameter matrix, we call it a Lasso method; if $g(x) = ||x||_2$,the L2-norm of the parameter matrix, we call it a Ridge method. One advantage of Lasso over Ridge Regression is that in Ridge Regression, as the penalty is increased, all parameters are reduced while still remaining non-zero, while in Lasso, increasing the penalty will cause more and more of the parameters to be driven to zero. Lasso discards unimportant features for prediction and it is great for feature selection.
The data used in this study, MNIST database, has 70,000 images in total whose size is $28*28$ each, yielding in an input matrix A of 70,000 rows and 784 columns. In the first part *Mapping Determination*, we are solving an overdetermined system

$$Ax = b$$

with the methods mentioned above: Least Squre Fit, Ridge and Lasso to determine which mapping gives the best performance. For the following parts, we retain the loading matrices of Lasso because Lasso has done the dimensionality shrinkage by making less predictive pixels' coefficients zeros.

2.2 Pareto Optimality

Pareto Optimality is an economic concept stating that *An economy is in a Pareto Optimal state when no further changes in the economy can make one person better off without at the same time making another worse off.* In the matter of machine learning, facing thousands of predictive models, we need to have a trade-off between the number of features and the model performance. The more features a model uses the more likely it is overfitting the training set hence drives down the performance of the testing set. Reaching Pareto Optimality is when we have a good balance between the model's interpretability and performance.
Our goal in the second part *Feature(Pixels) Selection of Multiple Digit Prediction* and the third part *Feature(Pixels) Selection of Single Digit Prediction* is to promote sparsity by finding a model with Pareto Optimality.

2.3 K-Fold Cross Validation

Cross validation is used to test the model's ability to predict new data that was not used in training it. In K-Fold validation, the original dataset is randomly divided to K partitions. Of the K partitions, one partition is retained as testing data, the remaining $K - 1$ partitions are used to train the model. The procedure is iterating for every single partition and the K results can be averaged to produce a single estimation.
K-Fold Cross Validation is used throughout the first part *Mapping Determination* to do the selection of hyperparameter λ.

3 Algorithm Implementation and Result

3.1 Mapping Determination

We use one-hot encoding to transform the training and testing label matrices to the size of $758 * 10$. We fit the training data on Least Square Fit, Ridge and Lasso. To determine the hyperparameters *lambda* for Ridge and Lasso, we implement 5-Fold cross validation on MNIST trainining dataset, and choose λ based on the highest R^2 score. We calculate the accuracy score on testing set of each model to find the best mapping x from image matrix to label matrix

Method	$Optimal\lambda$	Accuracy Score(%)
Least Sqaure Fit		81.17
Ridge	1.0	81.15
Lasso	0.01	81.26

Table 1: Implementation Result of LSF, Ridge and Lasso

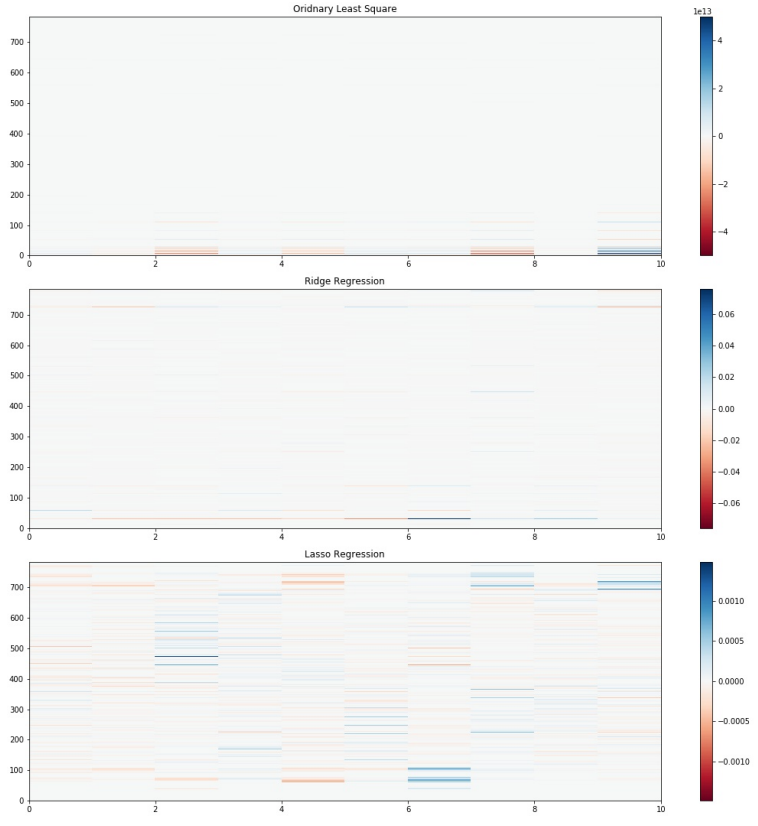

Figure 1: Coefficient Matrix Plot of Each Method

3.2 Feature(Pixels) Selection of Multiple Digit Prediction

Lasso's implementation in the previous part shrinks 3588 pixels to zero.The coefficients of the Lasso loading are all extremely close to zero with a mean(m) of $1.29e^{-}6$ and a standard deviation(sd) of $9.0e^{-}5$.

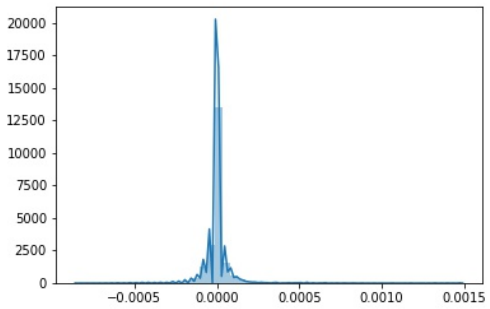

Figure 2: The Distribution of Lasso Coefficients

We create a range of emperical rules, which set the pixels ranging in $[m - i * sd, m + i * sd]$ zeros where $i = 0.1, 0.2, 0.5, 1, 2, 3, 4$, feed the set of modified loading matrices to the testing data and measure the performance based on accuracy score.

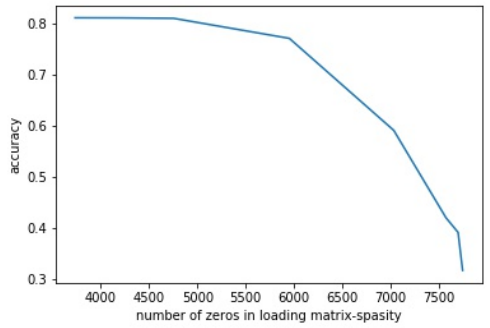

Figure 3: The Relationship between Spasity and Model Performance of Multi-digit Prediction

3.3 Feature(Pixels) Selection of Single Digit Prediction

In this analysis, we transform the training and testing label matrices to 10 sets of $758 * 1$ vector for each digit's prediction.That is to say, if we predict on a certain digit, we make the label of images pertaining to this digit one, and

the others zero. We redo the previous Lasso implementation with 5-Fold cross validation and it turns out that the coefficients of multiple-digit prediction in the second part and single-digit prediction here are exactly the same. However, using the same heuristics as the previous part for each digit does not lead to the same result as we calculate the mean and standard deviation of each digit individually and apply the heuristics to the ten individual digit loadings.

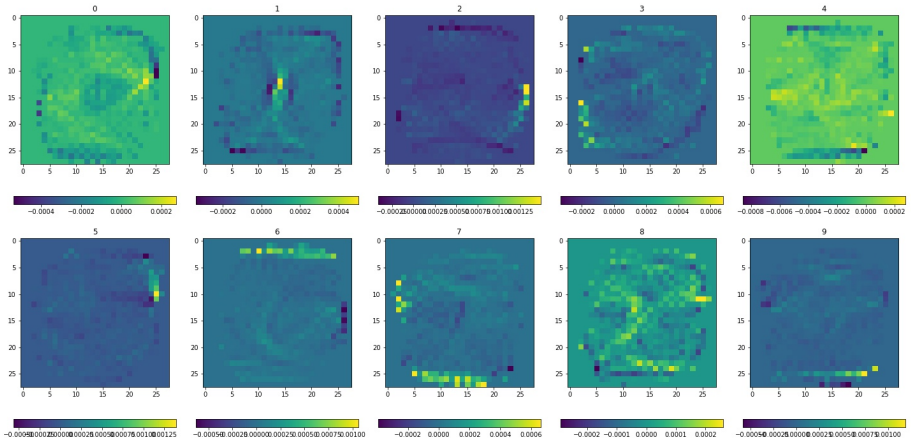

Figure 4: The Loading Matrix of Each Digit in Multi-digit Prediction Lasso Model and Single-digit Prediction Lasso Model

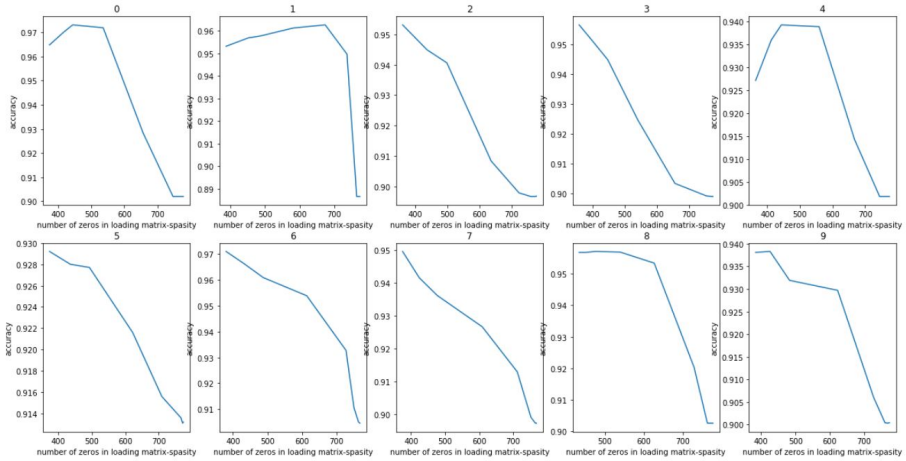

Figure 5: The Relationship between Spasity and Model Performance of Single-digit Prediction

4 Summary and Conclusions

In the first part of the study *Mapping Determination*, from Table 1, we see that all models have very similar performance but Lasso Method has a slightly advantage over the others. It is probably due to the fact that without regularization, the Least Square Fit Method overfits the training set a bit which affects the predictive power on testing set. In addition, the Lasso Method predicts labels better than the Ridge Method because of the nature of our data: the pixels are not hightly correlated to each other.
In the second part *Feature(Pixels) Selection of Multiple Digit Prediction*, Fgure 3 shows that as we shrink more coefficients to zeros, the model performance gets worse. However, Fgure 5 in the third part *Feature(Pixels) Selection of Single Digit Prediction* shows a variety of trends between the sparsity and the model accuracy. For idividual models of digits 0,1,4,9 and 9, promoting sparsity helps with the model performance, while for the other digits, the sparsity and accuracy alway have a negative relationship.
Moreover, single digit model on average gives around 95% accuracy without promoting sparsity, which is around 15% more than the multiple digit model. This demonstrates that Lasso does a better job in distinguishing a digit from another than trying to tell all the digits apart.

A Python Functions Used in this Study

1.sklearn.linear_model.LinearRegression()
LinearRegression fits a linear model with coefficients w = (w1, . . . , wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

2.sklearn.linear_model.RidgeCV()
Ridge regression with built-in cross-validation.

3.sklearn.linear_model.Ridge()
Ridge regression.

4.sklearn.linear_model.Lasso()
Lasso regression.

5.sklearn.model_selection.GridSearchCV()
Exhaustive search over specified parameter values for an estimator. Lasso estimator is used here.

6.numpy.mean()
Calculate the mean of an array. Lasso loading matrix is used here as a part of getting heuristics.

7.numpy.std()
Calculate the standard deviation of an array.Lasso loading matrix is used here as a part of getting heuristics.

8.matplotlib.pyplot.imshow()
Display data as an image.

9.sklearn.metrics.accuracy_score()
Calculate accuracy score given predicted label and true label.

10.keras.utils.to_categorical()
Convert a class vector(integers) to binary class matrix.

11.seaborn.lineplot()
Draw a line plot.

12.matplotlib.pyplot.pcolor()
Display data as an image.

B Original Jupyter Notebook

Please see `https://github.com/gloriatang0325/AMATH-563`