AMATH 563: CLASSIFYING THE MNIST DATASET WITH KERNEL RIDGE REGRESSION

WINNIE LAU

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

1. Introduction

In this report, we will build three types of ridge regression kernels using the linear kernel, polynomial kernel, and Radial basis function (RBF) kernel. These kernels will be applied to the MNIST data set in order to classify the pairs of numbers $(1,9)$, $(3,8)$, $(1,7)$, $(5,2)$. Two different methods were used for preprocessing the data. In the first method, the data was split into a training and test data set with each pair filtered out. Each pair was centered, and had their dimensions reduced using principal component analysis (PCA), and then normalized. The second method normalized the entire training set then had dimensions reduced with PCA before filtering out each pair. The best parameters values for each kernel and digit pair were identified with cross validation. The kernel that had the highest accuracy was the linear and RBF kernel, and the pair of $(1,9)$ had the highest accuracy of bring classified, while $(3,8)$ had the lowest accuracy.

2. Methods

The dataset used in this report is the MNIST dataset, which is a database of handwritten digits, split into a training set and a test set. Each digit consists of an image size 28×28 with corresponding label of an integer from 0 and 9.

To process the data, two different methods were implemented. In the first method, the training data set was centered by the mean of the training set. The dimensions were reduced using principal component analysis (PCA) while preserving 95% of the variance. Afterwards, the data was normalized with StandardScaler. The same scaling and PCA transformation determined from the training set was applied to the test set. The second method normalized the entire training dataset, and then PCA was applied to reduce the dimensions while retaining 95% of the variance. The same transformation was then applied to the test set.

The labels, defined as $y \in \mathbb{R}^N$, contained the number label of each data point. For this report, pairs of numbers were filtered out from the overall dataset to be compared and classified, where one number was given the label -1 and the other was given the label 1.

We define a kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ where given a set of data $X = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$, we design three different kernels:

- Linear Kernel: $K(x,x') = x^T x'$
- Polynomial Kernel: $K(x,x') = (x^T x' + c)^\beta$, for $c \in \mathbb{R}^d$, $\beta \in \mathbb{N}$
- Gaussian/Radial Basis Function (RBF) Kernel: $K(x,x') = \exp(-\frac{1}{2l^2}\|x-x'\|_2^2)$, for $l > 0$.

Date: April 28, 2025.

For the polynomial kernel, $c = [1, \dots, 1]$ and $\beta = 2$ was chosen such that we have a quadratic kernel. Our optimization problem is

$$\begin{aligned} f^* &= \arg\min_{f \in H} \frac{1}{2\sigma^2} \|f(X) - y\|_2^2 + \frac{\lambda}{2} \|f\|_H^2 \\ &= \alpha^{*T} K(\cdot, X) \end{aligned}$$

where H is the reproducing kernel Hilbert space of the positive definite symmetric kernel K defined earlier. Applying the Hilbert space Representation Theorem gives us the second form for f^*. The equation for α^* is given by

$$\begin{aligned} \alpha^* &= \arg\min_{\alpha \in \mathbb{N}} \frac{1}{2\sigma^2} \|K(X, X)\alpha - y\|_2^2 + \frac{\lambda}{2} \alpha^T K(X, X)\alpha \\ &= (K(X, X) + \lambda\sigma^2 I)^{-1} y \end{aligned}$$

derived from finding the gradient and setting it to 0. Let $\gamma = \lambda\sigma^2 > 0$ be another parameters that will be tuned. This yields our expression

$$f^* = K(\cdot, X)^T (K(X, X) + \gamma I)^{-1} y.$$

Because we are classifying given some points $\{(x_i, y_i)\}_{i=1}^N$, $y_i \in \{-1, 1\}$ with $f^*(x) \notin \{-1, 1\}$, we use the label predictor

$$l(X) = \text{sign}(f^*(X)).$$

The accuracy is then calculated as the number of predicted labels that match the known labels, divided by the total number of labels (Equation 1).

$$\text{accuracy} = \frac{\sum_{i=1}^N \mathbb{1}_{[l(x)_i = y_i]}}{N} \tag{1}$$

$$\mathbb{1}_{[l(x)_i = y_i]} = \begin{cases} 1 & \text{if } l(x)_i = y_i \\ 0 & \text{if } l(x)_i \neq y_i \end{cases}$$

To find the best parameters γ for all the kernels and pairs, along with the best l for the RBF kernel, cross validation was ran with `sklearn.model_selection`. The pairs and parameters considered are seen in Table 1.

Variables	Initialization
x_train	$N = 60000$, $d = 734$
y_train	$N = 60000$
x_test	$N = 10000$, $d = 734$
y_test	$N = 10000$
pairs	$[(1, 9), (3, 8), (1, 7), (5, 2)]$
γ	[1e-5, 1e-3, 1e-1]
l	$[0.01, 0.05, 0.1, 0.5]$

Table 1. Initializations for the training sets, test sets, pairs used for classification, and parameters γ and l.

3. Results

In the first method, the dimensions of each data set was reduced from $d = 784$ to $d = \{108, 145, 115, 149\}$ for each pair of numbers, respectively. The optimal γ and l found from cross validation was $\gamma = 1.0e - 05$ and $l = 0.500$ for the RBF kernel, and $\gamma = 0.01$ for the other kernels. The resulting training and test accuracy is reported in Table 2.

We see that the training accuracy was 1, i.e. all correct, for all pairs and kernels except for the linear kernel. However, the linear kernel still had high accuracy on the tests sets, having the highest accuracy for two pairs, with the RBF kernel having the highest accuracy for the other two pairs. The kernel that had the lowest accuracy was the polynomial kernel, especially with the pair $(5, 2)$. The polynomial kernel was defined to be quadratic, so it may not have done the best job to fit to the large data set.

Digit Pair	Kernel	Training Accuracy	Test Accuracy
$(1, 9)$	RBF	1	0.98927239
	Linear	0.99495706	0.9948694
	Polynomial	1	0.99300373
$(3, 8)$	RBF	1	0.96975806
	Linear	0.96344517	0.96219758
	Polynomial	1	0.95362903
$(1, 7)$	RBF	1	0.98566805
	Linear	0.99200431	0.98844198
	Polynomial	1	0.99445215
$(5, 2)$	RBF	1	0.99532225
	Linear	0.97275683	0.97920998
	Polynomial	1	0.81600832

Table 2. Data from the first method of preprocessing. A consistent $\gamma = 1e - 5$ and $l = 0.5$ was used for the RBF kernels for all pairs, and $\gamma = 0.01$ for all other kernels and number pairs. The training and test accuracy is reported as above.

In the second method, the training data was normalized and PCA was applied to the entire training set to reduce dimensions from $d = 734$ to $d = 331$. The ideal parameters for each pair and kernel were found with cross validation, resulting in the training and test accuracy seen in Table 3.

Once again the linear kernel from this method was the only one that did not have 100% accuracy for the training set, but it had the overall highest test accuracy across all pairs. A possible reason could be each kernel overfitting to the training set leading to decreased accuracy for the test set, which the linear kernel was able to avoid. Simmilar to before, the polynomial kernel also had the lowest accuracy, indicating that the kernel built may not have been the most accurate, and other degrees for the kernel could be considered.

The digit pair with consistently low accuracy across all the kernels was $(3, 8)$ with all three kernels having an accuracy aroung 0.95 to 0.97 in the first method, while the second method resulted in two kernels having an accuracy less than 0.9. This aligns with the fact that 3 and 8 have a similar shape, increasing the difficulty for the algorithm to tell them apart. Meanwhile, the highest accuracy was for the pair $(1, 9)$. This also aligns with the fact that 1 and 9 have very different shapes in comparison to each other, likely making it easier to differentiate the numbers.

Digit Pair	Kernel	Parameters	Training Accuracy	Test Accuracy
$(1,9)$	RBF	$\gamma = 1.0e-05$, $l = 0.500$	1	0.97434701
	Linear	$\gamma = 1.0e-01$	0.99590261	0.99347015
	Polynomial	$\gamma = 1.0e-01$	1	0.94402985
$(3,8)$	RBF	$\gamma = 1.0e-05$, $l = 0.500$	1	0.89919355
	Linear	$\gamma = 1.0e-01$	0.9661993	0.96068548
	Polynomial	$\gamma = 1.0e-01$	1	0.88810484
$(1,7)$	RBF	$\gamma = 1.0e-05$, $l = 0.500$	1	0.95423024
	Linear	$\gamma = 1.0e-05$	0.99346506	0.9889043
	Polynomial	$\gamma = 1.0e-01$	1	0.92094313
$(5,2)$	RBF	$\gamma = 1.0e-05$, $l = 0.500$	1	0.87110187
	Linear	$\gamma = 1.0e-01$	0.98092978	0.972920998
	Polynomial	$\gamma = 1.0e-05$	1	0.97401247

Table 3. Data from the second method of preprocessing. Each pair and kernel had cross validation run to find the optimal γ and l leading to the resulting training and test accuracy.

4. Summary and Conclusions

The digit pair with the highest accuracy was $(1,9)$, while the digit pair with the lowest accuracy was $(3,8)$. The kernel that had the highest overall test accuracy was between the linear and RBF kernel, while the polynomial kernel had the lowest accuracy.

Possible future directions includes exploring more values for γ and l to see how well the accuracy can be improved. Moreover, it would be interesting to test the polynomial kernel further with varying values of c and degree β. We could also consider testing the kernels with other pairs of the numbers that were not considered in this report, along with using other kernels such as the Sigmoid kernel. An interesting consideration could also be figuring out how to classifying three or more digits instead of just pairs of digits.

Acknowledgements

The author is thankful to Prof. Bamdad Hosseini for useful discussions about the reproducing kernel Hilbert spaces and kernel ridge regression. Thank you also to David Kieke and Christina Nicolaides for insight on pre-processing the data and implementing the kernels in Python.