AMATH 563 - HOMEWORK 2 (COMPUTATIONAL REPORT)

NATE WHYBRA

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

1. Introduction

Given the MNIST data set composed of 28×28 images of handwritten digits between 0 and 9, we would like to create a model to classify between different pairs of digits. Our method of choice is Kernel Ridge Regression (KRR), and we test various models which utilize 3 different kernel functions (linear, quintic, RBF) over various hyper-parameters (regularization, and kernel parameters). To increase processing speed, we employ Principal Component Analysis (PCA) to reduce the dimensions of our data prior to training our KRR models.

2. Methods

Our training set consists of $n_1 = 60,000$, 28×28 images, which have been flattened into row vectors of length 784 and stacked into a $60,000 \times 784$ real matrix X_1 with corresponding training labels $Y_1 \in \mathbb{R}^{n_1}$. Our test set is represented in the same manner with $n_2 = 10,000$ images by a 1000×784 real matrix X_2 with corresponding test labels $Y_2 \in \mathbb{R}^{n_2}$. We first perform principal component analysis (PCA) [**1**] on X_1 by subtracting the column-wise mean from each column in X_1 and then subtracting those same means from each column in X_2, computing its SVD so that $X_1 = USV^T$, and computing the number k of singular values needed to explain 95% of the data variance as:

$$k = \arg\min_{k \geq 1} \left(\frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{784} \sigma_i^2} \geq 0.95 \right)$$

We then redefine $X_1 := X_1 V[0:k,:]^T \in \mathbb{R}^{(n_1,k)}$ and $X_2 := X_2 V[0:k,:]^T \in \mathbb{R}^{(n_2,k)}$ (Python slicing notation) so that 95% of the original data variance is captured in the new matrices which now have smaller dimensions [**1**]. By compressing the data in this way, training our KRR models will take much less computational time and storage. After performing PCA, to make classifiers between pairs of digits (i, j), the data corresponding to the digits are selected from X_1, Y_1, X_2, Y_2 and standardized so that each column in the training data has mean 0 and variance 1. These same means and variances that were used to standardize the training data are then used to standardize the test data column-wise.

Date: 4/28/2025.

Now for Kernel Ridge Regression (KRR), suppose we have some $d-$dimensional data vectors $\{x_1, \cdots, x_n\} = X$, a positive definite symmetric kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, and H is the Reproducing Kernel Hilbert Space (RKHS) [**2**] generated by K, then traditional regression problems of the form:

$$f^* = \arg\min_{f \in H} \left(\frac{1}{2\sigma^2} \|f(X) - Y\|_2^2 + \frac{\lambda}{2} \|f\|_H^2 \right)$$

Can be uniquely solved [**2**] as:

$$f^*(x) = K(x, X)^T (K(X, X) + \gamma I_d)^{-1} Y$$

Where I_d is the $d \times d$ identity matrix and $\gamma = \lambda \delta^2$ is our regularization parameter. The function f^* will take values in $\mathbb{R}$, so to make a classifier with binary outputs, we say our classifier is:

$$C(x) = \begin{cases} -1 & \text{if } f^*(x) < 0 \quad \text{(Corresponding to Digit 1)} \\ 1 & \text{if } f^*(x) \geq 0 \quad \text{(Corresponding to Digit 2)} \end{cases}$$

We will be testing our classifiers on 3 different kernel functions:

$$K_L(x, y) = x^T y \quad \text{(Linear Kernel)}$$

$$K_Q(x, y) = (1 + x^T y)^5 \quad \text{(Quintic Kernel)}$$

$$K_{RBF}(x, y) = \exp\left(-\frac{\|x - y\|_2^2}{2l^2} \right) \quad \text{(RBF Kernel)}$$

Finally, we will be experimenting with the regularization parameter γ for all kernels, and also with the length scale parameter l for the RBF kernel, with values of $\gamma, l \in \{0.01, 0.1, 1, 10, 100, 1000\}$.

3. Results

After performing the PCA, the number of singular values needed to capture 95% of the data variance was $k = 330$. We then trained 3 classifiers (1 for each kernel function) over the various values of γ and l. Below, we supply the values of γ, l that were the "best" in the sense they had the highest testing accuracy. Here, the accuracy is defined as the number of correct classifications divided by the total number of data points in the training and test sets respectively.

Parameters	Linear	Quintic	RBF
γ	1	1000	0.01
l	-	-	10
Train Accuracy (%)	99.56	99.76	99.99
Test Accuracy (%)	99.49	99.44	99.53

TABLE 1. Best-performing parameters and metrics for each kernel when classifying between digits **1 and 9**.

Parameters	Linear	Quintic	RBF
γ	10	100	0.01
l	-	-	10
Train Accuracy (%)	96.49	99.90	98.72
Test Accuracy (%)	96.02	97.53	98.03

TABLE 2. Best-performing parameters and metrics for each kernel when classifying between digits **3 and 8**.

Parameters	Linear	Quintic	RBF
γ	100	100	0.01
l	-	-	10
Train Accuracy (%)	99.30	99.11	98.15
Test Accuracy (%)	99.03	98.15	98.03

TABLE 3. Best-performing parameters and metrics for each kernel when classifying between digits **1 and 7**.

Parameters	Linear	Quintic	RBF
γ	10	1000	0.01
l	-	-	100
Train Accuracy (%)	98.07	99.93	99.30
Test Accuracy (%)	98.01	98.54	98.96

TABLE 4. Best-performing parameters and metrics for each kernel when classifying between digits **5 and 2**.

4. Summary and Conclusions

Classifying between 2 digits in the MNIST dataset seems to be a relatively "easy" task, as we were able to see great generalization to the test data for every kernel function tried and with various sets of hyper-parameters γ and l, for every pair of digits we made a classifier for. The quintic and RBF kernels performed the most favorably the most often. If we were to repeat this experiment again, it would be interesting to see what happens if we performed the PCA independently for every pair of digits. In this manner we would be keeping the data that describes 95% of the total variance for every pair of digits we made a classifier for as opposed to just once for the entire dataset, and we hypothesize we could see even higher test accuracies.

Acknowledgments

The author would like to thank James Hazeldon for reminding us we could use PyTorch and CUDA to significantly speed up the processing time of our experiments.

References

[1] W. contributors. Principal component analysis. `https://en.wikipedia.org/wiki/Principal_component_analysis`, 2025. [Online; accessed 15-April-2025].

[2] B. Hosseni. Lecture notes for AMATH 563: Lecture 10. Lecture notes distributed in class, 2025. University of Washington, unpublished lecture notes.

[1] [2]