AMATH 563: COMPUTATIONAL REPORT 2

NGHI NGUYEN

Applied Math Department, University of Washington, Seattle, WA
nghiiuu@uw.edu

1. Introduction

In this report, the Kernel Ridge Regression is used to classify a given MNIST data set into different pair of digits. The data sets will be preprocessed with the Principal Component Analysis (PCA), and both centralization and normalization on the mean and variance of the data. The kernel used in this report are Linear, Polynomial, and the Radial Basis Function (RBF). For each kernel, Grid Crossing and Cross Validation fine tunes the combinations of hyper-parameters to produce the lowest error for the training data set, subsequently the combinations will be applied onto the testing data.

2. Methods

Kernel Ridge Regression is a least squares regression with a regularization parameter λ that is implemented to approximate the objective function $f(x)$ with a positive semidefinite (PDS) kernel. Given a space of data $X = \{x_1, .., x_n\} \subset \mathbb{R}^n$ where each $\{x_i\}_{i=1}^n$ are called features and a vector $y = \{y_1, .., y_n\} \subset \mathbb{R}^n$ where $\{y_i\}_{i=1}^n$ are called labels, we will implement the Kernel Ridge Regression to find $f^x)$ such that $f^(x_i)$ classifies the specific label of x_i for all $i = 1, .., n$.
The label vector y and classification function $f(x)$ has the following equality,

$$y = f(x) + \epsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma I)$$

where noise, denoted by ϵ, has a mean of 0 and standard deviation of σ^2. To approximate $f(x)$, we have the following KRR implementation,

$$f^* = \text{argmin}_{f \in H} \frac{1}{\sigma^2} ||f(x) - y||_2^2 + \frac{\lambda}{2} ||f||_H^2$$

where H is a Reproducing Kernel Hilbert Space (RKHS), imposing f^* to lie in an inner product space. By Riesz Representation theorem and Representor theorem, we can express as a a linear combination,

$$f^* = \sum_{j=1}^{n} \alpha_j^* K(\cdot, x_j)$$

where $\alpha^* = \text{argmin}_{\alpha \in \mathbb{R}^n} \frac{1}{\sigma^2} ||K(X,X)\alpha - y||_2^2 + \frac{\lambda}{2} \alpha^T K(X,X)\alpha$. Denote $J(\alpha) = \frac{1}{\sigma^2} ||K(X,X)\alpha - y||_2^2 + \frac{\lambda}{2} \alpha^T K(X,X)\alpha$. Since $J(\alpha)$ is convex, the minimizer α^* can be computed by taking the gradient $\nabla f(\alpha^*) = 0$. The minimizer α and the objective function is $\alpha^* = (K(X,X) + \lambda\sigma^2 I)^{-1} y$ and $f^* = K(\cdot, X)^T (K(X,X) + \lambda\sigma^2 I)^{-1} y$. The Cholesky Decomposition and the function scipy.linalg.cho_solve on Python will be applied to$K(X,X) + \lambda\sigma^2 I$ to find its inverse.

Date: April 29, 2025.

To distinguish the labels, the following **kernels** will be used:

Denote the kernel $K : X \times X \to \mathbb{R}$.

(1) **Linear Kernel** $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, $K(x, x') = x^T x'$ for $x, x' \in \mathbb{R}^n \to \mathbb{R}$
(2) **Polynomial Kernel** $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, $K(x, x') = (x^T x' + c)^d$ for $x, x' \in \mathbb{R}^n \to \mathbb{R}$ and $c, d \in \mathbb{R}$.
(3) **Radial Basis Function (RBF)** $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, $K(x, x') = \exp(-\frac{1}{2l^2}||x - x'||_2^2)$ for $x, x' \in \mathbb{R}^n \to \mathbb{R}$ and the length scale $l > 0$

The given training data set and testing data set, denoted as X_{train} and X_{test}, is a MNIST data set comprised of 60000 training labels and 10000 testing labels. In this report, for each pair if digits (label), where (y_i, y_j), $y_i, y_j \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, y_i will be assign to 1 and y_j will be assign to -1.

Before the algorithm implementation for KRR, the MNIST dataset will be **pre-processed** with three of the following steps: **centering data, Principal Component Analysis (PCA),** and **normalizing data**.

First, the training and testing data will be centered such that the mean of each data set will be zero. Second, PCA is a dimensionality reduction method to ensure that the reduced data set as seen in 1 has relevant labels for classification; here, PCA will preserve 95% of the variance. Note that the centering step is imposed before PCA to ensure PCA only considers the direction of the variance. The objective function will assign each value in the reduced data set, now composed of the relevant pairs of digits $(x_i, x_j) = (y_i, y_j)$ to 1 or -1. The third step is to normalize the extracted data from the training and testing data. The mean of extracted data will be re-centered to zero and will be divided by the standard deviation, thus imposing a variance of one and a mean of zero.

Digit Pair	**PCA Components**
(1, 9)	108
(3, 8)	145
(1, 7)	115
(5, 2)	149

TABLE 1. Number of PCA components to retain 95% variance for each digit pair

For the KRR implementation, we have the follow hyper-parameters:

(1) The scaling parameter $\gamma = \lambda\sigma^2$ for f^* and α^*. Note that the pre-processing stage, the data sets were normalized to have a variance of one. Thus, the parameter $\gamma = \lambda$
(2) Degree parameter d for the polynomial kernel
(3) The scaling parameter $\xi = \frac{1}{2l^2}$ for the RBF kernel

After both training and testing sets have been pre-processed and the hyper-parameters have been determined, **Cross Validation** and **Grid Search** will be applied on the KRR. Grid Search finds the best combinations between the hyper-parameters for each kernel. Cross Combinations splits the data set into a folds where each folds have a uniform number of data; all of the combinations from the Grid Search will be applied on each fold as seen in 2 and the errors each unique combinations will be averaged across the folds for each kernel.

Table 2. Various Kernels, Hyper-Parameters and Fits

Kernels	α	γ	degree	Folds $\times$ Combinations
Linear	0.01 0.001 0.0001			3×3
Polynomial	0.01 0.001 0.0001		2 3 4	3×9
RBF	0.01 0.001 0.0001	0.01 0.001 0.0001		3×9

3. Results

The tables below show each pair of digits, each with the three different kernels and the best hyperparameter combinations from Grid Search and Cross Validation that give the smallest Error Train. As seen in table 3, 4, and 6, the Polynomial and RBF kernels give the smallest errors for the training data across the the pairs (1,9), (1,7), and (5,2). It is also consistent across all of the pairs that the Linear kernel gives the highest error for the data set. The better performance of Polynomial and RBF kernels could be caused by how the Grid Search and Cross Validation performs a total of 27 fits (3 folds by 9 combinations) whereas the Linear kernel only has a total of 9 fits (3 folds by 3 combinations). The higher number of fits provide may higher flexibility and adaptation to the kernels and hyperparameter.

In addition, the best scaling parameter $\gamma = \lambda$ for the objective function f^* and the optimal coefficient α^* is 0.01 in 3, 5, and 6, in other words the biggest regularization parameter performs better than any smaller values for most case. This could be the case as λ prevent the overfitting of the predictions to the labels and a higher regularization parameter impose less constrains for overfitting; note that since the labels are only denoted as 1 and -1, the model is not as complex and overfitting is not a big issue.

The errors for the testing data is the smallest for the RBF kernel for all of the pairs while the Linear kernel performs the worst for 3 pairs in 3, 4, 5, and the Polynomial kernel performs the worst in 6. Note that the Polynomial kernel gives relatively small errors for both training data and testing data for the pairs (1,9), (3,8), and (1,7); there is a correlation as the polynomial degree is all $d = 3$. However, when the Polynomial kernel gave the biggest error for the testing data for the pair (5,2), it has a degree of $d = 2$. In addition, the RBF kernel has the same best scaling parameters $\xi = .001$ for all of the pairs; the kernel shows consistency for both scaling parameters and the smallest errors for both testing and training sets.

Table 3. Results for Digit Pair (1,9) with PCA = 108

Group	**Detail**	γ	d degree	ξ	$\text{Error}_{\textbf{train}}$	$\text{Error}_{\textbf{test}}$
(1,9)	Linear	0.01	n/a	n/a	0.00504	0.00513
	Polynomial	.01	3	n/a	0.0	0.00373
	RBF	.01	n/a	0.001	0.0	0.00326

Table 4. Results for Digit Pair (3,8) with PCA = 145

Group	**Kernel**	γ	d degree	ξ	Error$_{\text{train}}$	Error$_{\text{test}}$
(3,8)	Linear	0.01	n/a	n/a	0.03655	0.03780
	Polynomial	0.0001	3	n/a	0.0	0.00756
	RBF	0.01	n/a	0.001	0.0	0.00353

Table 5. Results for Digit Pair (1,7) with PCA = 115

Group	**Kernel**	γ	d degree	ξ	Error$_{\text{train}}$	Error$_{\text{test}}$
(1,7)	Linear	0.01	n/a	n/a	0.00799	0.01156
	Polynomial	0.01	3	n/a	0.0	0.00786
	RBF	0.01	n/a	0.001	0.00008	0.00324

Table 6. Results for Digit Pair (5,2) with PCA = 149

Group	**Kernel**	γ	d degree	ξ	Error$_{\text{train}}$	Error$_{\text{test}}$
(5,2)	Linear	0.01	n/a	n/a	0.02724	0.02079
	Polynomial	0.01	2	n/a	0.0	0.23441
	RBF	0.01	n/a	0.001	0.0	0.00104

4. Summary and Conclusions

In summary, the Kernel Ridge Regression (KRR) was implemented to classify the on a MNIST training and testing data sets. Each data set was pre-processed with the Principal Component Analysis (PCA), centering, and normalizing the data. The three kernels that were used include the Linear, Polynomial, and Radial Basis Function (RBF) kernel. For each kernel, a Grid Search and Cross validations were performed on different combinations and folds of the kernels' hyper-parameters. The results shows that the RBF kernel and a high regularization parameter give the smallest errors on the testing and training data sets for every pair, while the Linear kernel gave the highest errors for most pairs. To expand this report, we can try different values for the constant in the Polynomial Kernel and include bigger regularization values.

Acknowledgements

I would like to thank Emily, Tomas, Christina, Howard, and David for a helpful discussion.