K-Fold Cross-Validation & Regularization

SENG 474 Assignment 2
Austin Bassett
V00905244

February 2021

1 Introduction

The Fashion-MNIST (MNIST) dataset classifies clothing/apparel items into one of ten classes. Due to the size of this dataset a smaller subset was used for the experiments in this report. The chosen subset only used class 5 and 7 (Sneakers and Sandals respectively), and contained 12,000 training examples and 2,000 testing examples. This dataset was modified slightly to better fit binary classification. All the inputs were divided by 255 to ensure their values lied in the range of [0,1], and reassigning class 5 and 7 to 0 and 1 respectively. The goal of these experiments is to see how regularization affects machine learning models.

The models to be examined are logistic regression (LR), linear support vector machines (LSVM), and Gaussian support vector machines (GSVM). Due to the support vector machines requiring long training times only 6,000 training examples were used in all the experiments. For the initial experiments a range of regularization values were calculated and the resulting errors were recorded and graphed. Next, I used a k-fold cross-validation method to find the optimal regularization value for LR and LSVM. Then I reran the LR and LSVM using their newly optimized regularization value recording the resulting errors for comparison. Finally, GSVM used the k-fold cross-validation method to find the optimal regularization value for a given gamma.

2 Logistic Regression

The logistic regression models used the L2 regularization and the value was being passed as a parameter. To produce a range of results I calculated a different regularization value for each model.

Using 10 calculated regularization values I trained 10 different models, and the resulting errors were graphed against the regularization value. The following formula was used to calculate regularization values.

$$\alpha^X * C_0 \tag{1}$$

The values α and C_0 were fixed while X was incremented with each model starting at 0 and ending at 9. By calculating regularization values this way a greater range of results could be produced. This was ideal because I wanted to see how low and high values of regularization affected logistic regression models.

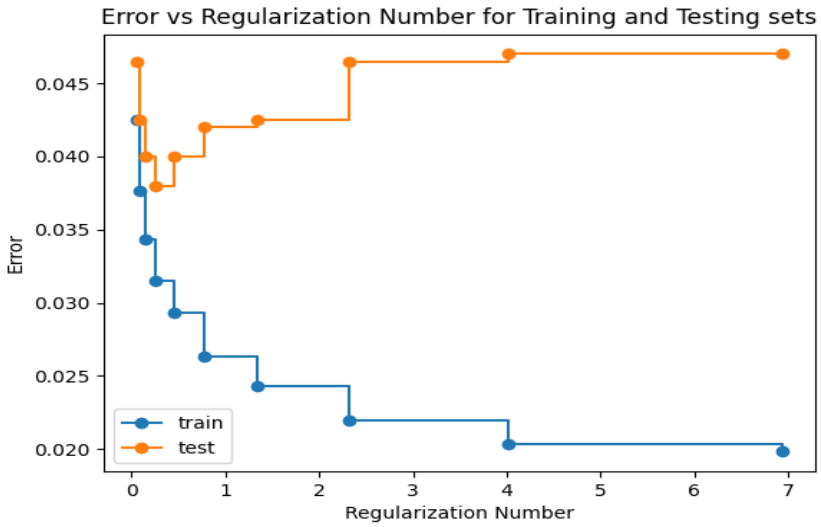

Figure 1: Unoptimized Logistic Regression Error Results

Using 0.05 for C_0 and 1.73 for α I trained 10 models using the regularization calculation (see Equation 1). The errors were recorded for each model along with its regularization value. I then graphed the errors against the regularization values (see Figure 1). The graph lines make it appear that the models are overfitting because the lines start diverging around the regularization value 0.5. However, looking at the actual error values at the closest point (Training: $\approx$4.25%; Testing: $\approx$4.6%) and the furthest point (Training: 2%; Testing: $\approx$4.6%) they only ever differ by a maximum of 2.6%.

Since the training error are small and relatively close to each other we can say this method produces best fit models.

Unoptimized (No KFold)	Error	Regularization Value
Training	0.019833333333333335	6.940406893818192
Testing	0.038	0.25888585

Table 1: Best Unoptimized Logistic Regression Errors.

For regularization the best value is a single value that produces low testing and training error while keeping the errors relatively close. The best results for training and testing error along with their regularization value used is recorded in Table 1. From there I calculated a 95% confidence interval for the testing error using the following formula.

$$Z * \sqrt{\frac{TestingError * (1 - TestingError)}{2,000}} \tag{2}$$

Using Z as 1.96, I calculated the following confidence interval 3.8% $\pm$ 0.8%. Meaning the true testing error is between 3.0% and 4.6%. This interval will be used for comparison to an optimized logistic regression testing error later. Finally, we see that the best regularization value was not found because, while the errors are relatively close, two different regularization values were needed.

3 Linear Support Vector machine

The next experiment I conducted was a support vector machine using a linear kernel (LSVM). I conducted this experiment exactly like I did in logistic regression, only I set C_0 to 0.002 and α to 3.24 for calculating a regularization value (see Equation 1).

Looking at the graphed results (see Figure 2) we see what appears to be models that were overfitting. However, upon closer inspection we see that the models were actually underfitting; because the errors are between 49% and 57%. Additionally, it appears that, given higher regularization values, the models would start overfitting the data. Which makes sense because higher values means less regularization occurs (overfitting).

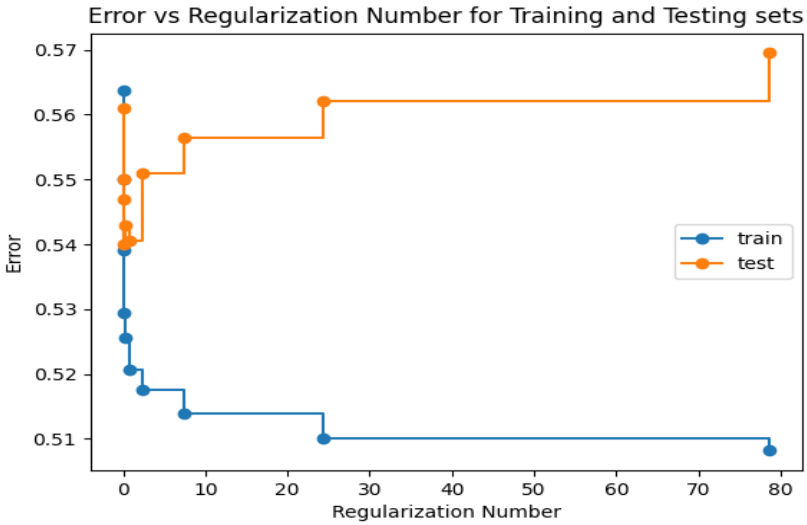

Figure 2: Unoptimized Linear Support Vector Machine Results.

The best results are recorded in Table 2 and we can see, just like with logistic regression, the best regularization value was not found. Because while errors are extremely close, two vastly different regularization values were needed. I also calculated a 95% confidence interval (see Equation 2) for the best training error which I found to be $54\% \pm 2.18\%$, this puts the true testing error between 51.82% and 56.18%. This interval will be used as a comparison for an optimized LSVM testing error later.

Unoptimized (No KFold)	Error	Regularization Value
Training	0.5081666666666667	78.69281615059312
Testing	0.54	0.068024448

Table 2: Best Unoptimized Linear Support Vector Machine Errors.

4 K-fold Cross-validation

Using k-fold cross-validation (KFold) we will find the optimal regularization value for a given model. KFold works by splitting the data into K folds (sections) leaving one fold out for testing while the other K-1 folds are used for training. This means there will be K iterations for each regularization value, producing a K x K matrix

with the main diagonal being the testing folds. The test scores of each iteration were recorded and the average was calculated. The higher the average the better the regularization value was. Both logistic regression (LR) and linear support vector machine (LSVM) used the same calculations and values from the unoptimized experiments (see Equation 1). Lastly, I set K to be 8 in every experiment I ran.

The LR results for both the unoptimized (no KFold) and optimized (KFold) are recorded in Table 3. Finding the best single regularization value did increase the training error by about 0.43% and the testing error by 0.5%. A confidence interval for the optimized testing error was calculated to be $4.3\% \pm 0.89\%$. Meaning the true testing error for optimized LR lies between 3.41% and 5.19%. Compared to the unoptimized true testing error which lies between 3.0% and 4.6%. Since there is overlap between the unoptimized and optimized true error we can not say for certain which hypothesis is better.

Unoptimized (No KFold)	Error	Regularization Value
Training	0.01983333333333335	6.940406893818192
Testing	0.038	0.25888585
Optimized (KFold)	Error	Regularization Value
Training	0.024166666666666666	1.4361450195312502
Testing	0.043	1.4361450195312502

Table 3: Best Unoptimized & Optimized Logistic Regression Errors.

The LSVM results for both the unoptimized (no KFold) and optimized (KFold) are recorded in Table 4. Finding the best single regularization value only increased the training error by about 2.13%; while the testing error was untouched. A confidence interval for the optimized testing error was calculated to be $54\% \pm 6.80\%$. Meaning the true testing error for optimized LSVM lies between 47.2% and 60.8%. Compared to the unoptimized true testing error which lies between 51.82% and 56.18%. Again, since there is overlap between the unoptimized and optimized true error we can not say for certain which hypothesis is better.

Unoptimized (No KFold)	Error	Regularization Value
Training	0.5081666666666667	78.69281615059312
Testing	0.54	0.068024448
Unoptimized (No KFold)	Error	Regularization Value
Training	0.5295	0.068024448
Testing	0.54	0.068024448

Table 4: Best Unoptimized & Optimized Linear Support Vector Machine Errors.

5 Gaussian Support Vector Machine

The final experiment used another support vector machine with a gaussian kernel (GSVM) instead of a linear kernel. GSVM was similar to LSVM with regards to using a regularization value, however, GSVM also used a gamma value. This gamma value determines how far the influence of a single training example reaches, low values meaning far and high values meaning close[1]. During the GSVM experiments KFold was used to find the optimal regularization value for a given gamma value. Due to GSVM requiring a considerable amount of time to train only the optimized version was done.

The regularization value was calculated exactly the same way as LR and LSVM with C_0 as 0.002 and α as 3.24 (see Equation 1). For the gamma value calculation I derived the following formula.

$$\frac{1}{\beta} \tag{3}$$

The 10 different β values were chosen from a logarithmically spaced grid with the range (199.5, 2511.9). During initial code debugging I discovered gamma values above 100 were producing graphs with 1.00 for training and testing error. Thus, I derived this formula for the gamma value calculation which produced better results. Now, I trained and tested GSVMs using the gamma with its optimized regularization value. The resulting errors were graphed against the gamma value (see Figure 3). Inspecting the graph we can see that the models were underfitting. This is because the range of error is from 50% to 53.5%. However, it does appear that the training error and testing error might reduce enough to produce best fit models with larger gamma values.

The best training and testing error along with the gamma value and regularization value were recorded in Table 5. We can see this

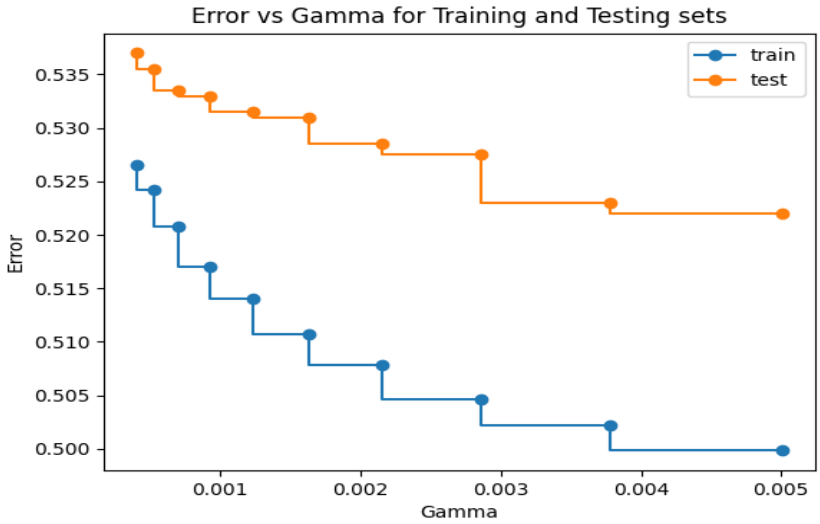

Figure 3: Optimized Gaussian Support Vector Machine Results.

regularization value was indeed optimal because the training and testing error were relatively close and low. The calculated confidence interval for the best testing error was 52.2% ± 2.19%, meaning the true error lies between 50.01% and 54.39%. Compared to LSVM's true error which lies between 51.82% and 56.18%. With the confidence intervals overlapping we can not say for certaint which model is better. However, given both methods are producing underfitting models a better method for calculating regularization values is needed.

6 Conclusion

After analyzing the results the best machine learning model for the modified dataset was logistic regression. This is because the logistic regression models achieved low training and testing errors. Compared to the support vector machines, which achieved high training and testing errors relative to logistic regression. Meaning logistic regression models were best fit models while support vector machines were under-fit models. However, both results are ideal because the purpose of using regularization is to avoid overfitting. In conclusion, using regularization can help prevent models from overfitting to training data and produce a generalized hypothesis.

7 References

[1] *"RBF SVM parameters"*, Scikit-learn, Accessed on: Feb 27, 2021.[Online]. Available: `https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html`