E3-23 Machine Learning Coursework

Ricardo Vera CID: 01061547
Imperial College London

1. Pledge

I, Ricardo Vera, pledge that this assignment is completely my own work, and that I did not take, borrow or steal work from any other person, and that I did not allow any other person to use, have, borrow or steal portions of my work. I understand that if I violate this honesty pledge, I am subject to disciplinary action pursuant to the appropriate sections of Imperial College London.

2. Problem

The aim of this report is to address the best machine learning algorithm to solve the problem concerning Wine Quality Classification. Given a set of 11 of the wine's parameters such as *pH* or *fixed acidity*, the algorithm should be able to give the wine a rating from 1 to 10. Wine Quality Classification is restricted to the evaluation of human wine experts, thus an algorithm that could simulate this decision would provide a method resistant to human subjective behavior, providing a more objective classification. The data used for this report and project will be datasets related to red and white variants of the Portuguese "Vinho Verde" wine [1].

It is worth to note the relevant issues regarding this data. The dataset is not completely reliable to perform a proper model as the training will be subject to the flaws in the dataset. The data set is composed of 4898 white wines and 1599 red wines, however these are greatly decompensated, with a huge proportion of fives and sixes and considerably fewer examples of lower quality (3 and 4) or higher quality values (7 to 9). The dataset in fact has no wines with quality 1, 2 or 10, which makes the algorithm unlikely to classify any input into these classes, which is unrealistic.

Loss Function

The objective of the project is to be able to match the rating of the wine from 1 to 10, given the wine's parameters. As a result two different loss functions are suitable for the analysis, classification error and mean squared error.

Mean squared error provides the most suitable loss function from the point of view of the problem. By measuring the difference between the prediction and the actual rating, the MSE provides a way of measuring how far are the estimates from the real value, as large differences will be punished in a squared manner. As the objective is to successfully simulate a wine expert's decision, small differences in estimation (such as rating a wine as 6 when it is really 7) are not a huge concern, however predicting a rating very differently to its real value (rating a wine as 3 instead of a 9) is a noticeable issue.

On the other hand, the classification error provides a measure of how many different wines did the algorithm classify incorrectly, as a percentage. This loss function does not punish considerable differences and instead provides a measure on how many mistakes does the algorithm make on average (x every 100).All errors shown in this report are an average of 5 executions.

3. Baseline Predictors

Aiming to give each wine a rating from 1 to 10, a regression implementation is suitable to minimize the error. In order to compare with the real value the output of the algorithm can be rounded to give a suitable value for comparison.

The datasets are divided arbitrarily as of 30% for testing and 70% for training.

Linear Regression

The baseline and simplest algorithm implemented is the Linear Regression algorithm, which minimizes the squared error by setting $w_{opt} = (X^TX)^{-1}X^Ty$, with X representing the wine parameters and y the corresponding wine rating.

The linear regression algorithm has a closed form solution provided X^TX is invertible, which added onto its simplicity makes it the ideal baseline predictor for comparison and improvement.

	Classification Error		Mean Squared Error	
Dataset	$\hat{R}_n(w)$	$R(w)$	$\hat{R}_n(w)$	$R(w)$
Red Wine	40.21%	41.88%	0.4933	0.5167
White Wine	48.53%	47.92%	0.6401	0.6630
Combined	46.83%	47.05%	0.6203	0.6593

Table 1: Error measures for Linear Regression

As can be seen from Table 1, the Linear algorithm performs best in the Red Wines dataset, but maintains reasonable error values in both classification and mean squared error.

Ridge Regression

A similar algorithm implemented for the project is the Ridge Regression algorithm. This is a modified version of the Linear Regression, but restricted to $w^Tw \leq C$, hence introducing a penalty function which penalizes large $||w||^2$. Ridge Regression is used to avoid over-fitting, thus obtaining a more appropriate estimate of the feature importance, the weights are set to $w_{opt} = (X^T X - \lambda I)^{-1} X^T y$. λ is not defined and so is set empirically, in this case via cross-validation.

A cross-validation of 10-fold is performed in order to obtain the value of λ that minimizes the error, from Figure 1 it can be observed how the value of lambda that minimizes the training error is 180.19. It is worth to note, even though two different loss functions are being used, λ coincides for both, hence reinforcing its validity as the λ that minimizes error.

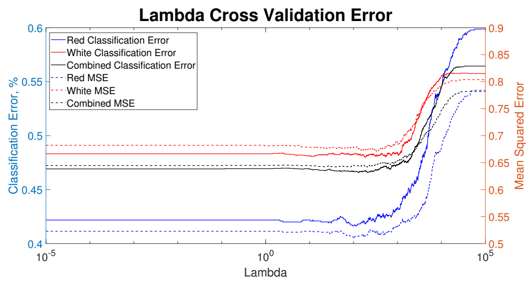

Figure 1: Training Error for different values of λ

After selecting the optimal λ, this value is used to train the model with all the training dataset (70% of all data) to obtain the optimum weights.

	Classification Error		Mean Squared Error	
Dataset	$\hat{R}_n(w)$	$R(w)$	$\hat{R}_n(w)$	$R(w)$
Red Wine	41.22%	39.83%	0.4209	0.4276
White Wine	47.73%	49.00%	0.5741	0.5621
Combined	46.04%	47.84%	0.5388	0.5537

Table 2: Error measures for Ridge Regression

Table 2 shows that like Linear Regression, Ridge Regression performs better in the Red Wines dataset than in the rest. It is clear how this algorithm has lower test error $R(w)$, which reinforces the over-fitting suggestion in the linear approach, which Ridge Regression appropriately prevents. The algorithm also performs better in the MSE meaning the deviation from the actual value is less and hence the prediction is better.

4. Advanced Algorithms

Until now the algorithms used to solve the problem have been basic algorithms based on the objective of minimizing the error by providing an appropriate fit. The wine quality problem is a more heuristic problem and so more complex algorithms may be able to detect different trends in the data. Three different complex algorithms will be used: Multi-Class Perceptron, Multi-Class Support Vector Machine and Neural Network.

The Multi-Class Perceptron operates on the principle of one-vs-one (OVO) or one-vs-all (OVA), in order to be able to classify different classes. This implementation allows the problem to be reduced to a multi-dimension binary classification problem over the different classes, which hence avoids the problem with rounding in the linear algorithms (where a 5.55 will be classified as a 6 even though its distance to 5 is not significantly more).

The one-vs-one strategy implies the one-vs-one ensemble will have $\frac{n(n-1)}{2}$ binary classifiers, in order to classify the data into the different n classes, and then classification is done by majority voting. Suppose you have 3 classes A, B and C, therefore having 3 binary classifiers. The classifiers will segregate A from B, A from C and B from C. If a data point x is needed to classify the ensemble will give something like (A, B, B), meaning A classifies better than C, but B classifies better than A, and thus the final classification is B.[2]

One-vs-all on the other hand will develop one classifier per class, hence producing n classifiers. The strategy implies treating each class in an isolated way, and only considering *is of that class* or *isn't from that class* for every class, and thus developing individual classifiers for each class. A visual representation is shown in Figure 7. [3]

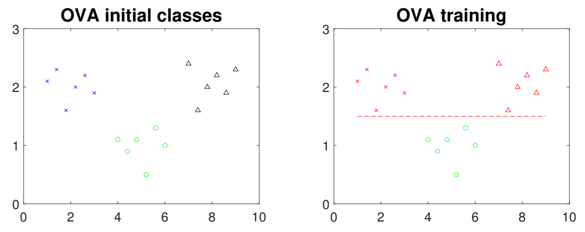

Figure 2: One-vs-all approach

One-vs-all often leads to imbalanced datasets, whilst the complexity of one-vs-one is larger as more classifiers are needed, thus a trade-off is needed.

The Multi-Class Support Vector Machine (SVM) operates on the same principle as the Multi-Class Perceptron, also with one-vs-one or one-vs-all, but with the objective of maximizing the margin. The Multi-Class provides a more efficient algorithm as it only considers the points delimiting the margin and not all points, therefore providing a better result. Finally, the possibility of using Kernel Functions in the SVM provides a better result as the problem can be mapped to higher dimensions for classification.

The Neural Network provides a different approach. Neural Networks are very good at pattern recognition problems due to their capability to generalize and to react to abrupt inputs/patterns, thus being able to detect the heuristic patterns in wine quality classification and replicate wine experts' reasoning. Neurons in the network take many weighed inputs and produce an output, therefore the packing of different neurons in subsequent layers provides an ability to learn complex patterns, outside the scope of other learning algorithms. The main drawback on Neural Networks however, is the tendency to over-fit the data, thus producing increased test error.

Implementation

Again, the datasets are randomly divided, the division is such that 30% is used for testing data and 70% for training data.

Multi Class Perceptron

As mentioned above the multi-class Perceptron will divide the problem into different binary problems and then implement the normal Perceptron algorithm, to finally combine the results together and give a multi-class classification algorithm.

The implemented version of the algorithm is the Ridge Regression method, in order to penalize larger errors. In addition to the *regularization*, the algorithm can choose a *solver* which upon algorithm recommendation was chosen to be *lbfgs* or *Limited-memory BFGS* [4]. The Limited-memory BFGS is an optimization algorithm in the family of quasi-Newton methods that uses limited memory to approximate the BFGS algorithm. Quasi-Newton methods approximate Newton's method in order to find the 'flat' value (and hence minimize the cost), this operation involves the inverse of the Hessian matrix (partial derivatives matrix). This is very computational intensive and hence an approximation is used in BFGS, however as data grows the matrix becomes bigger and bigger, thus LBFGS further approximates this by only storing certain vectors from which the matrix can be restored. This solver proved to be the most effective in contrast with the rest available, *sgd*, *asgd* or *dual*.

This algorithm is trained using both the one-vs-one and one-vs-all approach to obtain the appropriate classification. One-vs-one is more computational intensive as more classifiers are needed, however as computational complexity is not a constraint for our problem it is preferred to use the one-vs-one approach and avoid possible imbalances introduced by one-vs-all. Upon testing of both approaches, one-vs-one performed better and so was selected for the algorithm. Like in all algorithms discussed cross-validation of 10-fold is used when possible to select the parameters within the model, and then train that hypothesis with all the training data.

Multi Class SVM

The multi-class SVM is a modification of the multi-class Perceptron discussed above. When setting the different parameters the main focus is not on the gradient descent or regularization methods, which are set to the default settings. Instead the main focus is concentrated on the implementation of the kernel used. The kernel used in the algorithm was the Gaussian or Radial Basis Function kernel which follows the equation:

$$G(x_j, x_k) = \exp(-||x_j - x_k||^2) \tag{1}$$

For a Gaussian kernel the functions in $\mathcal{H}$ are smooth, as they are determined by the kernel, thus as a result the selected hypothesis will be smooth. Usually this assumption is sensible for most datasets as smooth functions generally avoid over-fitting. Upon investigation it was confirmed the Gaussian kernel performed better than the other available kernels, $linear$ and $polynomial$ which are coarse for this application.

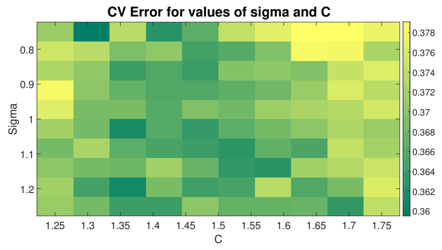

Figure 3: Cross-Validation Error for SVM parameters

The cross-validation parameters found to minimize the error were $\sigma = 0.75$ and $C = 1.306$, darkest point in Figure 3. These were found upon a 10-fold cross-validation procedure.

Neural Network

The main concerns are: number of layers, number of neurons in each layer and the transfer function used by the neurons.

The number of layers in the network implies a difficult evaluation of performance and over-fitting, multiple layers will be able to do more complex tasks hence being appropriate to give a minimum error, however too many layers tend to over-fit the data looking for a complex solution.

The general approach towards neuron per layer is to follow a descending number of neurons from layer to layer, however this is not always the optimal case. Again too many neurons will be able to perform the task even if it is complex, however the network will tend to over-fit the data.

As a result the number of layers and neurons per layer is found empirically to find those that minimize the cost function. These could be found by cross-validation or other validation method however the number of combinations of different implementations is excessively large to consider this. The final implementation that minimized the error the most was two layers with neurons per layer as [8 6].

Finally the transfer function used will determine the output delivered by neurons. The transfer function used was the Positive Linear transfer function, this follows a behavior like that in Figure 4. Other transfer functions include the Logarithmic Sigmoid, the Soft Max or the Positive Hard Limit transfer functions, but did not perform as good.

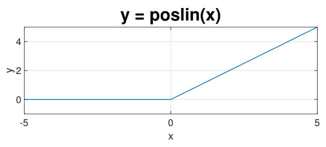

Figure 4: Positive Linear Transfer Function

5. Performance Assessment

All the algorithms where tested on three different sets of data: red wines, white wines and the combination of both.

Multi-Class Perceptron

	Classification Error		Mean Squared Error	
Dataset	$\hat{R}_n(w)$	$R(w)$	$\hat{R}_n(w)$	$R(w)$
Red Wine	42.09%	39.38%	0.5416	0.4667
White Wine	47.91%	48.81%	0.6827	0.6698
Combined	46.35%	48.33%	0.6612	0.6603

Table 3: Error Measures for Multi-Class Perceptron

The Multi-Class Perceptron's performance is not particularly better than the previously analyzed Ridge Regression algorithm. It performs better in the classification of Red Wines, but has worse performance for White Wines and the combination dataset. As a result this algorithm is not really better than other algorithms considered for the problem.

Multi-Class SVM

	Classification Error		Mean Squared Error	
Dataset	$\hat{R}_n(w)$	$R(w)$	$\hat{R}_n(w)$	$R(w)$
Red Wine	6.70%	35.00%	0.1582	0.4917
White Wine	7.29%	37.58%	0.1969	0.5970
Combined	10.64%	36.58%	0.2131	0.5408

Table 4: Error measures for Multi-Class SVM

The error measures for the Multi-Class SVM showed it is the best classifier. It shows a noticeable low training error, and performs best in test error by a reasonable margin. It follows the general trend of classifying red wines better, and has lowest MSE, hence producing low dispersion in the results. Figure 5 shows the confusion matrix of the results.

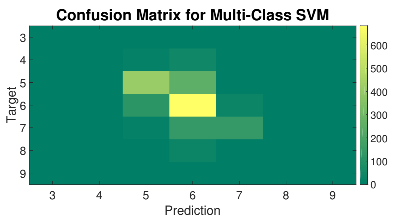

Figure 5: Confusion matrix for Multi-Class SVM

Neural Network

	Classification Error		Mean Squared Error	
Dataset	$\hat{R}_n(w)$	$R(w)$	$\hat{R}_n(w)$	$R(w)$
Red Wine	41.16%	38.96%	0.5330	0.4983
White Wine	46.82%	49.27%	0.6889	0.7293
Combined	47.40%	46.82%	0.6730	0.6864

Table 5: Error measures for Neural Network

The Neural Network algorithm performs best in the red dataset, as is for the rest of the algorithms. It performs very similar to the other best algorithms in classification error, however does not perform as well in MSE, suggesting the incorrect predictions have a larger offset with the actual rating.

6. Conclusion

To conclude, this report has presented various algorithms proposed to solve the Wine Classification problem. A note has to be made concerning the low reliability of the dataset, as discussed in this report, which conditions greatly the final result. Baseline predictors involved doing linear analysis to fit the data as best as possible, the Linear and Ridge Regression proved to be relatively competent given their simplicity and in fact performed very similar to some of the more complex algorithms, however being competent was not enough.

Out of the complex algorithms only the Multi-Class SVM justified its complexity, with the rest giving just slight improvements or performing worse. Considering the results and complexity of the algorithms the most appropriate algorithm to use to solve the set problem is the Multi-Class SVM. This performed best in the Classification and Mean Squared Errors, giving $R_{Class}(w) = 36.58\%$ and $R_{mse}(w) = 0.5408$ for the combined dataset.

The linear predictors give a small error given its complexity, and the more complex algorithms, apart from the SVM, do not a give a sufficiently low error given its complexity. As a result the proposed solution for the Wine Classification problem is to use a Multi-Class SVM.

Appendix

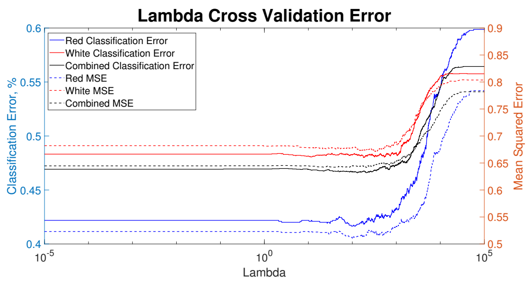

Figure 6: Training Error for different values of λ

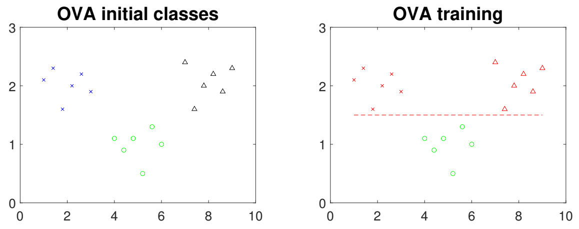

Figure 7: One-vs-all approach

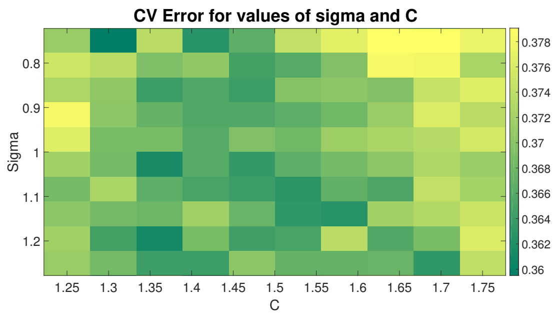

Figure 8: Cross-Validation Error for SVM parameters

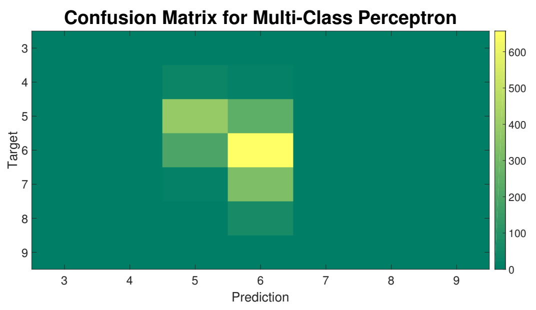

Figure 9: Confusion matrix for Multi-Class Perceptron

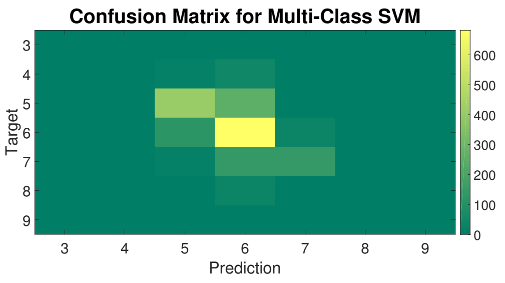

Figure 10: Confusion matrix for Multi-Class SVM

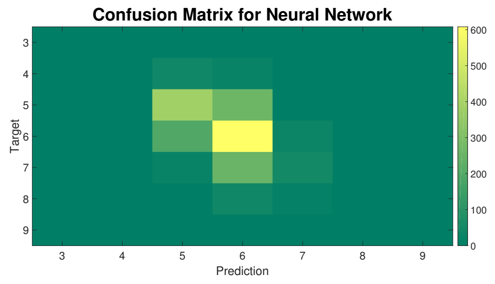

Figure 11: Confusion matrix for Neural Network

References

[1] P. Cortez et. al. “Modeling wine preferences by data mining from physicochemical properties.” In: *Decision Support Systems, Elsevier* 47.4 (2009), pp. 547–553. DOI: `http://dx.doi.org/10.1016/j.dss.2009.05.016`.

[2] Joel Silva. *What's an intuitive explanation of one-versus-one classification for support vector machines?* URL: `https://www.quora.com/Whats-an-intuitive-explanation-of-one-versus-one-classification-for-support-vector-machines/answer/Joel-Silva-10`.

[3] *Multiclass classification*. URL: `https://en.wikipedia.org/wiki/Multiclass_classification#One-vs.-rest`.

[4] *Limited-memory BFGS*. URL: `https://en.wikipedia.org/wiki/Limited-memory_BFGS`.