Investigating the optimization of hyperparameters using an RBF SVM, on different datasets

Edward Tallentire (edward.tallentire@postgrad.manchester.ac.uk)
MSc. Advanced Computer Science

***Abstract*—This paper looks at the tuning of hyperparameters for the Support Vector Machine (SVM) classifier. Methods such as Cross-Validation for the tuning of C and γ, Nested CV for an unbiased score of the classifier and ROC Curves for the analysis of the classifier will be used. This will be conducted on 5 different UCI Datasets (Dheeru & Karra Taniskidou, 2017) to see how different variables such as the number of examples and features or class distributions within datasets effect the classifiers optimal hyperparameters.**

Keywords: Hyperparameters, SVM, Nested Cross-validation, ROC Curves

I. Introduction

Tuning hyperparameters can at first seem a relatively simple problem, just change their values until an optimal score or lowest error result is reached, but this can lead to a multitude of problems. If the training data is skewed towards one class, will the classifier end up class biased?, reducing the validity of the model. If the hyperparameters are too finely tuned to the sample dataset will the real world use of the model give unreliable results?, How much training data is needed for a generalisable model?, And so on and so forth. Using Scikit Learn (Pedregosa et al., 2011) the SVM model will train on example data sets from the UCI Repository, methods including Nested Cross-Validation and ROC Curve analysis are used to tune two hyperparameters, ' C ' and ' γ '. Conclusions for optimal values of these variables are reached which are based on multiple items such as the number features in a dataset and size of dataset.

II. Background

A. Support Vector Machine (SVM)

The SVM model (Boser, Guyon, & Vapnik, 1992) is a classifier that allows non-linearly separable data to become linearly separable through projecting the data into n-dimensional space through the use of a 'kernel trick', once an optimal linear decision boundary is found it is projected back into the original n-dimensions. Various hyperparameters exists for each kernel that manipulate how the decision boundary is found, e.g.: for the rbf (Gaussian) kernel there exists two simple hyperparameters ' C ' and ' γ ', these two variables allow for outlying data to be 'miss-classified' to find an optimal decision boundary whilst maximizing the margin between data points and the decision boundary. Three other popular kernels that will not be used are Linear, Polynomial and Sigmoid.
Boser et al. (1992) states that 'Classifiers with a large number of adjustable parameters and therefore large capacity likely learn the training set without error, but exhibit poor generalization. Conversely, a classifier with insufficient capacity might not be able to learn the task at all. In between, there is an optimal capacity of the classifier which minimizes the expected generalization error for a given amount of training data.' This knowledge is useful when optimising the SVMs hyperparameters, if adjusted too finely then the model may end up overfitting on the data resulting in poor generalisability, if adjusted too little then the model may end up underfit, becoming useless.

$$E = \sum_{i=1}^{N} \max\left\{0, 1 - y_i f(x_i) - \xi_i\right\} + \frac{1}{2}\sum_{j=1}^{d} w_j^2 + C\sum_{i=1}^{N} \xi_i \quad (1)$$

B. Hyperparameters

The error function for the 'soft' margin SVM can be seen in equation 1, in the last part of the equation the **C** parameter applies a penalty to the sum of the amount slack (ξ) given to each data point. This penalty determines how strict the SVM solution is, if C is set to a small value such as 0.1 then the penalty for slack values will be small therefore a decision boundary with larger margins will be found, this will mean more data points will be miss-classified leading to an increased error, but the larger margins of the decision boundary allow the model to increase its generalisability as future unseen data points have a larger space between classes to be classified correctly.

C. The Radial Basis Function (RBF) Kernel

$$K(x_i, x') = e^{-\gamma(x_i - x')^2} \quad (2)$$

In equation 2 a larger γ results in a tighter area around each data point possibly resulting in overfitting, whereas a lower γ does the opposite. The RBF kernel is being used under recommendation from (Hsu, Chang, Lin, et al., 2003) due to its accessibility whilst still being very effective. Hsu et al. (2003) also states that an RBF kernel may not be suitable when the number of features is very large, therefore when looking at the 'Musk Molecule' dataset with its 167 features the amount of features supplied to the model will be varied to see if this advice holds true.

D. Validation Method

Varma and Simon (2006)'s study 'evaluated the validity of using the CV error estimate of the optimized classifier as an estimate of the true error expected on independent data.' They concluded that 'using CV to compute an error estimate for a classifier that has itself been tuned using CV gives a significantly biased estimate of the true error.' and 'A nested CV procedure provides an almost unbiased estimate of the true error.' (Krstajic, Buturovic, Leahy, & Thomas, 2014) also have similar findings. Therefore **Nested Cross-Validation** (NCV) will be used for evaluating the true error from the predictive model when tuning the hyperparameters.

E. Information Leakage

Another problem when creating a predictive model is **Information Leakage**, this is when information about your dataset e.g. the testing data is 'leaked' into your model when training, this may lead to overfitting which in turn gives an optimistic lower error when predicting with the model. Methods such as scaling or normalising data will need to be conducted separately to avoid this, for example scaling the training and testing data individually.

F. ROC Analysis

'The ROC curve shows the ability of the classifier to rank the positive instances relative to the negative instances' (Fawcett, 2006), in graph form this means that a value with any point along $(x, 1)0 => x <= 1$ will be a perfect classification for positive instances.
In some datasets the class distribution will be naturally skewed, this can lead to many different performance assessing metrics such as accuracy and F score, changing. Fawcett (2006) backs this up by stating 'In some cases, the conclusion of which classifier has superior performance can change with a shifted distribution.' and when talking about ROC curves 'They are able to provide a richer measure of classification performance than scalar measures such as accuracy, error rate or error cost.' therefore when assessing each classifiers performance both the Nested Cross-Validation error and analysis of the ROC graphs will be used when deciding the optimal classifier. The ROC Graph code was taken and adapted from (Pedregosa et al., 2011)'s web page 'Receiver Operating Characteristic (ROC)'

III. Experiments

A. Datasets

5 Datasets from the UCI machine learning repository (Dheeru & Karra Taniskidou, 2017) were used for this paper

- Breast Cancer (Wolberg, Street, & Mangasarian, 1995) This dataset contains 683 examples, is slightly skewed with a 444/239 class split and contains 10 numeric features
- Wireless Indoor Localization (Bhatt, Thakur, Narayanan, Perumal, & Rohra, 2017) This dataset contains 2000 examples, is skewed with a 500/1500 class split and contains 7 numeric features.
- Banknote authentication (Lohweg & Doerksen, 2013) This dataset contains 1372 examples, is relatively balanced with a 762/610 class split and contains only 4 numeric features.
- Musk molecule Version 1 (AI-Group, 1994a) This dataset is the balanced version of the Musk Molecule datasets, it contains 476 examples, is relatively balanced with a 207/269 class split and contains 166 numeric features.
- Musk molecule Version 2 (AI-Group, 1994b) This dataset is the unbalanced version of the Musk Molecule datasets, it contains 6598 examples which will be shuffled and cut down to 2000 due to computer processing restraints, is unbalanced with a 5581/1017 class split and contains 166 numeric features.

B. Method

When finding optimal C and gamma parameters for the SVM model via grid search, only a small number of values can be compared in one program execution, this is because the time needed to compute all possible parameter variations for the SVM is exponential. Therefore each calculated optimal C and gamma, taken from the mean value of the results in the iteration space, will be fed back into the parameter grid with a +/-0.5 and +/-0.05 variation respectively. The maximum values for each hyperparameter will be C = 10 and γ = 1. Once a value of C or gamma in the range of 0.2 and 0.02 respectively is repeated, the value shall no longer be manipulated to avoid potential overfitting of hyperparameters to the dataset. When these optimal values are found for each dataset an ROC Analysis will be conducted to find the usefulness of the model. Unfortunately data leakage via normalization of the entire dataset is unavoidable due to the nature of Scikit learns' KFold() and GridSearchCV() methods, in an ideal world the training folds would be normalised independently of the testing and validation folds.

C. Nested Cross-Validation vs Cross-Validation

The code for NCV has been taken and adapted from (Pedregosa et al., 2011) 'Nested versus non-nested cross-validation' web page. Before testing, the Wifi dataset was used to show the importance of Nested Cross-Validation, in figure 1 below two graphs are shown, both graphs describe the difference in the scored values for each CV and NCV over a certain iteration space. These graphs wont be used any further but show the importance of NCV when controlling for optimistic model scores.

D. Results

Table 1 shown below contains all the optimsed hyperparameters alongside the resulting Nested Cross-validation score and the ROC Area under Curve (AUC). Musk V1/2 is the Second Musk dataset that has been shuffled and reduced to 476 examples, this is to compare Musk V1 and Musk V2, an evenly distributed class dataset and a skewed one respectively.

Fig. 1. Non-Nested and Nested Cross Validation on Wifi Localisation Dataset

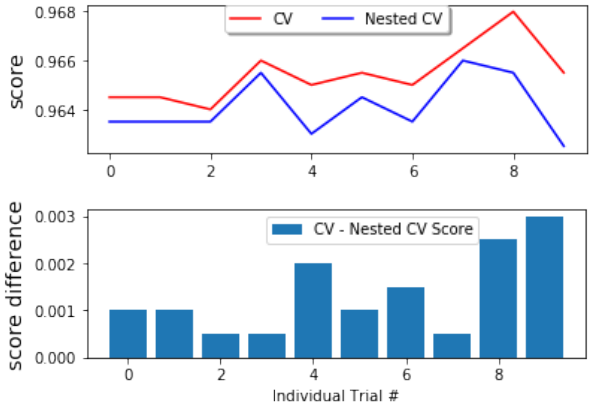
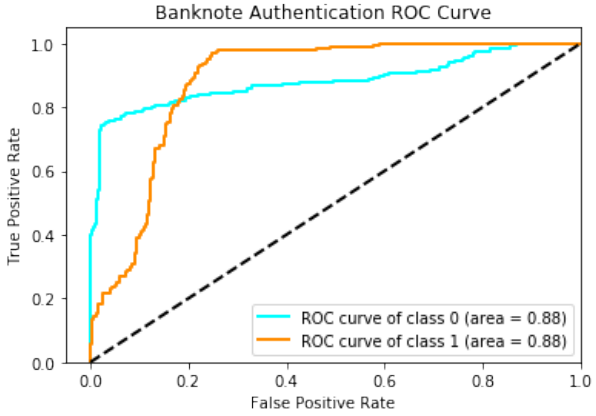
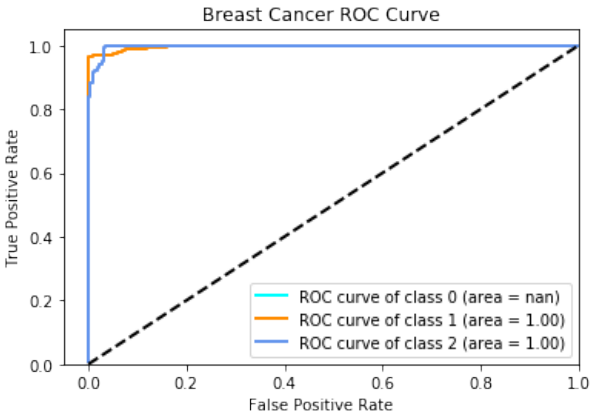
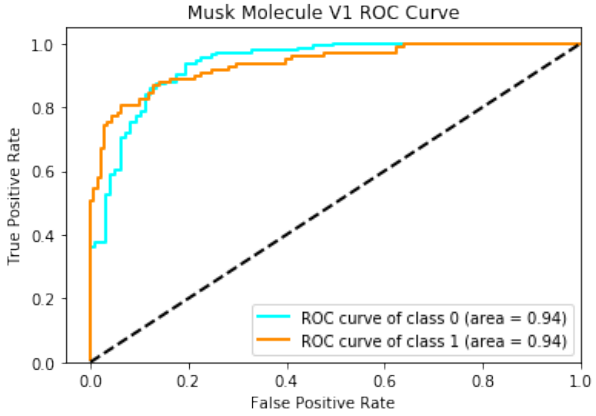


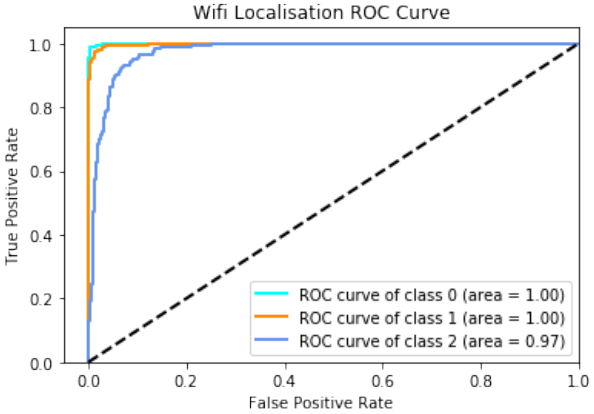
TABLE I
RESULTS TABLE OF EXPERIMENTS

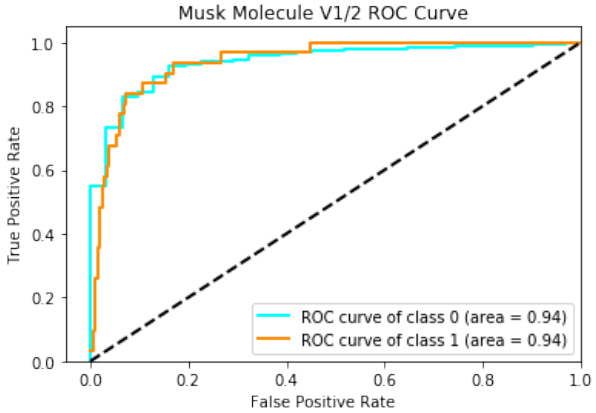
Dataset	Examples	Features	C	γ	NCV	AUC
BC	683	10	0.57	0.18	0.966	1
Wifi	2000	7	6	0.26	0.97	0.99
Banknote	1372	4	2.25	0.28	0.83	0.88
Musk V1	476	166	5.5	0.17	0.933	0.94
Musk V1/2	476	166	4.195	0.183	0.947	0.94
Musk V2	2000/6598	166	8.2	0.244	0.971	0.96

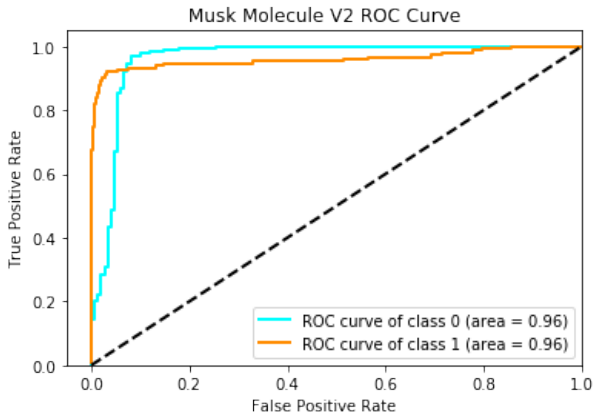












E. Analysis of results

1) Graph analysis

The Example - feature - C - γ distribution graphs can be seen in appendix 1. For the 'increasing C with increasing examples' graph we see not much change in the C value for the number of examples until we used the datasets with 2000 examples, these both needed the C value to be over 5.5 for an optimal hyperparameter. The second graph is more interesting, except for the last data point, the optimal gamma value seems to increase as the number of examples in the dataset increase. For the third graph 'increasing C with increasing features', there seems to be no valid pattern, there were low C values when the number of features where high and vice versa. The last graph has a similar findings with the gamma value not really varying as the number of features is varied.

For the first 2 datasets the ROC curves are perfect, the next 4 datasets also have a good looking curve except for Banknote authentication, which is interesting because this dataset has the most balanced class distribution of them all, this being said, the curve is still close to the ideal (0,1) point and shows decent classification performance.

2) Class skews

The 2 datasets with an even class distribution are Banknote and Musk V1, looking at Table 1 above we can see that these two datasets both have the lowest Nested Cross-Validation value, this may be because the even class distribution makes it harder for the model to give an unbiased result therefore the NVC is reporting a correct result which doesn't give us a 'false confidence' or false optimism when reporting on the generalisability of the model. These 2 datasets also have the lowest ROC Area Under curve value, albeit still a very high value.

There seems to be a high correlation for uneven class distributions to give high NCV and AUC scores, this may be because the model has an easier time classifying examples due to a biased model trained on an uneven training set which allow easier classification of true positives.

IV. Conclusion

In conclusion when finding optimal hyperparameters for use on a dataset, Nested Cross-Validation should be used with Cross-Validation to find the true error of the classifier. The only real correlation between tuning hyperparameters and the items we varied were between the number of examples and the gamma value, although this was a weak correlation at best. Unfortunately normalisation of data for individual folds was unable to take place and to what effect this had on our results is unknown. Real world bias should always be taken into account when generalising models over the world for example data sets such as face recognition will always be biased until all races are included in the data set.

References

AI-Group. (1994a). *Uci musk version 1.* Retrieved from http://archive.ics.uci.edu/ml/datasets/Musk+%28Version+1%29

AI-Group. (1994b). *Uci musk version 2.* Retrieved from http://archive.ics.uci.edu/ml/datasets/Musk+%28Version+2%29

Bhatt, R., Thakur, P., Narayanan, S., Perumal, B., & Rohra, J. (2017). *Uci wireless indoor localization.* Retrieved from http://archive.ics.uci.edu/ml/datasets/Wireless+Indoor+Localization

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/130385.130401 doi: 10.1145/130385.130401

Dheeru, D., & Karra Taniskidou, E. (2017). *UCI machine learning repository.* Retrieved from http://archive.ics.uci.edu/ml

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, *27*(8), 861–874. doi: 10.1016/j.patrec.2005.10.010

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). A practical guide to support vector classification.

Krstajic, D., Buturovic, L. J., Leahy, D. E., & Thomas, S. (2014). Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of Cheminformatics*, *6*(1). doi: 10.1186/1758-2946-6-10

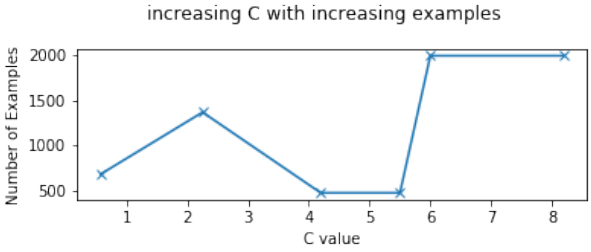
Lohweg, V., & Doerksen, H. (2013). *Uci banknote authentication.* Retrieved from http://archive.ics.uci.edu/ml/datasets/banknote+authentication

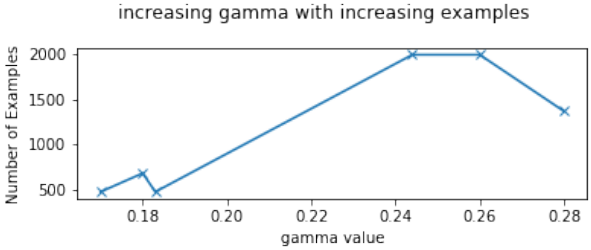
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

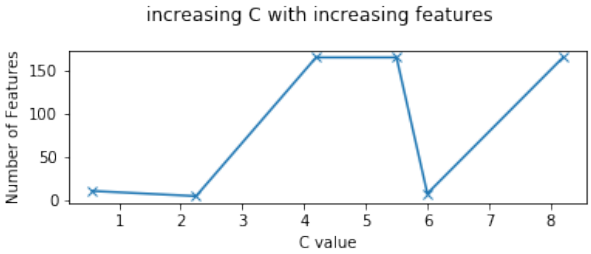
Varma, S., & Simon, R. (2006, Feb). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, *7*(91). doi: 10.1186/1471-2105-7-91

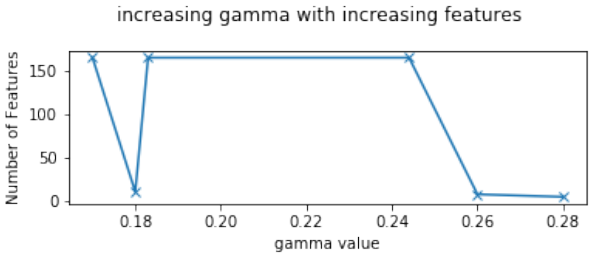
Wolberg, W., Street, N., & Mangasarian, O. (1995). *Uci breast cancer wisconsin (diagnostic).* (http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29)

APPENDIX


increasing C with increasing examples
Number of Examples
2000
1500
1000
500
1
2
3
4
5
6
7
8
C value


increasing gamma with increasing examples
Number of Examples
2000
1500
1000
500
0.18
0.20
0.22
0.24
0.26
0.28
gamma value


increasing C with increasing features
Number of Features
150
100
50
0
1
2
3
4
5
6
7
8
C value


increasing gamma with increasing features
Number of Features
150
100
50
0
0.18
0.20
0.22
0.24
0.26
0.28
gamma value