A Comparative Study in Recognizing Patterns through Multilayer Perceptrons and Support Vector Machines

Aimoré Resende Riquetti Dutra
Aimore.Resende-Riquetti-Dutra@city.ac.uk

Abstract

This paper aims to present a critical evaluation of two algorithm models performed in a supervised pattern recognition task on breast cancer traits. The two algorithms being compared are Feedforward Multilayer Perceptron (MLP) and Support Vector Machines (SVM). Different models are tried varying their hyperparameters in a grid search manner, and validated through a stratified cross validation. The tested results from the best evaluated models are compared by Confusion Matrices and Receiver Operation Curves (ROC). For such classification problem, the MLP algorithm demonstrated to be is preferable.

1. Introduction

Breast cancer is an ever-growing problem in the female community. According to data compiled by the Cancer Research UK, it shows that breast cancer diagnoses among women aged under 50 is increasing [1]. Whilst the number of female with breast cancer is increasing, identifying cancerous cells at an early stage in order to facilitate early diagnoses could be extremely beneficial for treating the disease.

The purpose of this paper is to critically evaluate two models designed to determine the diagnosis of breast cancer based on recognizing the pattern of 9 specific traits. These models considered are a Feedforward Multilayer Perceptron (MLP) and a Support Vector Machine (SVM). We investigate various configurations of these models and different data distribution to tackle the problem of breast cancer diagnosis.

In Section 2, we provide a brief description of the dataset used for training and testing the models. Section 3 covers the comparison of the approaches and methods used during implementation stage. Section 4 evaluates the results of the implementation whilst critically comparing the models. Section 5 concludes the paper.

1.1. Multilayer Perceptron (MLP)

The Multilayer Perceptron, also known as Artificial Neural Network, is the most regularly adopted technique for the purpose of pattern recognition [2]. MLPs are supervised learning classifiers that consist of an input layer, an output layer, and one or more hidden layers that extract useful information (features) during learning and assign modifiable weighting coefficients to components of the input layers.

In the first (forward) pass, the values coming from the input are modified by the weights assigned to the hidden units to determine the output. The output is compared with the target output, creating an error signal which is then back propagated. The connection weights are adjusted correspondingly in this back-propagation process. After the neural network has been trained, it is able to output a number which is coded as a class according to the input.

1.2. Support Vector Machines (SVM)

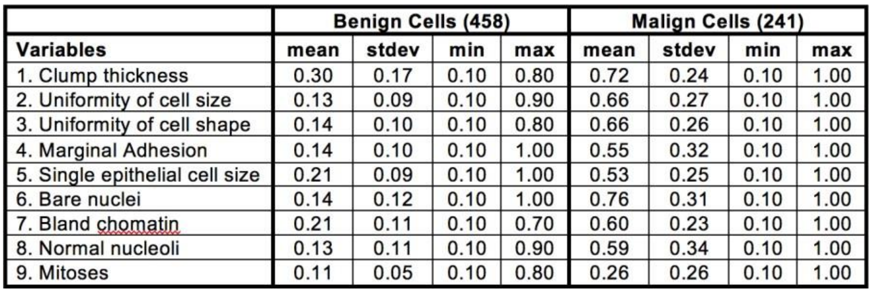
SVM can be formulated in the context of binary classification as the model that finds the decision boundary which maximizes the margin distance between two data classes. It also learns in a supervised paradigm. The key element of the SVM for a non-linear separable problem is the kernel trick, which takes the samples to an additional dimension so that the dataset can be then divided by a hyperplane [3]. SVM has two main advantages over the MLP, when working with multi variables. SVMs consider much fewer data points (only the support vectors) to do the calculations, and it can go to theoretically any number of hyper dimensions because it uses the dot product reducing the time spent in computation.

2. Dataset

The dataset used to perform analysis and experimentation is based on clinical data regarding breast cancer obtained from UCI Machine Learning Repository [4]. This dataset contains 699 biopsy samples with 9 attributes. It was identified that there was no need to perform normalization as the values of the dataset were already from 0 to 1. The variables and a summary of their statistics can be seen in the Table 1.

From Table 1, we observe that there is an imbalance number of normal (458 samples) and cancerous cell (241 samples) in the dataset. Since the difference in number of samples is not too massive, techniques such as SMOTE to equilibrate the number of samples are not compulsory [5]. Below are the two notations used for the report:
Class 1 = Benign Cells (Normal), Class 2 = Malign Cells (Cancerous).

Table 1 - Statistic Summary of The Dataset



	Benign Cells (458)				Malign Cells (241)			
Variables	mean	stdev	min	max	mean	stdev	min	max
1. Clump thickness	0.30	0.17	0.10	0.80	0.72	0.24	0.10	1.00
2. Uniformity of cell size	0.13	0.09	0.10	0.90	0.66	0.27	0.10	1.00
3. Uniformity of cell shape	0.14	0.10	0.10	0.80	0.66	0.26	0.10	1.00
4. Marginal Adhesion	0.14	0.10	0.10	1.00	0.55	0.32	0.10	1.00
5. Single epithelial cell size	0.21	0.09	0.10	1.00	0.53	0.25	0.10	1.00
6. Bare nuclei	0.14	0.12	0.10	1.00	0.76	0.31	0.10	1.00
7. Bland chomatin	0.21	0.11	0.10	0.70	0.60	0.23	0.10	1.00
8. Normal nucleoli	0.13	0.11	0.10	0.90	0.59	0.34	0.10	1.00
9. Mitoses	0.11	0.05	0.10	0.80	0.26	0.26	0.10	1.00

2.1. Initial Data Analysis

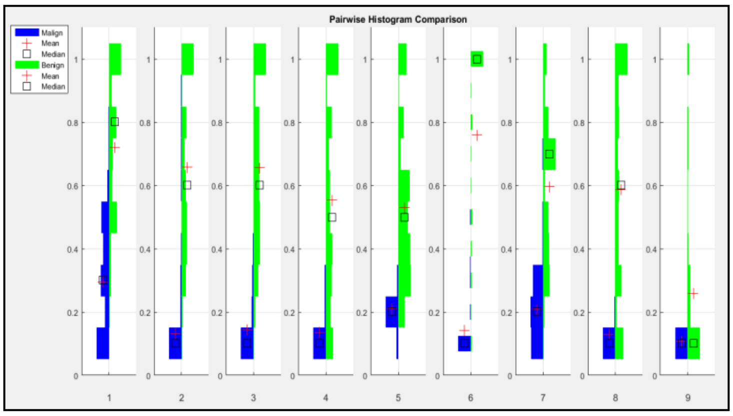


Figure 1. Pairwise Comparison Histogram for Benign and Malign Cells

As part of the initial data analysis process, pair wise comparison histograms were generated to visualize the distribution of the 9 variables available from the dataset, as shown in Figure 1. The purpose of such visual representation is to enable the attempt of obtaining some specific information hidden in the dataset. The pair wise comparison also best demonstrates the distribution of both classes within the dataset, allowing us to identify which variable is more valuable in distinguishing both classes. Because the dataset is presented in discreet form (intervals of 0.1), we have opted to use boxes rather than violin plot.

Some of the key findings obtained from the analysis of the pairwise comparison histogram includes the observation of normal cells having the tendency to be greater in size than that of cancerous cells for all the variables, except for variable 9 (Mitose). Variable 9 significantly shows that the two classes are rather similar in distribution. This means that normal cells and cancerous cells are very likely to have similar sizes for variable 9 (Mitose). Furthermore, the pairwise comparison histogram shows that variable 6 (Bare Nuclei) is the most obvious distinguishable variable between normal and cancerous cells as there is a vast difference between the distribution. Hence, suggesting that variable 6 is likely to be a useful determinant factor in deciding whether cell is cancerous.

3. Methods

In this section, we provide details for how the training, validation and testing steps were made, as well as a description of the architecture and hyperparameters used for building the MLP and SVM models.

3.1. Methodology

The methodology consists of holding out 10% of the original dataset as testing data, for the algorithm comparison process, between the best MLP and SVM models. The rest of the data (90%) is used for training and validating in the process of model selection, and also, used for training in the process of comparing both algorithms.

The **model selection** process entails a grid search to adjust the hyperparameters of both MLP and SVM models. The training and validation of each model is done in a 10-fold stratified cross-validation fashion, that divides the training and validation data by the same number of samples for both classes (benign and malign). According to [6] this approach is more accurate and reliable compared to simple k-fold CV as it obtains a higher accuracy and specificity and also provides a more stable network validation in terms of sensitivity. The average of the validation accuracies, over the cross-validation, was accessed and the best model was selected.

For the **algorithm comparison**, the models were retrained using the data training and validation data, and tested in the holdout data. In the MLP case, part of the training data (validation samples from the model selection process) was used for early stopping, because the technique tends to train better the network and it would not result in data snooping since there is still the hold out testing data to be used as unseen data [7]. Using validation check as early stop criteria allow us to increase the number of epochs and decrease the minimum training performance goal, giving more time for the network to train and letting it to pick the epoch moment where the best generalization and performance were achieved.

3.2. Architecture and Parameters used for the MLP

A “Bayesian Regulation backpropagation” training function is used to update the neurons weights. A softmax output function, which is commonly used for classification problems, was chosen, giving a probabilistic decision for the mutual exclusive outputs of the classification process; consequently, a cross entropy performance function was applied [8]. It is important to have 2 outputs in the neural network structure so that the softmax can be applied; even though with one output is possible to determine both classes (with or 0 or 1).

As part of the initial neural network process, it usually begins with random weight values to avoid networks being stuck in the same local minimum each time they are trained. The random weights in place will cause network to be initialized differently in each training process, hence causing the final results and accuracies to be different. We set the minimum training performance to 0.5, and the maximum number of epochs to 50, as early stopping criteria, because it was noticed, through the performance graphs, that most of the time high overfitting could be avoided if the training process was interrupted by the time these values were reached. In addition, the samples used for training, validating and testing will also affect the accuracy. This effect can be minimized by doing a k-fold cross-validation for the validating stage, but not for the testing one (new data). For the testing stage, we used 50 validation checks, with a minimum training performance of 0.001, and a maximum number of epochs of 300.

It was decided to vary the following hyperparameters: *learning rate*, *momentum* and *size of the hidden layers,* intending to increase the learning speed, avoid local minima and improve generalization and performance, respectively. [9]

3.3. Architecture and Parameters used for the SVM

SVMs do not begin with random weights, therefore they do not have the problem of falling into random local minima. However, SVM is still required to address the issue of training with different data sets which permits a more realistic validation of the model.

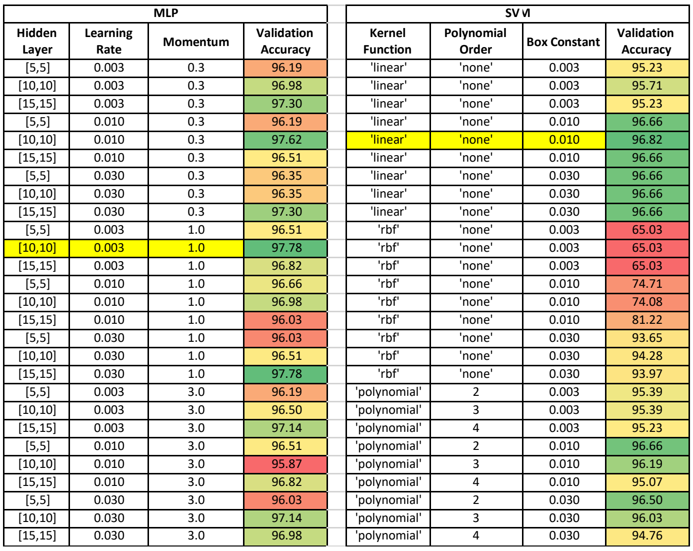
Usually the kernel function is the first parameter to be specified in a SVM. Defining it must be tried out, and it will depend on the complexity of how data is distributed. The box constrain is a parameter that controls the maximum penalty imposed on margin-violating observations, and aids in preventing overfitting (regularization). Increasing the box constraint assigns fewer support vectors; however, can lead to longer training times. [10] Besides the hyperparameters, SVMs do not have training parameters like the MLP do.

4. Results, Findings & Evaluation

4.1. Model Selection

Table 2 shows the hyperparameters grid search and the best models selected for MLP and SVM, highlighted in yellow. To test the generalisability of each best model, their prediction was compared with the class labels targets from the cross-validation process. In this grid search process, we noticed that MLP are extremely influenced by the initial neurons weights, because re-running the same grid search alters significantly the accuracies results of each MLP model but this does not happen for SVMs. On the other hand, SVMs are more sensitive by the combination of the hyperparameters. Accuracies can vary from 95 to 60% depending on it, but for MLP models this seems to not play a major role. The accuracies columns are ranked in a colored way to help visualize the combination of hyperparameters that contributed to improve or decrease the averaged validation accuracy.

Table 2 - Hyperparameters Grid Search



MLP				SV M			
Hidden Layer	Learning Rate	Momentum	Validation Accuracy	Kernel Function	Polynomial Order	Box Constant	Validation Accuracy
[5,5]	0.003	0.3	96.19	'linear'	'none'	0.003	95.23
[10,10]	0.003	0.3	96.98	'linear'	'none'	0.003	95.71
[15,15]	0.003	0.3	97.30	'linear'	'none'	0.003	95.23
[5,5]	0.010	0.3	96.19	'linear'	'none'	0.010	96.66
[10,10]	0.010	0.3	97.62	'linear'	'none'	0.010	96.82
[15,15]	0.010	0.3	96.51	'linear'	'none'	0.010	96.66
[5,5]	0.030	0.3	96.35	'linear'	'none'	0.030	96.66
[10,10]	0.030	0.3	96.35	'linear'	'none'	0.030	96.66
[15,15]	0.030	0.3	97.30	'linear'	'none'	0.030	96.66
[5,5]	0.003	1.0	96.51	'rbf'	'none'	0.003	65.03
[10,10]	0.003	1.0	97.78	'rbf'	'none'	0.003	65.03
[15,15]	0.003	1.0	96.82	'rbf'	'none'	0.003	65.03
[5,5]	0.010	1.0	96.66	'rbf'	'none'	0.010	74.71
[10,10]	0.010	1.0	96.98	'rbf'	'none'	0.010	74.08
[15,15]	0.010	1.0	96.03	'rbf'	'none'	0.010	81.22
[5,5]	0.030	1.0	96.03	'rbf'	'none'	0.030	93.65
[10,10]	0.030	1.0	96.51	'rbf'	'none'	0.030	94.28
[15,15]	0.030	1.0	97.78	'rbf'	'none'	0.030	93.97
[5,5]	0.003	3.0	96.19	'polynomial'	2	0.003	95.39
[10,10]	0.003	3.0	96.50	'polynomial'	3	0.003	95.39
[15,15]	0.003	3.0	97.14	'polynomial'	4	0.003	95.23
[5,5]	0.010	3.0	96.51	'polynomial'	2	0.010	96.66
[10,10]	0.010	3.0	95.87	'polynomial'	3	0.010	96.19
[15,15]	0.010	3.0	96.82	'polynomial'	4	0.010	95.07
[5,5]	0.030	3.0	96.03	'polynomial'	2	0.030	96.50
[10,10]	0.030	3.0	97.14	'polynomial'	3	0.030	96.03
[15,15]	0.030	3.0	96.98	'polynomial'	4	0.030	94.76

4.2. Algorithm Comparison

Table 2 displays the results of the testing stage comparing the best MLP and SVM models in a confusion matrix configuration. The validation samples, in the Validation Confusion Matrices, make part of the training samples, but they were plotted separately for convenience. Looking at the training process, which included all the available training data, the SVM model misclassified roughly 30% (18 over 26) less than the MLP. In the testing process; however, we see that both algorithms performed quite similar, misclassifying the same number of samples (4) for the unseen data.

In the model selection process, we saw that the best MLP model could achieve 97.78% accuracy and the best SVM model 96.82%. However, in the algorithm comparison process, where they have 10% more data to train than in the previous process, MLP performed worse with 95.68% [(1-(15+9+2+0)/(362+188+39+22))*100] and SVM performed better with 97.05% [(1-(8+8+2+0)/(363+187+39+22))*100]. The decrease in the accuracy for the MLP model can be explained by its random neuron`s weight initialization, and the increase for the SVM model accuracy was expected because more training samples were given.

Also, a Receiver Operation Curve (ROC) was plotted (Figure 3) to check the quality of the classifiers and visually enrich the analysis. In addition, the AUC (Area under a ROC), which measures the classification accuracy, is not influenced by the number of the samples in each class, an important advantage often used [11]. Most of the time unbalanced data is obtained, especially in this specific case, where cancerous cells are harder to find then normal ones. Remind that Class 1 refers to Benign cells, and Class 2 to Malign cells.

Although both algorithm seem to have a relatively same AUC, there should be one which is preferable. In the breast cancer diagnosis, it is more important to classify someone who does not have a threatening illness as someone who has, rather than the opposite. If the diagnosis exam fails to indicate that a person is ill, this person probably will get worse or even die. Therefore, having a True Negative Rate, from Class 1, equals to 1.0 (probability of classifying a sample as Malign equals 100%) is desirable. However, because our classifiers are not perfect some False Negatives from Class 1 (or FP from Class 2) must occur.

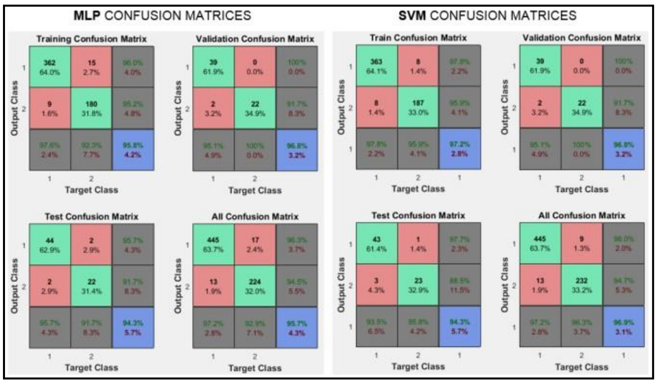


Figure 2. Confusion Matrices for MLP and SVM in the Test stage

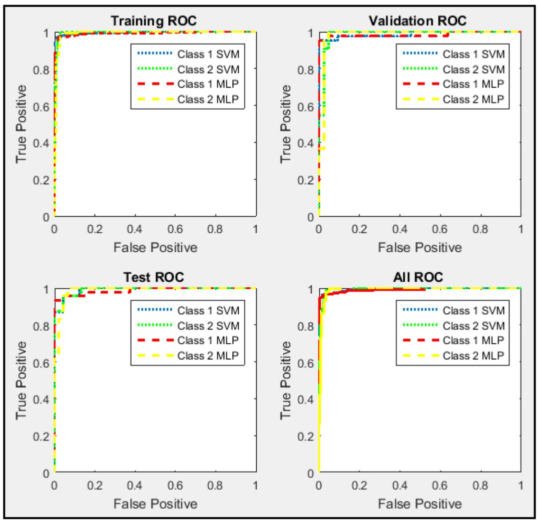


Figure 3. Receiver Operation Curve for MLP and SVM

Hence, the goal is to set the classifier threshold to classify all True Negatives from Class 1 minimizing the False Positives (or True Positives and False Negatives from Class 2, respectively). Examining the Test ROC plot in Figure 3, it is possible to see that maintaining a True Positive Rate (Class 2) of 1, the MLP model (yellow dashed line) has a lower False Positive Rate (Class 2) compared to the SVM (green dotted line). Therefore, this mean that the MLP is a better algorithm to use for this kind of problem and the considered dataset.

5. Conclusion

Our study reviews how accurately two trained models, a Multilayer Perceptron and a Support Vector Machine, can predict whether an unseen data (measurement of a breast cell) is in fact cancer or normal.

The conclusion is that MLP and SVM commensurable, but it is necessary to be aware that MLPs are highly contingent on its neuron`s weights, resulting in significant variations in the accuracies. Both models had similar results either in the training, validation and testing steps, having better accuracies in the training phase than in the testing, as expected. Nevertheless, the MLP showed to be a better algorithm for this specific problem, since it minimizes the false positives keeping the true negatives rate 100%.

We learned that a ROC is a very powerful tool for evaluating classifiers, measuring well their accuracies independently from the proportions of classes in the dataset and also giving a visual idea of the sensitivity and sensibility of the classifier model.

We believe that it would be very interesting to investigate the performance of both algorithms, MLP and SVM, using different training techniques, such as bagging [12] for example, to understand better what features are being extracted from the data. The idea is try to reduce the dependency of data allocation and random initial neuron`s weights increasing the number of training times so that this effect is minimized, which was not successfully achieved by the applied cross-validation in this work.

6. Reference

[1] Ahmad A S , Ormiston-Smith N and Sasieni PS. Trends in the lifetime risk of developing cancer in Great Britain: Comparison of risk for those born in 1930 to 1960' British Journal of Cancer (2015). DOI: 10.1038/bjc.2014.606

[2] Bishop, C.M., 1995. Neural networks for pattern recognition. Oxford university press.

[3] Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine learning, 20(3), pp.273-297.

[4] Murphy,P.M., Aha, D.W. (1994). UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[5] Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, pp.321-357.

[6] Mojarad, S.A., Dlay, S.S., Woo, W.L. and Sherbet, G.V., 2011. Cross validation evaluation for breast cancer prediction using multilayer perceptron neural networks. American Journal of Engineering and Applied Sciences, 4(4).

[7] Caruana, R., Lawrence, S. and Giles, L., 2000, November. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In NIPS (pp. 402-408).

[8] Michael, A.N., 2015. Neural networks and deep learning.

[9] Orr, G.B. and Müller, K.R. eds., 2003. Neural networks: tricks of the trade. Springer.

[10] Andrew, A.M., 2000. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods by Nello Christianini and John Shawe-Taylor, Cambridge University Press, Cambridge, 2000, xiii+ 189 pp., ISBN 0-521-78019-5 (Hbk,£ 27.50).

[11] Obuchowski, N.A., 2003. Receiver Operating Characteristic Curves and Their Use in Radiology 1. Radiology, 229(1), pp.3-8.

[12] Oza, N.C. and Russell, S., 2001, August. Experimental comparisons of online and batch versions of bagging and boosting. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 359-364). ACM.

Appendix 1 – glossary

Include definitions of all technical terms such as *perceptron*, *softmax*, *Bayesian regulation*, *generalizability*, etc.

Appendix 2 – Implementation details

Include any relevant implementation details such as negative results, changes made which improved results, training regimes and how to train the models to reproduce results.