Judith Grieves

A Comparison of Multi-Layer Feed Forward Network with Back Propagation and Support Vector Machine in the Prediction of Breast Cancer Recurrence

Abstract

This study evaluates 2 Neural Network models in the task of predicting Breast Cancer recurrence, given a dataset of diagnostic features. We find that the Multi-Layer Perceptron with Back Propagation and Support Vector Machine perform reasonably well and quite similarly with 75% accuracy. We note that the dataset used is rather small and a larger sample would achieve better and more resilient results.

Introduction

According to leading UK charity Breast Cancer Now [1], breast cancer is the most common cancer in the UK, with around 55,000 women and 370 men diagnosed each year. 87% of women survive breast cancer for five years or more and survival has doubled in the past 40 years but, each year around 11,500 women and 80 men die, making breast cancer the leading cause of death in women under 50 and the fourth most common cause of cancer death in the population overall.

Many of these deaths are as a result of distant recurrence following apparently successful treatment of the primary tumours. Estimates of recurrence vary, but a study over 10 years of 20,027 women between 65-80 [2] showed a recurrence rate of 36.8% with most of these cases (81.9%) occurring within 5 years after diagnosis.

Identifying patients at higher risk of recurrence would allow targeting of follow up diagnostic resources for patients who have already had treatment for a primary cancer, but according to [3] recurrence is not as well studied as breast cancer itself and data is difficult to obtain.

This paper uses the Ljubljana breast cancer dataset, containing attributes of patient diagnoses and an indication of whether their disease recurred. The dataset was obtained from the UCI machine learning repository [4] with the data initially provided by the Institute of Oncology, University Medical Center, Ljubljana, Yugoslavia [5]. We will use the data to train 2 different Neural Networks to predict the recurrence of breast cancer, based on diagnostic features, and compare and contrast the performance of the each.

Dataset Description

The original dataset has 286 instances, each one representing a diagnosis of breast cancer in an individual. Including the recurrence classifier, there are 10 attributes, each a feature of the individual patient or diagnosis, e.g. size of tumour (fig 1). The classifier is binary, representing whether or not the breast cancer had a recurrence. All attributes are either discrete or will be coded to create discrete attributes.

Attribute Name	Valid values	Coded values	missing data	data type
classifier	no-recurrence-events, recurrence-events	0 1	no missing data	nominal
age-band	20-29, 30-39, 40-49, 50-59, 60-69, 70-79	0 1 2 3 4 5 6	no missing data	ordinal
menopausal-status	premeno, lt40, ge40	0 1 2	no missing data	nominal
tumor-size	0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-5	0 1 2 3 4 5 6 7 8 9 10	no missing data	ordinal
node-involvement	0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 24-26	0 1 2 3 4 5 6	no missing data	ordinal
node-caps	yes, no	0 1	8 rows	nominal
malignancy-degree	1, 2, 3	1 2 3	no missing data	ordinal
breast-left-right	left, right	0 1	no missing data	nominal
breast-quadrant	right_up, right_low, left_up, left_low, central	0 1 2 3 4	1 row	nominal
irradiated	yes, no	0 1	no missing data	nominal

Fig 1. Breast Cancer Dataset attribute details

The distribution of the data attributes was visualised. For the classifier there are 85 recurrence vs 201 non-recurrence events – a recurrence rate of 29.7% - so the dataset is not perfectly balanced but not sufficiently imbalanced to suggest remedial action.

From a systematic review of breast cancer recurrence studies [3], it seems that there is no consensus on the best set of predictors for this disease so we look at correlations to identify a strategy.

Scatter plots are not particularly informative for a binary classifier so pairwise Pearson correlation values were calculated (fig 2). The Recurrence Classifier is most closely correlated with Malignancy, Nodes, Node Caps, Irradiated and Tumour size and least correlated with quadrant, menopausal and leftRight.

In order to identify any co-correlations in the data, we also calculate the Pearson value between pairs of attributes (fig 3). Strong correlations are noted between Age at Diagnosis and Menopausal status and between nodes/node-caps and between malignancy and node/node-caps.

	index	variable	value	abs_corr
60	class	malignancy	0.299400	0.299400
50	class	nodeCaps	0.276792	0.276792
40	class	nodes	0.276171	0.276171
90	class	irradiated	0.193912	0.193912
30	class	tumorSize	0.175065	0.175065
10	class	age	-0.071719	0.071719
70	class	leftRight	-0.058646	0.058646
20	class	menopausal	-0.052498	0.052498
80	class	quadrant	0.047380	0.047380

Fig 2. Pearson Correlations against Classifier

index	variable	value	abs_corr
menopausal	age	0.720322	0.720322
age	menopausal	0.720322	0.720322
nodes	nodeCaps	0.596347	0.596347
nodeCaps	nodes	0.596347	0.596347
nodes	malignancy	0.332824	0.332824
malignancy	nodes	0.332824	0.332824
nodeCaps	malignancy	0.325930	0.325930
malignancy	nodeCaps	0.325930	0.325930
nodes	irradiated	0.324621	0.324621
irradiated	nodes	0.324621	0.324621
nodeCaps	irradiated	0.303955	0.303955
irradiated	nodeCaps	0.303955	0.303955
tumorSize	malignancy	0.218169	0.218169
malignancy	tumorSize	0.218169	0.218169

Fig 3. Pearson co-correlation values

From this initial analysis, the best predictive features appear to be: malignancy, nodeCaps (or nodes), irradiated and tumorSize. Beyond these correlations, no others of any significance are noted. This will be taken into account in the training of the classifier.

To prepare the data for the learning algorithms, all features were translated to ordinal integer values (fig 1). This allows the data to be loaded as MATLAB matrices. As the quadrant attribute is not balanced, has missing values and is not correlated with the classifier in this dataset, it may be excluded from the analysis. As nodes and nodeCaps are highly co-correlated and nodeCaps has missing data it is also a candidate for exclusion.

Analysis of performance by feature selection in the experiments (from most to least correlated to the classifier) showed that the MLP performed best with all 9 potential predictors whilst the SVM had greatest accuracy for the 2 most strongly predictive features, Malignancy and Node Caps. These selections are used in the evaluation of the models.

The data is of good quality with minimal instances of missing data for node-caps and breast-quadrant. Node-caps is use in both models and breast-quadrant in the MLP so both are populated with the average value, where missing.

Neural Network Models

We will use the Breast Cancer dataset to train 2 different Neural Network learning algorithms, with the aim of comparing how suitable each model is for predicting disease recurrence from the set of attributes. Neural networks use data predictors and responses to learn from experience.

We chose 2 universal Feedforward Networks for this binary classifier task: Multi-Layer Feed Forward Network with Back Propagation and Support Vector Machine. Both are appropriate for binary classification and nonlinear regression problems. [6].

Multi-Layer Feed Forward Network with Back Propagation (MLP)

A MLP network consists of sets of nodes [6]: the input layer, one or more hidden layers and an output layer. Initial Node weights are arbitrarily set and input data is passed through the layers from input to output with the node weights fixed, calculating output as the sum of weight * input with the addition of a bias value and passed through an activation function.

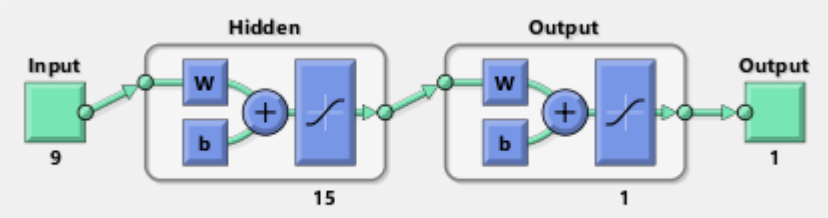

Fig 4 the Multi-Layer Feed Forward Network in MATLAB

The error back-propagation algorithm uses 2 passes through the network, one

forward and one backward. After the forward pass, the learned outcome is compared with the desired or target outcome by means of an error calculation. The error correction rule is used to adjust the network weights in the backward pass. This process continues for a number of iterations until the error is reduced to a parameter specified minimum value. Additionally, early stopping can be employed

in order to exit the iterations when, for example, there is a risk of over-training as the network becomes more accurate at predicting the training cases but less accurate at new unseen test data. MLP have many hyperparameters that can be adjusted to achieve the best performance.

Support Vector Machine (SVM)

For a binary classification problem, the Support Vector Machine feedforward network constructs a hyper-plane as the decision surface between the sets of positive and negative examples such that the separation between them is maximized [6]. The SVM is able to solve these types of problems 'close to the optimum with no domain knowledge but with a significant computing cost whilst the MLP is a more computationally efficient solution but has the onerous task of parameter selection and tuning'.

SVM is a good solution for linearly separable problems, and in non-linearly separable problems the use of a *kernel function* can transform the data points to a higher dimensional space to allow a linear separation [3].

Experiment Hypothesis

Our hypothesis is that breast cancer recurrence can be accurately predicted given a set of patient diagnostic attributes. We will attempt to prove this hypothesis with 2 Neural Network learning algorithms: a Multi-layer Feed Forward Network with back propagation and a Support Vector Machine. We anticipate that both will perform well on this type of Binary Classification problem.

Training and Evaluation Methodology

The original dataset is randomly split into test and training parts. The training set is used to train the classifiers with various parameters. Training functions will be executed for each combination of parameter values, described below, in a grid search, giving multiple results which show the performance of each.

The remaining test data is held back in order to generate predictions on the final learned network(s). Testing will evaluate the ability of the trained networks to generalise, by making predictions on previously unseen data.

For both algorithms, details of the parameter choice and accuracy results will be output to a csv file and analysed in MS Excel to investigate the effect of parameter choices

The trained networks will be evaluated for their ability to accurately predict the classifier outcome. Accuracy percentages are used to evaluate the models, i.e. True predictions / Total predictions. The prediction accuracy of each parameter set will be compared in order to establish an optimal combination of values, ie. that which gives overall best accuracy of predictions.

The best model for each algorithm will be saved and used to classify the test data. This will give a direct comparison of the accuracy of each, in order to compare/evaluate them against each other.

For the MLP, MATLAB's 'train' function will perform a further random splitting of the data into training-validation-test sets according to a parameterised ratio. The network is trained on the training set and 'validation vectors are used to stop training early if the network performance fails to improve or remains the same for max_fail epochs in a row. Test vectors are used as a further check that the network is generalizing well, but do not have any effect on training' [7].

For the SVM, MATLAB's 'crossval' function is used to validate the training results using a k-fold cross validation. This will divide the dataset into k folds, train the model k times each time on k-1 folds, using the kth fold as a validation set.

Parameters and Experimental Results

For both network types, we experiment with varying the proportion of training to held-out test data. A trial of train/test ratios for the best (SVM) parameters showed that training accuracy increase with higher proportions of training data but test accuracy peaks at 80/85% training. A ratio of 80% training (and 20% test) was used in final experiments.

Because of the relatively small size of the dataset used, the performance of both network types is subject to the randomisation of this initial train/test splitting. MLP performance is also subject to the initial random setting of node weights and the further splitting of the dataset for validation/testing. We investigated the impact of these randomisations on output but will seed random values for consistency in the final grid search.

Multi-Layer Feedforward network

Accuracy	epochs		
layers/mu	3	4	5
10	70.7%	80.2%	70.6%
0.001	59.8%	79.5%	64.6%
0.01	76.4%	79.9%	71.6%
0.1	76.0%	81.2%	75.5%
15	77.7%	77.3%	80.6%
0.001	77.3%	76.4%	84.3%
0.01	80.3%	80.3%	77.7%
0.1	75.5%	75.1%	79.9%
20	76.0%	80.9%	78.7%
0.001	73.4%	77.3%	78.6%
0.01	76.9%	83.0%	78.6%
0.1	77.7%	82.5%	79.0%

Fig 5. MLP grid search results – training accuracy

The network was created with a gradient-descent algorithm using Levenberg-Marquardt backpropagation. Training was carried out in batch mode, where the network weights are only updated after each pass of the training dataset. The *tansig* output function was applied. The default performance function, *Mean Squared Error* is used to calculate the errors. The network was created with an input node per dataset feature (9), a single hidden layer and an output node per classifier (1). The target error is defaulted to zero. In the grid search, we choose to vary (a) the number of nodes in the hidden layer, (b) *epochs,* the number of training iterations, and (c) mu, the momentum of the gradient descent. The most accurate overall predictions were seen with 15 hidden layers, 5 epochs and mu of 0.001 giving a training accuracy of 84% (fig 5).

Support Vector Machine

For the Support Vector Machine, we perform the grid search with a selection of *Kernel functions*, the order of the polynomial (where appropriate) and the *box constraint.*

Avg of Train Accy	box constraint				
Kernel/order	0.1	0.5	1	1.5	2
gaussian	71.6%	75.1%	75.1%	72.9%	75.1%
n/a	71.6%	75.1%	75.1%	72.9%	75.1%
linear	70.7%	72.9%	72.9%	72.9%	72.9%
n/a	70.7%	72.9%	72.9%	72.9%	72.9%
polynomial	73.7%	74.4%	73.7%	74.4%	74.4%
1	72.9%	72.9%	72.9%	72.9%	72.9%
2	72.9%	75.1%	75.1%	75.1%	75.1%
3	75.1%	75.1%	72.9%	75.1%	75.1%
RBF	72.1%	75.1%	75.1%	75.1%	75.1%
n/a	72.1%	75.1%	75.1%	75.1%	75.1%

Fig 6. SVM grid search results – training accuracy

The Kernel functions evaluated are polynomial, gaussian, linear and RBF and, as *box constraint* is defaulted to 1 within MATLAB, we vary the values between 0.1 and 2. The highest training accuracy (75.1%) was obtained for a number of parameter combinations so we chose a Polynomial kernel function of order 2 (fig 6) for the final model. The accuracy did not change for the various values of box constraint so in the final trained model it was set at an arbitrary value.

Analysis and Critical Evaluation

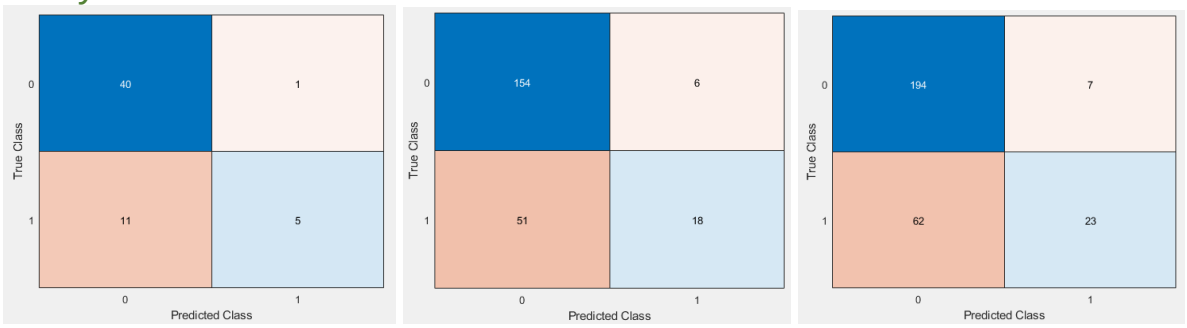

Fig 7. SVM Test Results - confusion matrix (a) Test set (b) Training set (c) all data

Running the SVM saved model for training and test data gave an accuracy of 75% (fig 7) for the training dataset and 79% for the test data, though this may be as a result of the small sample size - the accuracy of the overall dataset is 75%.

The MLP saved model gave 75% accuracy for both test and training datasets (fig 8).

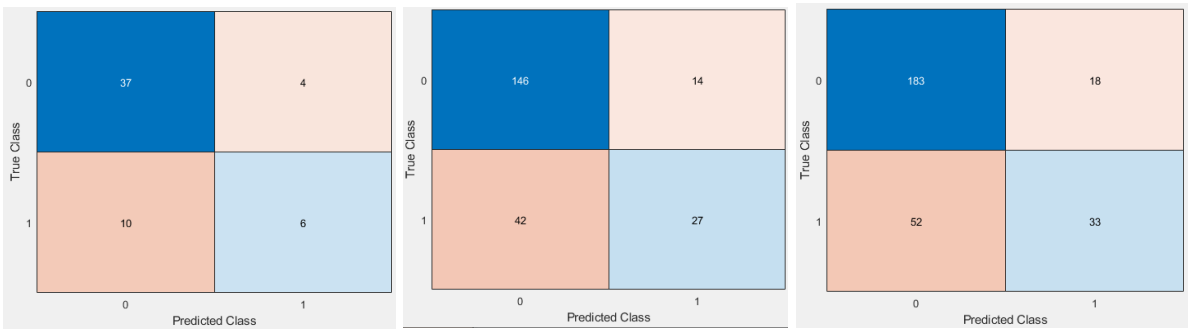

Fig 8. MLP Test Results - confusion matrix (a) Test set (b) Training set (c) all data

Both chosen networks perform the classification task reasonably well, though the SVM correctly classified 79% of the test instances, higher than the MLP at 75%. This accords with other studies [8] which have found SVM to be the better of the 2 for classification problems. Both network types performed the learning and grid search in a relatively short time although it was noted that for more than 2 features, the SVM with a polynomial kernel function of order > 2 takes an increasingly long time. There should be further work to evaluate this performance using a much larger dataset.

We note that within the 75% accurate predictions, the SVM model is slightly better at classifying the non-Recurrence events (97% correct vs 91%) but neither model does a good job of identifying Recurrence events (39% and 27% accurate). Similarly, of the incorrect predictions, the SVM has a higher proportion of False Negatives (62/69) and lower proportion of False Positives (7/69) than MLP (52/70 and 18/70).

Inaccurate results have a human consequence, ie. patients at risk of recurrence receiving a negative result, and vice versa. In these results, a FN result will fail to identify a patient at risk of recurrence while a FP will falsely identify a patient who is not at risk of recurrence. In a clinical setting, results would have to be discussed with and understood by stakeholders and must be interpreted with all caveats.

The SVM model was slightly more accurate for the unseen test cases. For the non-recurrence instances, correctly predicting 40/41 (vs the MLP 37/41) and for the recurrences, 11/16 (vs MLP 10/16). As the number of test instances is small this figure may not be reliable but this might also be explained as a result of the lack of recurrence data instances. Though the distribution of the classifier in this dataset was 70/30 (0/1) we did not to employ any data synthesis, such as *SMOTE*.

It would be interesting to apply these methods and compare results.

As previously stated, the random splitting of the dataset into train and test sets, coupled with the relatively small set of data, allows for some variation in distribution of Recurrence instances and therefore train and test results. For example, MLP training accuracy varied between 85-6% and test between 66-75%. A seeding of the randomised split function was chosen to give a balanced classifier split of the training and test files.

For the same set of parameters, the random allocation of the initial weights gave varying results; this randomisation was also seeded so that results could be reproduced.

For the MLP, there are large numbers of parameter combinations and it is a non-trivial task to choose the selection which optimised the final accuracy of the network. Especially difficult is balancing the competing aims of minimising overfitting and thereby maximising results for previously unseen test data.

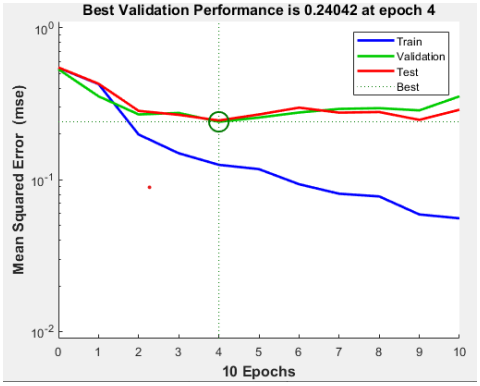

Fig 9. MLP test results – error performance graph

The performance graph of an MLP training iteration up to 10 epochs (fig 9) show that as the number of iterations increases, the errors observed in the training data reduce, but those for the previously unseen test data begin to increase – this shows that at higher iterations the network is at risk of overfitting and will not generalise well to new data. We compromise the number of iterations to minimise overfitting and aid generalisation by resetting the max epoch to stop at 5 where the test and train accuracy are maximised.

An experiment was run with the dataset columns normalised to evaluate whether this would affect results but no improvement was seen.

The dataset used was not particularly large and more examples of training data would undoubtably improve the accuracy and resilience of the results. Given the small amount of data available in this sample, a K-fold validation in the MLP in future work would help evaluate the training/validation mechanism. Also, we did not employ any synthesis of extra data to create noise in the data – this would be a useful future area of experiment.

Conclusion

According to these experiments, breast cancer recurrence can be predicted from a small number of patient attributes and both Neural Networks used achieved a similar performance of around 75% accuracy of predictions, though there were marked differences in the composition of the incorrect predictions (False Negative or False Positive) of each. The Support Vector Machine was less influenced by the randomisation of the file splitting and the choice of parameters whilst the Multi-Layer Perceptron required more selection and tuning of parameters.

The randomisation variation and resilience of outcome would improve with a larger sample. On reflection, this dataset was insufficiently large to give consistent results. Future work should include obtaining a larger dataset or using techniques to synthesise data.

References

[1] https://breastcancernow.org/about-us/media/facts-statistics

[2] Cheng, L., Swartz, M.D., Zhao, H., Kapadia, A.S., Lai, D., Rowan, P.J., Buchholz, T.A. & Giordano, S.H. 2012, "Hazard of recurrence among women after primary breast cancer treatment--a 10-year follow-up using data from SEER-Medicare", *Cancer epidemiology, biomarkers & prevention : a publication of the American Association for Cancer Research, cosponsored by the American Society of Preventive Oncology,* vol. 21, no. 5, pp. 800-809.

[3] Abreu, P., Santos, M., Abreu, M., Andrade, B. & Silva, D. 2016;2017;, "Predicting Breast Cancer Recurrence Using Machine Learning Techniques: A Systematic Review", *ACM Computing Surveys (CSUR),* vol. 49, no. 3, pp. 1-40.

[4] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer

[5] This breast cancer domain was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. Thanks go to M. Zwitter and M. Soklic for providing the data.

[6] Haykin, S. S. 1931- (Simon Saher) 1999, *Neural networks: a comprehensive foundation,* 2nd edn, Prentice Hall, Upper Saddle River, N.J., pp.156, 218, 318, 339.

[7] https://uk.mathworks.com/help/deeplearning/ref/trainlm.html

[8] Osowski, S., Siwek, K. & Markiewicz, T. 2004, "MLP and SVM networks - a comparative study", IEEE, , pp. 37.

Glossary

Levenberg-Marquardt	Levenberg-Marquardt is a variation of the gradient descent algorithm used to minimise the error and amend the weights in the back-propagation error calculations of a Multi-Layer Perceptron Neural Network model. It does this by increasing and decreasing the momentum value in the search for the error minima.
Epoch	In a Multi-layer feed forward network with back propagation, an epoch defines a number of forward and backward iterations through the network for a single pass of the training set data.
tansig	Hyperbolic tangent sigmoid transfer function. A MATLAB neural network function returning, for vectors of input values, the hyperbolic tangent value between 1 and -1.
Box Constraint	In a Support Vector Machine, box (constraint) applies a cost to the mis-classification of the training set, such that a high cost will lead to a stricter separation of the data.
Kernel function	In a Support Vector Machine, a kernel is a function applied to the support vectors that will project the data into a higher dimensional space such that the problem becomes linearly separable.
SMOTE	Synthetic Minority Over-sampling Technique, SMOTE is a technique to handle datasets where the distribution of a Classifier is not balanced. It uses a combination of under-sampling the majority class and over-sampling the minority class, by creating synthetic minority class examples.

Implementation Details

Exploratory Data Analysis in Python Notebook

For both neural network algorithms, training and testing code was implemented in MATLAB.

All programs are repeatable can be run stand-alone.

There are 2 additional programs to run the MATLAB suite for each model type:

RunSVM.m runs file splitter, train and test programs for the SVM model.

RunMLP.m runs file splitter, train and test programs for the MLP model.