Artificial Neural Networks ANN vs Support Vector Machines SVM, Pattern Recognition in Credit Card Defaults

By Kiril Tsarvenkov

kiril.tsarvenkov@city.ac.uk

1. Description and motivation of the problem

The big recession of 2008 has thought the financial industry an important lesson. Due diligence on clients and their credit scores are important aspects of the risk management systems of a bank as accumulation of bad loans can lead to economic disasters. Therefore, banks need to develop effective risk management models which can predict default on their client loans.

The purpose of this report is to evaluate the classification performance of two neural computing models in recognizing credit card clients which are likely to default. The adopted models are Multilayer Perceptron MLP with Backprogation also known as Artificial Neural Network ANN and Support Vector Machines SVM. The paper carries a number of experiments in order to find the best model for each algorithm which can solve the pattern recognition task of credit card client defaults.

This paper is organized as follows: Section 2 describes the data and provides summary statistics, Section 3 outlines the two algorithms with their advantages and disadvantages, Section 4 states the hypothesis of this paper, Section 5 provides information about the methodology, Section 6 describes the choice of parameters and experimental results, Section 7 analyses and evaluates the results and Section 8 concludes the paper with lessons learned and future work.

2. Description of the dataset

The dataset for this study is downloaded from the UCI Machine Learning repository and can be obtained from here: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients. The data contains 23 explanatory variables for credit card clients in the country of Taiwan. There is also a response variable which takes the value of "1" when a Default occurs and "0" when there is a Non-Default. An additional "reverse" feature is created by the author of this study which takes the value of 1 for Non-Default cases and 0 when a Default occurs. This is implemented in order to facilitate the

	Default				Non-Default			
Variable/Stat	**Mean**	**Std**	**Max**	**Min**	**Mean**	**Std**	**Max**	**Min**
LIMIT_BAL	130109.66	115378.54	740000.00	10000.00	178099.73	131628.36	1000000.00	10000.00
AGE	35.73	9.69	75.00	21.00	35.42	9.08	79.00	21.00
BILL_AMT1	48509.16	73782.07	613860.00	-6676.00	51994.23	73577.61	964511.00	-165580.00
BILL_AMT2	47283.62	71651.03	581775.00	-17710.00	49717.44	71029.95	983931.00	-69777.00
BILL_AMT3	45181.60	68516.98	578971.00	-61506.00	47533.37	69576.66	1664089.00	-157264.00
BILL_AMT4	42036.95	64351.08	548020.00	-65167.00	43611.17	64324.80	891586.00	-170000.00
BILL_AMT5	39540.19	61424.70	547880.00	-53007.00	40530.45	60617.27	927171.00	-81334.00
BILL_AMT6	38271.44	59579.67	514975.00	-339603.00	39042.27	59547.02	961664.00	-209051.00
PAY_AMT1	3397.04	9544.25	300000.00	0.00	6307.34	18014.51	873552.00	0.00
PAY_AMT2	3388.65	11737.99	358689.00	0.00	6640.47	25302.26	1684259.00	0.00
PAY_AMT3	3367.35	12959.62	508229.00	0.00	5753.50	18684.26	896040.00	0.00
PAY_AMT4	3155.63	11191.97	432130.00	0.00	5300.53	16689.78	621000.00	0.00
PAY_AMT5	3219.14	11944.73	332000.00	0.00	5248.22	16071.67	426529.00	0.00
PAY_AMT6	3441.48	13464.01	345293.00	0.00	5719.37	18792.95	528666.00	0.00

Table 1: Descriptive Statistics for the 14 Numerical Attributes of the Credit Card Clients Dataset

inputs and the outputs for the ANN classifier. Moreover, the dataset contains 9 categorical and 14 numerical variables. The data is collected from 30,000 individual banking clients. The number of default cases is 23,364 which means that 77.88% of the dataset contains instances of Non-Default and 22.12% instances of Default. There are no missing values. Table 1 shows the descriptive statistics for the 14 numerical attributes. The features BILL_AMT and PAY_AMT track the billing period of a client for a period of 6 months, BILL_AMT is the amount of the bill statement, PAY_AMT is the amount of the previous payments and LIMIT_BAL is the amount of the credit when the loan is initiated.

Figure 1 shows a correlation matrix for the numerical attributes of the data set. There are some strong correlation between the features, which is expected as the amounts of the BILL_AMT and PAY_AMT track the financial condition of a client over a number of months. This produces a correlation on a month by month basis, hence the correlations are strong in the region of 90% on a month to month basis and slowly decreasing to 70% between month 1 and month 6. A part from that, there are no other significant correlations in the data set.

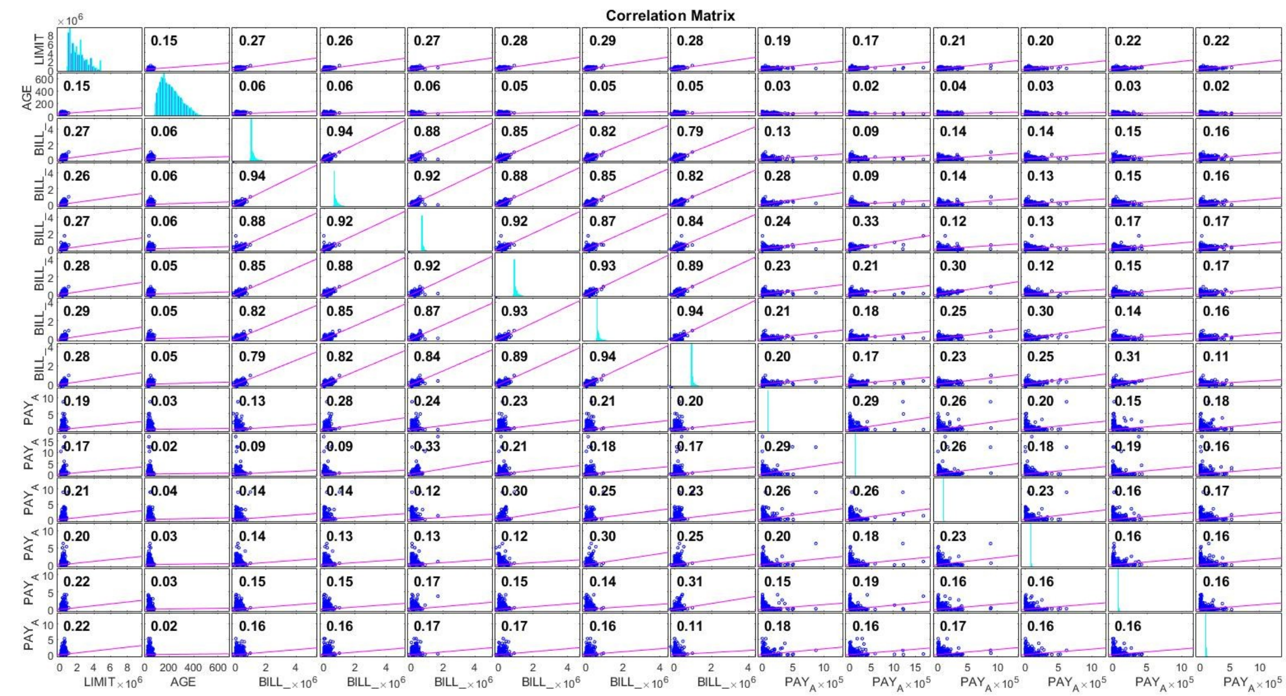

Figure 1: Correlation matrix for the 14 numerical attributes of the dataset

3. Brief summary of the two neural network models with their pros and cons

3.1 Artificial Neural Networks ANN

Artificial Neural Networks (ANN) are inspired by the biological learning systems which are built of complex neuron interconnections (Mitchell 1997). The algorithm consists of a number of layers which contain neurons connected by synapses. Typically, there would be an input, hidden and output layers where the input layer propagates the inputs forward through the computation nodes (synapses) in the hidden layer until a response is produced by the network through the output layer which classifies the original inputs. The computational layers and nodes of the network are connected with weights which remain fixed during the forward propagation which is also known as feed-forward step. Then, the actual response of the network is compared to a target value in order to obtain an error signal (error =

output – target). The error is propagated back through the layers, while the synaptic weights are adjusted in order to move the response of the network closer to the desired output. This process is iterated for all learning examples. As a result, the neural network is trained through the back-propagation algorithm (Haykin 1999).

ANN provide a number of benefits. Firstly, networks with smooth activation functions have output function derivatives that are able to approximate the derivatives of an unknown function. In that sense, ANN are *universal approximators*. As a result, they are able to approximate any function. Secondly, the hidden layers in an ANN can be used as replicators or identity maps. They can be used to detect features of the input data which is especially useful when domain knowledge is limited. In that sense, ANN perform feature extraction on behalf of the user. Finally, ANN are computationally efficient as their computational complexity is linear in the weights (Haykin 1999).

Despite the number of positive attributes of an ANN, there are some drawback to this approach. Firstly, the algorithms converges slowly. This is due to the fact ANN are an application of a statistical method known as stochastic approximation proposed by Robbins and Monro (1951). Secondly, since the algorithm is essentially a hill climbing method, it is possible to reach a local minima where changing the synaptic weights only increases the cost function. A possible solution is the use of momentum which can propagate the error to a global minimum from a local minima. However, the opposite scenario is also possible. In general, it is impossible to know if the algorithm has reached a global or local minima.

3.2 Support Vector Machines SVM

An alternative to ANN are Support Vector Machines SVM. They aim to construct hyperplanes as a decision surface with a maximized margin of separation between the separate classes. SVM achieve this through a procedure known as *structural risk minimization*. The performance of the algorithm largely depends on the extraction of a subset of the training data which constructs support vectors. These support vectors are used as reference points for the construction of an optimal hyperplane. There are three special cases of SVM which can be used in different settings: radial-basis function, polynomial learning machine and two-layer perceptron.

The SVM exhibit several beneficial characteristics. Firstly, the algorithm minimizes a quadratic loss function which produces two positive results. The first is that the algorithm is guaranteed to find a global extremum of the error surface and the second is that the computation can be performed efficiently. Secondly, SVM control the model complexity independently of dimensionality which bypasses the curse of dimensionality problem.

Despite the neat design of this algorithm, there are some challenges for SVM that need to be addressed. Firstly, the algorithm is computationally slower than other neural network models. This is mainly due to the lack of control over the number of data points selected by the algorithm for support vectors which could be very large. Also, prior knowledge is ignored which makes the learning task computationally intensive. Advanced studies have tried to solve these problems but with mixed success (Haykin 2009).

4. Hypothesis statement

Yeh and Lien (2009) argue that ANN are the only machine learning algorithm which correctly predicts the probability of a default in credit card clients data. In their study, ANN algorithm outperforms Decision Trees, Naive Bayes, Discriminant Analysis, Logistic Regression and K-Nearest Neighbor methods, however, their work does not examine the performance of SVM. Given the design of SVM, one would think that finding an optimal hyperplane will yield better results than simply finding a hyperplane which separates the classes, as is the case with ANN. Therefore, the main hypothesis of this paper is that SVM should outperform ANN in default classification tasks. However, it is also possible that SVM fail to achieve this, as the algorithms exhibits difficulties to perform when the classes are overlapping.

5. Description of choice of training and evaluation methodology

The dataset consists of 30,000 observations and it is split into training, validation and test sets. The training set consists of 24,000 data points while the validation and test set consist of 3,000 observations each. The validation test will be used to tune the parameters of the model and the overall performance of the algorithms will be assessed based on the results of the test data. Moreover, the result of the validation sets will be discussed to show how the behavior of the algorithms change with the variation of the parameters. In order to achieve this, a default model will be set for each algorithm, which will be compared to other models with experimental parameters developed in a ceteris paribus manner. The performance of the algorithms will be assessed using accuracy and F1 scores as the data set is left unbalanced and best models will be selected in order to compare SVM and ANN. The reason for leaving the data unbalanced is to maintain a degree of comparability between this and the previous study of credit card defaults by the author.

6. Choice of parameters and experimental results

6.1 Parameters ANN

The ANN is defined in terms of three parameters:

1. Number of Hidden Layers = 2
2. The learning rate – $\eta = 0.01$
3. Momentum - $\mu = 0.3$

The starting number of hidden neurons for the ANN algorithm is set at two, as the literature suggests that a neural network can perform successfully with as much as two hidden layers. The learning rate is set at $\eta = 0.01$ as it is preferred that the algorithm is slowly adjusting its weights. Furthermore, the momentum parameter is set at $\mu = 0.3$. In addition, the parameters of the model will be changed in order to explore how the predictive accuracy fluctuates given a parameter. Therefore, the experimental values for the ANN are set at 64 Neurons, the learning rate is set to 0.5 and Momentum at 0.9. For each parameter change a new ANN model is initiated. The results can be observed in table 2.

6.2 Parameters SVM

The SVM algorithm does not have explicit parameters as the ANN. The performance of the model is mostly affected by the choice of a kernel. There are three possible options: 1) Radial Basis Function RBF 2) Polynomial and 3) Linear kernels. Experiments are carried out with each option, in order to find the best model. The results can be observed in table 3.

6.3 Experimental Results

The results suggest that the most promising model for the SVM algorithm is the Linear model, which exhibits accuracy of 80.54% which is 1.57% more compared to the second best model – RBF with 78.97% (Table 3). The polynomial model performs worst with 76.61% accuracy on the test set and 78.67% on the validation set. As a result the Linear SVM is selected for comparison with the ANN.

ANN	Accuracy		F score	
Model	Validation Set	Test Set	Validation Set	Test Set
Default model	82.2	81.8	89.49	88.97
Momentum = 0.9	82.16	81.87	89.17	89.04
Neurons = 64	82.19	81.53	89.49	89.14
Learning Rate = 0.5	82.23	81.61	89.27	89.48

Table 2: Accuracy and F1 score for the Artificial Neural Networks Algorithm, the highlighted model is the chosen model for comparison with SVM

SVM	Accuracy		F score	
Kernel	Validation Set	Test Set	Validation Set	Test Set
Linear	81.83	80.54	89.13	89.21
RBF	79.37	78.97	88.13	86.92
Polynomial	78.67	76.61	82.83	83.82

Table 3: Accuracy and F1 score for the Support Vector Machines Classifier, the highlighted model is the chosen model for the comparison between the ANN and SVM

The result for the ANN algorithm are mostly uniform with very small margins of difference in accuracy. The highest accuracy is achieved for a model with the momentum parameter set at 0.9, however, this result is only 0.07% better compared to the Default model which achieves an accuracy of 81.8% (Table 2). Moreover, varying the number of neurons in the network and changing the learning rate do not improve performance, however, the results of these models are not considerably worse compared to the Default and Momentum models. As a result, the momentum model is selected for the comparison with the SVM algorithm.

7. Analysis and critical evaluation of results

The ANN algorithm reaches performance of 81.87% on the test set and 82.16% on the validation set. This is consistent with the original paper of Yeh and Lien (2009), where they achieve very similar results. Experimenting with different values for the parameters does not alter significantly the performance of the algorithm. This can be attributed to the hidden neurons which act as feature extractors, thus facilitating better learning. Moreover, the data set is sufficiently large for the neural network to adjust its weights with respect to the individual features which translates into a uniform performance across the experimental models. In addition, varying the activation function did not produce considerably different or better results.

The SVM model performs slightly worse than the ANN. The Linear SVM algorithm reaches accuracy of 81.83% on the validation set and 80.54% on the test set which is 1.33% less compared to the ANN. The choice of different kernels changes the accuracies of the SVM algorithm by a small margin. A polynomial kernel performs worse than the linear SVM model and the radial basis function models.

Given that SVM and ANN are two very different neural networks models which aim to find separation between the classes in a classification task, they achieve very similar performance as mentioned above. The major difference between them is the aim of the SVM to find an optimal hyperplane which seems to be irrelevant in this task as the results are very similar. Furthermore, SVM with radial basis function and polynomial kernels perform considerably worse by 1.9% and 5.16% in accuracy. Despite this difference, the optimal models for the algorithms reach almost identical results. They achieve this in a very similar manner as exemplified in table 4. Both algorithms find very similar number of True Positives TP, True Negatives TN, False Positives FP and False Negatives FN which means that neither model can find a better solution in distinguishing the Default from the Non-Default cases.

ANN	Test Set			Validation Set	
	Positive	Negative		Positive	Negative
Positive	2231	409	Positive	2227	411
Negative	117	243	Negative	126	236

SVM	Test Set			Validation Set	
	Positive	Negative		Positive	Negative
Positive	2209	434	Positive	2202	403
Negative	110	247	Negative	132	262

Table 4: Confusion Matrices for the ANN and SVM classifiers, for the Test and Validation sets

Despite the very similar performance of the algorithms, there are some notable differences in their architectures and they should not be considered interchangeable. For example, an ANN controls the complexity of the model by keeping the number of features (hidden neurons) small. In that regard the SVM controls the complexity of the model independently of dimensionality (Vapnik 1998 as seen in Haykin 2009). Moreover, the computational cost of the SVM is considerable for a learning task, whereas the ANN is much faster. However, the ANN requires more domain knowledge which can be a problem for a difficult learning task (Haykin 2009). In this research, the data has a number of useful attributes which aid the ANN. The result could be very different for a less suitable data set.

8. Conclusions, lessons learned, references and future work

This study compared the performance of two neural computing models in classifying client credit card defaults. The results of the experiments suggest that ANN perform more successfully than SVM in this classification task. Nonetheless, the difference in performance is rather marginal as the ANN outperform SVM by 1.33%. Moreover, the distinctive feature of the SVM, i.e. forming an optimal hyperplane does not seem to provide a better solution than a standard artificial neural net which rejects the hypothesis of this paper.

Future work will focus on feature engineering and developing a better model for the ANN classifier. This will involve Principal Component Analysis PCA for pre-processing and feature engineering with the aim to produce features with low correlations. This will aim to solve the problem of high correlations in the features BILL_AMT and PAY_AMT which can be problematic for a standard neural net model as suggested by Haykin (2009).

References:

Haykin, S.S., 1931 2009, *Neural networks and learning machines,* 3rd, International edn, Pearson Education, Upper Saddle River, N.J.

Mitchell, T. 1997, *Machine learning,* [International]. edn, McGraw-Hill, London;New York;.

Yeh, I. & Lien, C. 2009, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients", *Expert Systems With Applications,* vol. 36, no. 2, pp. 2473-

Haykin, S. S. 1931- (Simon Saher) 1999, *Neural networks: a comprehensive foundation,* 2nd edn, Prentice Hall, Upper Saddle River, N.J

Vapnik, V. (1998). *Statistical learning theory*. New York [etc.]: Wiley.

Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3), pp.400-407.