DJIA stock selection assisted by neural network

Tong-Seng Quah *

School of Electrical and Electronic Engineering, Nanyang Technological University, Republic of Singapore

Abstract

This paper presents methodologies to select equities based on soft-computing models which focus on applying fundamental analysis for equities screening. This paper compares the performance of three soft-computing models, namely multi-layer perceptrons (MLP), adaptive neuro-fuzzy inference systems (ANFIS) and general growing and pruning radial basis function (GGAP-RBF). It studies their computational time complexity; applies several benchmark matrices to compare their performance, such as generalize rate, recall rate, confusion matrices, and correlation to appreciation. This paper also suggests how equities can be picked systematically by using relative operating characteristics (ROC) curve.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Stock selection; Neural Network; DJIA

1. Motivation

There have been active researches on applying soft-computing models in areas of investment since decade ago. The main motivation is to develop an expert system to resemble the decision making process of investment experts. Soft-computing models are attractive as it offers a method to formulate the noisy and non-deterministic environments. As the cost of computational power decreases, we can afford to have more complex techniques which are expected to have lower signal-to-noise ratio.

There are two branches for applying soft-computing models on investment, which are technical analysis and fundamental analysis. Technical analysis is the most popular area in research. It is easier to predict due to the less noisy environment. Generally, what it applies are time-series prediction and pattern recognition. Such work includes equities price and volume movement. Technical analysis does not consider the underlying factors of equities’ financial health profile. Intuitively, this is only useful for short-term trading decision making.

Fundamental analysis is mostly for long-term investment decision. Accounting variables and financial ratios are usually used for inspecting the health of the investment products. Lesser studies were conducted in this area as compared to technical analysis. Since fundamentals will have stronger relationship to the price movement in the long-run, this makes them good candidates for neural network applications. We conduct a study on fundamental analysis with the three selected soft-computing models.

2. Literature reviews

2.1. Basic of soft computing

Soft computing is useful for solving problems which are described by multiple variables and multiple parameters. These problems may have non-linear coupling among these variables and parameters which are extremely difficult to find mathematical solutions. Therefore, it can be very costly to find solutions for such problems. To deal with such problems, one has to trade off the complexity with the uncertainties and imprecision. Thus, soft computing exploits the tolerance for imprecision, uncertainty, partial truth and approximation to achieve tractability, robustness and low solution cost.
2.2. Introduction of multi-layer perceptrons (MLP)

Fig. 1 shows a single layer of neurons. It contains S neurons and R inputs in the layer. In the network, each element of input vector \( P \) is connected to each neuron input through the weight matrix \( W \). The \( i \)th neuron has a summation function that gathers its weighted inputs and bias to form its own scalar output \( n(i) \). The various \( n(i) \) taken together form an \( S \)-element net input vector \( n \). Finally, the neuron layer outputs form a column vector \( a \). Multi-layer network can be created by feeding the outputs of layer to be input of next layer.

MATLAB toolbox help defines learning rule as a procedure for modifying the weights and biases of a network, as known as training algorithm (MATLAB). The learning rule is applied to train the network to perform some particular task. Learning rules may be broadly categorized as supervised learning and unsupervised learning. In supervised learning, the learning rule is provided with a set of examples (training set) which contain many pairs of inputs and target outputs. As the inputs are applied to the network, the network outputs are compared to the targets. The learning rule is used to adjust the weights and biases of the network in order to move the network outputs closer to the targets. In unsupervised learning, the weights and biases are modified in response to network inputs only. There are no target outputs available. Most of these algorithms perform clustering operations. They categorize the input patterns into a finite number of classes. This is especially useful in such application as vector quantization.

2.3. Introduction of radial basis neural network (RBF)

Radial basis neural network (RBF) is defined as means for interpolation in a stream of data as it has built into a distance criterion with respect to centre (RBF, 2006). Fig. 2 shows a radial basic network with \( R \) inputs. The net input to the \( \text{radbas} \) transfer function is the vector distance between its weight vector \( W \) and the input vector \( P \), multiplied by the bias \( b \). The radial basis function has a maximum of 1 when its input is 0. As the distance between \( W \) and \( P \) decreases, the output increases. Thus, a radial basis neuron acts as a detector that produces 1 whenever the input \( P \) is identical to its weight vector \( P \).

2.4. Introduction of adaptive neuro-fuzzy inference systems (ANFIS)

ANFIS system, which is an instance of the more generic form of Takagi–Sugeno–Kang (TSK) model replaces the fuzzy sets in the implication with a first-order polynomial equation of the input variables (Jang, 1993). Generally, a \( r \)-input one-output ANFIS system consists of rules in the following form:

\[
R_i : \text{IF } (x_1 \text{ is } A_{i1}) \text{ and } \ldots \text{ and } (x_r \text{ is } A_{ir}) \text{ THEN } y = f_i(x_1, x_2, \ldots, x_r) = b_0 + b_1x_1 + \ldots + b_rx_r,
\]

where \( x \) is the input vector, \( A_j \) is the fuzzy membership function, \( f_i \) is a first-order polynomial function, and \( b_j \) \( j = 0,1,\ldots, r \) are real-valued parameters. The fuzzy inference performed by ANFIS is an interpolation of all the relevant rules based on the physical location of the input data in the fuzzy subspaces. The predicted output of the model is generally given by the following equation:

\[
y = \frac{\sum_{i=1}^{L} a_i f_i(x_1, x_2, \ldots, x_r)}{\sum_{i=1}^{L} a_i} = \frac{\sum_{i=1}^{L} a_i (b_0 + b_1x_1 + \ldots + b_rx_r)}{\sum_{i=1}^{L} a_i},
\]

where \( a_i \) is the matching degree of rule \( R_i \), which is computed by considering the product of all the relevant membership functions of the rule.
2.5. Soft computing in financial market

Investment and financial trading problems are usually divided into two disciplines, which are fundamental analysis and technical analysis. Fundamental analysis techniques provide a framework for examining underlying forces which affect the price of an investment; on the other hand, technical analysis techniques analyze the past trading data, which includes prices, volume, open interest, etc. and believing these are reflecting the behavior of market participants (Vanstone & Tan, 2003). Generally, fundamental analysis is preferred for long-term investment whereas technical analysis is for short-term trading. However, they also complement each other for better trading decisions in the research on trading expert system, for example, the work of Baba et al. (2000).

The common soft computing techniques which are applied in both analyses are time series prediction, pattern recognition and classification, as well as optimization. Time series technique forecast future data points using historical data sets, for example, studying at the historical daily closing price in order to predict tomorrow closing price. Pattern recognition and classification try to classify observations into classes, for example, classifying securities into “winner” and “loser” classes. Optimization involves solving problems where patterns in the data are not known, for example, determining the optimal point to enter the securities market (Vanstone & Tan, 2003).

The focus of this article is to apply soft computing with fundamental analysis in DJIA equities picking.

2.6. Challenges

This problem is not as easy as it appears. There are well-known challenges for equities picking with soft computing, such as:

- Selection of additional features to improve the performance, as suggested by Falas, Charitou, and Charalambous (1994), Quah and Bobby Srinivasan (2000).

Intuitively, the more the features, the more accurate the neural networks performance are. However, the nature of financial market is noisy and stochastic. The stock market itself is not only driven by fundamental data, but also by human psychological factors and market principles. Because of this, the system may suffer the curse-of-dimensionality issue. Hence, the fundamental rule is to select most suitable features but not trying to cover as many features as possible. We will also present the time complexity comparison of the under studied soft computing models.

- Poor predictability accuracy. Due to the non-deterministic nature of the financial market, artificial neural network models may not be able to out-perform significantly but slightly to the logit model (Falas et al., 1994). With such, we will present not only the accuracy as one of the performance matrices, but also the appreciation of the picked equities, as (Vanstone, Finnie, & Tan, 2004).

- Data availability. It is highly impossible to obtain all the data that impact the stocks price movements. We need to maximize the accuracy and equities appreciation based on the limited data available (Quah & Bobby Srinivasan, 2000).

There are many other challenges, such as trading rules to simulate real life trading system to include trading cost, time management, etc. The fundamental objective is to build a reliable decision support system to replace expert knowledge in financial world.

3. Methodology

The history of fundamental analysis may be traced back to the work done by Benjamin Graham who is acknowledged as the father of modern security analysis. The work of Henry, Oppenheimer, and Schlarbaum (1981) suggested that it is possible and cheap to have positive risk-adjusted rates of return with Benjamin Graham’s common stock selection rules. This indicates the existence of relationship between the returns of the equities and their fundamental attributes, such as price-to-earning ratio, capitalization size of the firms. This finding spun off much research work, such as Kanas (2001), Aby Carroll, Briscoe Nat, Stephen, and Bacadayan (2001), Banz (xxx), Basu (1977), Henry et al. (1981), Lowe (1995), Joseph (2000), Reinganum Marc (xxx), Frankel and Lee (1998), De Bondt and Thaler (1987) as mentioned in Vanstone et al. (2004), Vanstone, Finnie, and Tan (2005). Their work further support the ten attributes which was proposed by Graham in his first book, “Security Analysis” in 1934 (Graham, 2004), to screen undervalued equities. Besides Graham’s ten attributes, Aby further developed another four fundamental rules for equities screening (Aby Carroll et al., 2001). As such, Vanstone chooses the combination of Graham’s rules and Aby’s rules to form the attributes of soft-computing models to identify high potential equities (Vanstone et al., 2004), which are listed in Table 1.

This work is therefore based on the above eleven attributes, selectively chosen by Vanstone, based on Granhom’s and Aby’s rules (Vanstone et al., 2004).

The classification problem of equities selection is defined as the following. “Class 1” is defined as any stock which appreciates in share price in value equal or more than 80% within one year, otherwise is classified as “Class 2”. This is in-line with (Vanstone et al., 2004; Vanstone et al., 2005) with the exception that we are using 80% cut-of-point instead of 100% to separate the data set into two classes. The reasons are that “Class 1” data can be increased by almost 50% if we lower the cut-of-point to 80%, and it is also highly desirable even if we have 80% of share price appreciation instead of 100%.
The nature of collected data set is imbalanced. Imbalance data essentially means at least one of the classes constitutes only a very small minority of the data (Chen, Liaw, & Breiman, 2004). And, the interest usually leans towards correct classification of the “rare” class (which we will refer to as the “Class 1” in our context). Refer to Chen et al. (2004), there are two common approaches to handle imbalanced data. One is by assigning high cost to misclassification of the minority class and trying to minimize the overall cost, which is called, cost sensitive learning. Another is to use a sampling technique, which is either down-sampling the majority class or over-sampling the minority class, or both. Down-sampling means reducing the size of the samples and over-sampling means blow up the samples by data replication. We will choose the latter as down-sampling may result in loss of information.

4. Results

The market under study is DJIA. A total of 1630 equities have been extracted with a period of ten years, from 1995 to 2004. We used all equities, including those that have been de-listed in order to avoid bias. For the required features with no data available, the value zero will be assigned. We then remove those entries that have missed more than half of the required features, in order to reduce the possible noise to the benchmark.

To be specific, every row contains eleven attributes and a known class, which is either “Class 1” or “Class 2”. The information of the “Class” forms the output of our soft computing models. The training set (eight years) consists of 10,243 input rows. Out of these 10,243 rows, there are 9,224 rows classified as “Class 2” and 1,019 “Class 1” inputs. This is an imbalanced data; “Class 2” dominates the data set but our interest is on identifying the minority class – “Class 1”. Over-sampling technique is applied on “Class 1”; it blows up “Class 1” from 1,019 input rows to 9,224 input rows and the data now is balanced by having half as “Class 1” and half as “Class 2. It is not necessary to apply over-sampling technique on validation set and test set, the soft-computing models only be trained with training set but not the rest.

In Neural Network methodology, the sample is often subdivided into “training”, “validation”, and “test” sets (Neural Network FAQ, 2006). Ripley (1996) contains detail discussion as well as the definitions. The following table summarizes the designs of the processed data (see Table 2).

Setting I will be used for Experiment I, which is for the comparisons of the accuracies and appreciation among the three soft computing models. We follows (Vanstone et al., 2004) 80:20 rules, which is using the first 80% (eight years) of the data set to predict the known results for the last 20% (two years) of the data set. Setting 2 will be used for Experiment II, which is for picking the most valuable equities by choosing the best cut off point for the soft computing models as well as picking the equities that will appreciate the most based on the significant of the output values. That is the reason of having validation set here, which is to choose the best cut off point, such that the appreciation of the signaled equities can be maximized.

4.1. Neural network training

MLP is configured with the number of hidden neurons being two-times of the input layer, which is twenty-two neurons. The training algorithm is gradient descent with

<table>
<thead>
<tr>
<th>Setting</th>
<th>Training set</th>
<th>Validation set</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1995–2002 (8 years)</td>
<td>N/A</td>
<td>2003–2004 (2 years)</td>
</tr>
<tr>
<td></td>
<td>Original: 10,243 inputs</td>
<td></td>
<td>Original: 2,422 inputs</td>
</tr>
<tr>
<td></td>
<td>Over-sampling: 18,448 inputs</td>
<td></td>
<td>No over-sampling is done</td>
</tr>
<tr>
<td>II</td>
<td>1995–2002 (8 years)</td>
<td>2003 (1 year)</td>
<td>2004 (1 year)</td>
</tr>
<tr>
<td></td>
<td>Original: 10,243 inputs</td>
<td>Original: 1,448 inputs</td>
<td>Original: 974 inputs</td>
</tr>
<tr>
<td></td>
<td>Over-sampling: 18,448 inputs</td>
<td>No over-sampling is done</td>
<td>No over-sampling is done</td>
</tr>
</tbody>
</table>

Table 3

Comparisons of computational time

<table>
<thead>
<tr>
<th>Soft-computing models</th>
<th>Computational time (for training)</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>188.45 s</td>
<td>Training algorithm: gradient descent with momentum; Hidden layers: 22 neurons; 500 epochs</td>
</tr>
<tr>
<td>ANFIS</td>
<td>396.85 s</td>
<td>Subtractive clustering*10 epochs</td>
</tr>
<tr>
<td>GGAP-RBF</td>
<td>360.7 min</td>
<td>After training: 90 neurons have been added.</td>
</tr>
</tbody>
</table>

* Subtractive clustering is a fast, one-pass algorithm for estimating the number of clusters and the cluster centers in a set of data MATLAB.
momentum and adaptive learning rate. Both the hidden layer neurons and output layers neurons have tangent sigmoid activation functions, which have the output values between -1 and +1. ANFIS is configured using subtractive clustering with a radius of 0.20. And, it is trained for 10 epochs. The trained ANFIS model has two rule nodes, each nodes is represented as a locally-defined linear functions. GGAP-RBF which has been proposed by Huang, Saratchandran, and Sundararajan (2005) in 2005, is also used in our comparative study. We applied the provided MATLAB source codes by Huang et al. (2005) for training. The CPU computational time for training with GGAP-RBF is 360.7 minutes. The time complexity for GGAP-RBF is obviously too high as compared to MLP and ANFIS, which spent 188.45 s and 396.85 s respectively, as summarized in Table 3.

For this problem, GGAP-RBF, which is proposed by Huang et al. (2005), obtains a total of 90 neurons after the six-hour training. The time complexity for GGAP-RBF is obviously too high as compared to MLP and ANFIS, which spent 188.45 s and 396.85 s respectively, as summarized in Table 3.

For this problem, GGAP-RBF, which is proposed by Huang et al. (2005), obtains a total of 90 neurons after the six-hour training. The time complexity for GGAP-RBF is exponential. The more the neurons are added, the slower the algorithm work. This shows that GGAP-RBF does not scale well with the growing numbers of inputs especially when there are large numbers of instances for financial problems (see Fig. 3).

4.2. Experiment I

In Setting I – Training set, we have in total of 18,448 input rows (also known as samples, instances or observations) which comprises 9,224 of “Class 1” data and 9,224 of “Class 2” data, after over-sampling technique has been applied on “Class 1”. This forms the input data for the soft-computing models. We can apply the same input data, which we used for training, to the trained models (also known as networks), to obtain the recall rate. Recall is the process of putting input data into a trained network and receiving the output; subsequently compare the output with the desire output (see Table 4).

The obtained recall rates are comparable to the studies performed by Falas et al. (1994), which are in the range of 57.9–65.7% for MLP models ². We further extend the study to the relationship between the predictions with the average appreciation in the equities price of the selected equities.

The analysis of the predictions performance of the soft-computing models against the known equities next-year share price appreciation (in terms of percentage, %), on the training set as shown in Figs. 4 and 5.

From the above observations, MLP and ANFIS models show good positive correlation for the appreciation of the equities price in the following year and the models’ predictions, for both training and test set of data. In contrast to MLP and ANFIS models, GGAP-RBF model gives a different scatter chart. It does not form a clear skewing curve. However, it is noticeable that there exist some correlation between the equities appreciation and the predicted output values from neural network along the −1 and +1 x-axis values.

The correlation between the percentage of appreciation of the equities share price and the predicted output values, has been shown in Table 5.

Table 4
Summary of accuracies results (recall rate) (choose intuitively: MLP and ANFIS: cut off point a of Zero; GGAP-RBF: cut off point of +0.25)

<table>
<thead>
<tr>
<th>Model</th>
<th>Recall rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>62.787</td>
</tr>
<tr>
<td>ANFIS</td>
<td>62.538</td>
</tr>
<tr>
<td>GGAP-RBF</td>
<td>54.51</td>
</tr>
</tbody>
</table>

² Benchmark on all manufacturing companies which are obtained from Compustat database.

Table 5

<table>
<thead>
<tr>
<th>Equities Share Price Percentage</th>
<th>Predicted Output Values from Neural Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>10%</td>
<td>10%</td>
</tr>
</tbody>
</table>

Please cite this article in press as: Quah, T.-S., DJIA stock selection assisted by neural network, Expert Systems with Applications (2007), doi:10.1016/j.eswa.2007.06.039
It is expected that GGAP-RBF has the lowest correlation between its predicted output values and the appreciation of the equities share price on Test set. Again this demonstrates that GGAP-RBF performs poorly for equities prediction problems, which belong to class of stochastic prediction problems.

### 4.3. Experiment II

Prediction accuracies do not necessarily lead to monetary return. The main profit is to pick the most valuable equities to be invested in such that we can receive high return for next-year equities appreciation. In this section, Setting II is used for experiment. Experiments show that as the value of cut off point increases, the number of selected or picked equities drop, and the rate of True Positive increases\(^3\) (Kubat & Matwin, 1997). This is true for all the models under studied. As such, we only present the experiment result for MLP in Fig. 6.

As the value of cut off point increases, the average appreciation of picked equities increases and the total number of signaled equities drops. As there is always a trade off between True Positive rate and True Negative\(^4\) rate, to

---

\(^3\) True Positive rate or Accuracy, is the proportion of “Class 1” cases that were correctly identified.

\(^4\) True Negative rate is the proportion of “Class 2” cases that were correctly identified.
maximize the output performance of the trained soft-computing models, we are interested to maximize the True Positive rate, at the same time, minimize the True Negative rate. This is the methodology of our study to configure the trained soft-computing models.

ROC curve has two variables from confusion matrix, which are True Positive rate (TP) and True Negative rate (TN) (Kubat & Matwin, 1997). We want to maximize True Positive rate and minimize True Negative rate for optimal performance. As a result, the interceptor point of TP and TN will be the best optimal cut off point for our problem. This is true to all the models under studied. As such, we only present the experiment results for MLP as Fig. 7.

By applying ROC curve to systematically pick the cut off point, we have the cut off points of 0.08, 0.08 and 0.27 for MLP, ANFIS and GGAP-RBF, respectively. The results are shown in Table 6 below.

ANFIS model has the highest precision and average appreciation of the signaled stocks. On the other hand, GGAP-RBF has demonstrated its low ineffectiveness in picking valuable equities.
The above experiment assumes that we have unlimited resources. With such assumption, we can trade as many equities as possible. What if we want to focus on a certain number of equities only, say top 10 equities? We have early demonstrated that there indeed has positive correlation between the outputs of trained models and the appreciation value.

Fig. 8 shows the average appreciation of the top picked equities. Intuitively, we can choose the top 10 of the signaled equities as the average appreciation is about 40–60% for all three soft-computing models (55.15%, 51.66% and 46.39% are obtained from MLP, ANFIS and GGAP-RBF models, respectively), which is about doubling the average market appreciation, 22.99%.

5. Conclusion

This paper shows that GGAP-RBF has huge time complexity as compared to MLP and ANFIS. Moreover, GGAP-RBF does not out-perform MLP and ANFIS in Recall Rate. The paper also shows that there is positive relationship between predictions of the trained networks with the equities appreciation, which may result in better earnings for investment. A systematic equities selection approach based on ROC curve is proposed. As investors may want to focus on limited number of equities, we can choose the equities based on the strength of the predicted output values from neural network. We demonstrated that, the higher the predicted values, the higher the chances of having positive appreciations.

The neural networks used here only trained against DJIA equities from 1995 to 2004. It is advised to do experiments on more years of data and different markets to study their impacts and whether the results obtained in this research is applicable. The study is based on eleven identified features. Features sensitivity analysis can be performed to understand the significance of each feature. Most of the time, we can reduce the eleven features to a lesser numbers. Moreover, Logit regression analysis can be applied in our developed environment to compare the results as Logit approach is still very popular in financial market. We can use Witten and Eibe (2005) software to achieve this.

We can further develop a trading system to simulate real-life trading activities based on the work of this dissertation. To do this, we need to include some trading rules, such as transaction cost, limited fund and transaction timing.
6. Uncited references

References


About Prof. Lotfi A. Zadeh: [http://www.cs.berkeley.edu/~zadeh/acpr.co.html](http://www.cs.berkeley.edu/~zadeh/acpr.co.html) (current April 8, 2006).