Stock Index Forecasting Using PSO Based Selective Neural Network Ensemble

Xueqin Zhang, Yuehui Chen
School of Information Science and Engineering
University of Jinan, Jinan 250022, P.R.China
Email: zhangxueqin_2005@163.com, yhchen@ujn.edu.cn

Jack Y. Yang
Harvard University
P.O.Box 400888 Cambridge, MA 02140 USA
Email: jyang@hadron.mgh.Harvard.edu

***Abstract*— Stock market analysis is one of the most important and hard problems in finance analysis field. Recently, the usage of intelligent systems for stock market prediction has been widely established. In this paper, a PSO based selective neural network ensemble (PSOSEN) algorithm is proposed, which is used for the Nasdaq-100 index of Nasdaq Stock Market^{SM} and the S&P CNX NIFTY stock index analysis. In the algorithm, each neural network is obtained by bagging and is trained by PSO algorithm, and then the networks selected according to the pre-set threshold are combined. Experimental results show that the improved algorithm is effective and outperforms GA based selective ensemble (GASEN) algorithm for the stock index forecasting problems.**

I. Introduction

During the last decade, stocks and futures traders have come to rely upon various types of intelligent systems to make trading decisions. Several intelligent systems have in recent years been developed for modeling expertise, decision support and complicated automation tasks, etc [1]. In this paper, two well-known stock indices namely Nasdaq-100 index of Nasdaq^{SM} [2] and the S&P CNX NIFTY stock index [3] were analyzed. The Nasdaq-100 index reflects Nasdaq's largest companies across major industry groups, including computer hardware and software, telecommunications, retail/wholesale trade and biotechnology [2]. The Nasdaq-100 index is a modified capitalization weighted index, which is designed to limit domination of the Index by a few large stocks while generally retaining the capitalization ranking of companies. Through an investment in Nasdaq-100 index tracking stock, investors can participate in the collective performance of many of the Nasdaq stocks that are often in the news or have become household names. Similarly, S&P CNX NIFTY is a well-diversified 50 stock index accounting for 25 sectors of the economy [3]. It is used for a variety of purposes such as benchmarking fund portfolios, index based derivatives and index funds. The CNX Indices are computed using market capitalization weighted method, wherein the level of the Index reflects the total market value of all the stocks in the index relative to a particular base period.

Leigh et al. [4] introduced a method for combining template matching, using pattern recognition and a feed-forward neural network, to forecast stock market activity. The authors evaluated the effectiveness of the method for forecasting increases in the New York Stock Exchange Composite Index at a 5 trading day horizon. Results indicate that the technique is capable of returning results that are superior to those attained by random choice.

Kim and Chun [5] explored a new architecture for graded forecasting using an arrayed probabilistic network (APN) and used a "mistake chart" to compare the accuracy of learning systems against default performance based on a constant prediction. Authors also evaluated several backpropagation models against a recurrent neural network (RNN) as well as probabilistic neural networks, etc.

Van den Berg et al. [6] proposed a probabilistic fuzzy systems to develop financial models where one can identify different states of the market for modifying ones actions. Authors developed a Takagi-Sugeno (TS) probabilistic fuzzy systems that combine interpretability of fuzzy systems with the statistical properties of probabilistic systems. The methodology is applied to financial time series analysis and demonstrated how a probabilistic TS fuzzy system can be identified, assuming that a linguistic term set is givenyue [7].

Because of their ability to deal with uncertain, fuzzy, or insufficient data which fluctuate rapidly in very short periods of time, neural networks (NNs) have become very important method for stock market predictions. Numerous research and applications of NNs in solving business problems has proven their advantage in relation to classical methods.

In recent years, neural network ensemble has become a very hot topic and has already been successfully applied to diverse real domains such as face recognition, handwritten digit recognition, optical character recognition, scientific image analysis, medical diagnosis, seismic signals classification, etc [8].

Recently, Zhou et al. showed for neural regressors that output continuous values, ensembling an appropriate subset of individual neural networks is superior to ensembling all the individual neural networks in some cases [8], [9], [10]. Based on this recognition, they proposed a method named GASEN, which trains several individual neural networks and then employs genetic algorithm to select an optimum subset of individual neural networks to constitute an ensemble.

In this paper, an improved PSO based Selective neural network ENsemble (PSOSEN) algorithm was put forward. We analyzed the Nasdaq-100 index value from 11 January 1995 to 11 January 2002 [2] and the NIFTY index from 01 January 1998 to 03 December 2001 [3]. The illustrations showed that

this algorithm was efficient in stock forecasting.

II. Neural Networks

The foundation of neural networks in a scientific sense begins with biology. The human brain consists of an estimated 10 billion neurons (nerve cells) and 6000 times as many synapses (connections) between them. All information taken in by a human is processed and assessed in this particular part of the body. A neuron in itself is relatively slow compared to a silicon logic gate. However, this amazing amount of neurons and synapses suites as compensation. Thus the brain operates as nothing less than a complex, non-linear and parallel computer. With this notion present we are ready to describe a neural network mathematically [11].

A typical neural network consists of layers. In a single layered network there is an input layer of source nodes and an output layer of neurons. A multi-layer network has in addition one or more hidden layers of hidden neurons. Some standard three-layer feed-forward networks are used in this research.

A representative feed-forward neural network consists of a three layer structure: input layer, output layer and hidden layer. Each layer is composed of variable nodes. The type of this network is displayed in Fig.1. The number of nodes in the hidden layers is selected to make the network more efficient and to interpret the data more accurately. The relationship between the input and output can be non-linear or linear, and its characteristics are determined by the weights assigned to the connections between the nodes in the two adjacent layers. Changing the weight will change the input-to-output behavior of the network.

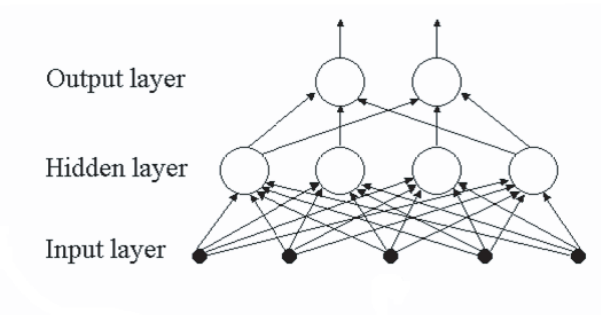

Fig. 1. A fully connected feed-forward network with one hidden layer and one output layer

A feed-forward neural network analysis consists of two stages, namely training and testing. During the training stage, an input-to-output mapping is determined iteratively using the available training data. The actual output error, propagated from the current input set, is compared with the target output and the required compensation is transmitted backwards to adjust the node weights so that the error can be reduced at the next iteration.

The training stage is stopped once a pre-set error threshold is reached and the node weights are frozen at this point. During the testing stage, data with unknown properties are provided as input and the corresponding output is calculated using the fixed node weights. The feed-forward neural network has been shown to perform well in many areas in previous research.

III. Selective Neural Network Ensemble

For most regression and classification problems, combining the outputs of several predictors improves on the performance of a single generic one [20]. Formal support to this property is provided by the so-called bias/variance dilemma [21], based on a suitable decomposition of the prediction error. According to these ideas, good ensemble members must be both accurate and diverse, which poses the problem of generating a set of predictors with reasonably good individual performances and independently distributed predictions for the test points. Diverse individual predictors can be obtained in several ways. These include: (i) using different algorithms to learn from the data (classification and regression trees, artificial neural networks (ANNs), support vector machines, etc.), (ii) changing the internal structure of a given algorithm (for instance, number of nodes/depth in trees or architecture in neural networks), and (iii) learning from different adequately- chosen subsets of the data set. The probability of success in strategy (iii), the most frequently used, is directly tied to the instability of the learning algorithm. That is, the method must be very sensitive to small changes in the structure of the data and/or in the parameters defining the learning process. Again, classical examples in this sense are classification and regression trees and ANNs. In particular, in the case of ANNs the instability comes naturally from the inherent data and training process randomness, and also from the intrinsic non-identifiability of the model. In what follows, three ensemble methods are employed for the stock index forecasting problems.

Neural network ensemble [12] utilizes a set of trained neural networks for the same task. In the beginning of the 1990's, Hansen and Salamon showed that the generalization ability of a neural network system can be significantly improved through ensembling neural networks, i.e. training several neural networks and combining their results in some way. Later, Sollich and Krogh defined neural network ensemble as a collection of a number of neural networks that are trained for the same task [9]. Since it behaves remarkably well and is easy to use, neural network ensemble is regarded as a promising methodology that can profit not only experts in neural computing but also ordinary engineers in real world applications.

In general, a neural network ensemble includes two steps: (1) training a number of component neural networks; (2) combining the component predictions.

For training component neural networks, the most prevailing approaches are Bagging and Boosting. Bagging is proposed by Breiman based on bootstrap sampling. It generates several training sets from the original training set and then trains a component neural network from each of those training sets. Boosting is proposed by Schapire and improved by Freund et al. It generates a series of component neural networks whose training sets are determined by the performance of former

ones. Training instances that are wrongly predicted by former networks will play more important roles in the training of later networks [10].

For combining the prediction of component neural networks, the most prevailing methods are simple averaging or weighted averaging for regression tasks, and plurality voting or majority voting for classification tasks.

IV. Particle Swarm Optimization

The Particle Swarm Optimization(PSO) [13], [14], [15] algorithm was first introduced by James Kennedy and Eberhart as an alternate tool to genetic algorithms (GA) and gained a lot of attention in various optimal control system applications. In PSO, positions of N particles are candidate solutions to the D-dimensional problem, and the moves of the particles are regarded as the search process of better solutions. The position of the $i-th$ particle at t iteration is represented by $X_i(t) = (x_{i1}, x_{i2}, \cdots, x_{iD})$, and its velocity is represented by $V_i(t) = (v_{i1}, v_{i2}, \cdots, v_{iD})$. During the search process the particle successively adjusts its position according to two factors: one is the best position found by itself ($pbest$), denoted by $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$; the other is the best position found so far by its neighbors ($gbest$), denoted by $P_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$. The neighbors can be either the whole population (global version) or a small group specified before run (local version). The velocity update equation (1) and position update equation (2) are described as follows.

$$V_i^{(t)} = w * V_i^{(t-1)} + c1 * rand() * (P_i - X_i^{(t-1)}) + c2 * rand() * (P_g - X_i^{(t-1)}) \quad (1)$$

$$X_i^{(t)} = X_i^{(t-1)} + V_i^{(t)} \quad (2)$$

Where w is inertia weight which balances the global exploitation and local exploration abilities of the particles, cl and $c2$ are acceleration constants, $rand()$ are random values between 0 and 1. The velocities of the particles are limited in $[Vmin, Vmax]^D$. If smaller than $Vmin$, an element of the velocity is set equal to $Vmin$; if greater than $Vmax$, and then set equal to $Vmax$ [16], [17].

V. PSOSEN Algorithm

PSO has been developed through simulation of simplified social models. It is a stochastic search technique with reduced memory requirement, computationally effective and easier to implement compared to other evolutionary algorithms (EAs) [18]. Also, PSO will not follow survival of the fittest, the principle of other EAs. PSO has very fast converging characteristics when compared to other EAs.

A. GASEN Algorithm

GASEN algorithm is put forward to find out the neural networks that should be excluded from the ensemble. The basic idea of the approach is a heuristics, i.e. assuming each neural network can be assigned a weight that could characterize the fitness of including this network in the ensemble, and then the networks whose weight is bigger than a pre-set threshold λ could be selected to join the ensemble [10].

In GASEN algorithm, $\widehat{E}_w^V$ denotes the estimated generalization error of the ensemble corresponding to the individual w on the validation set V. It is obvious that $\widehat{E}_w^V$ can express the goodness of w, i.e. the smaller $\widehat{E}_w^V$ is, the better w is. So, GASEN uses $f(w) = 1/\widehat{E}_w^V$ as the fitness function. The GASEN approach is summarized as follows:

Input: training set S, learner L, trials T, threshold λ
Procedure:
1. for t = 1 to T $\{$
2. S_t = bootstrap sample from S
3. N_t = $L(S_t)$
4. $\}$
5. generate a population of weight vectors
6. evolve the population where the fitness of a weight vector w is measured as $f(w) = 1/\widehat{E}_w^V$
7. w^* = the evolved best weight vector
Output: ensemble N^*

$$N^*(x) = Ave \sum_{w_t^* > \lambda} N_t(x)$$

Where S_1,S_2,...,S_t are T bootstrap samples generated from the original training set; N_t is a individual neural network trained from each S_t; N^* is the ensemble whose output is the average output of the component networks in regression [9], [10].

B. PSOSEN Algorithm

PSO algorithm is a relatively recent heuristic search method whose mechanics are inspired by the swarming or collaborative behavior of biological populations. PSO algorithm is similar to GA in the sense that these two evolutionary heuristics are population-based search methods. In other words, PSO algorithm and GA move from a set of points to another set of points in a single iteration with likely improvement using a combination of deterministic and probabilistic rules. The drawback of the GA is its expensive computational cost. PSO algorithm has the same effectiveness (finding the true global optimal solution) as the GA but with significantly better computational efficiency (less function evaluations) [19]. So an improved algorithm named PSOSEN is proposed, which uses PSO algorithm to optimal the ensemble weights instead of GA.

In the PSOSEN algorithm, a set of individual networks are trained firstly, and each neural network can be assigned a weight stochastically that could characterize the fitness of including this network in the ensemble. Then the weights of networks are optimized using PSO algorithm. Finally, the networks whose weights are bigger than a pre-set threshold could be selected to join the ensemble. The algorithm described as follows:

Inputs: training sets S, learner L, trials T, threshold λ.

TABLE I

COMPARISON OF TEST RESULTS BETWEEN GASEN AND PSOSEN(AE:ALL ENSEMBLE;SE:SELECTIVE ENSEMBLE)

	GASEN		PSOSEN	
Data Sets	AE	SE	AE	SE
Nasdaq-100	0.01931	0.01911	0.01874	0.01684
NIFTY	0.01574	0.01498	0.01453	0.01267

Output: the ensemble output which is the average output of the component networks.

Step1. Obtain training set of every individual neural network from the original training set S by bootstrap sampling;

Step2. Each neural network is trained with PSO algorithm;

Setp3. Generate a population of weight vectors randomly;

Step4. Optimize the weight of every network with PSO algorithm, and select several networks according to the pre-set threshold;

Step5. Combine the selected networks with simple averaging approach.

In each iteration step of the evolution, the weights are normalized so that they can compare with the pre-set threshold λ. PSOSEN uses a quite simple normalization scheme as follows:

$$w_i = w_i / \sum_{i=1}^{N} w_i \tag{3}$$

The fitness function used in this algorithm is the Root Mean Squared Error $(RMSE)$:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i(x) - y_i(x))^2} \tag{4}$$

Where X is the training set, N is the number of the samples, $\hat{y}(x)$ is the real output, and $y(x)$ is the expect output.

VI. EXPERIMENTS

We considered 7 year's stock data for Nasdaq-100 Index and 4 year's for NIFTY index. Our target is to predict the index value of the following trade day based on the opening, closing and maximum values of the same on a given day efficiently. We compared the test results of the GASEN algorithm and the PSOSEN algorithm on the two data sets, both methods used the same training and test data sets.

In our research, 10 neural networks were trained respectively. Each individual network is a feedforward neural network with one hidden layer. In order to distend the difference among individual networks, the number of the hidden layer nodes was respectively set as: 3, 4, 5, 6, 8, 10, 9, 11, 7, 12. The selected individual neural networks were combined with simple averaging. The test results comparison between GASEN algorithm and PSOSEN algorithm is shown in Table 1.

Test results of the PSOSEN for modeling Nasdaq-100 index is shown in Fig.2. and that for modeling NIFTY index is shown in Fig.3.

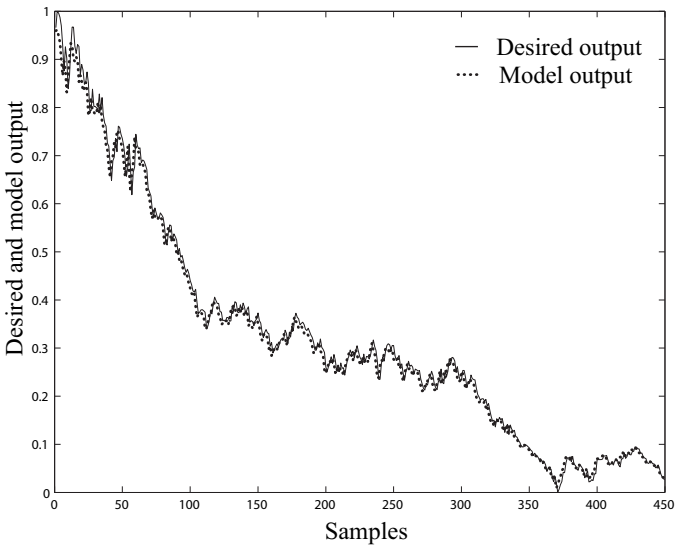

Fig. 2. Test results of the PSOSEN for modeling the Nasdaq-100 index

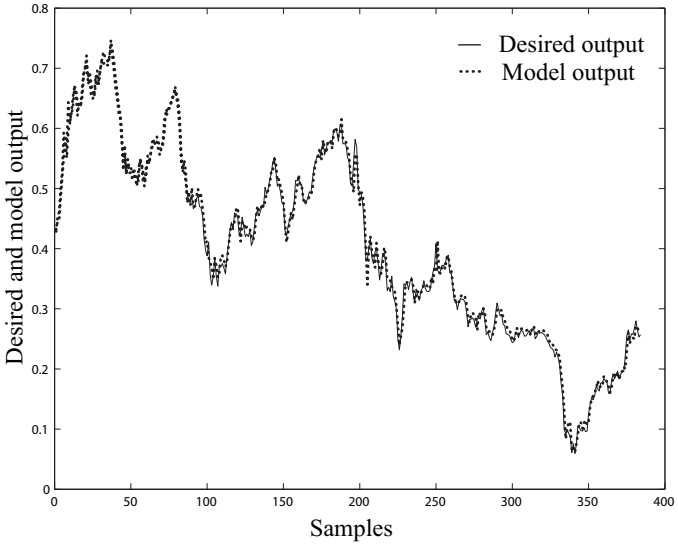

Fig. 3. Test results of the PSOSEN for modeling NIFTY index

VII. CONCLUSION

This paper presents a PSOSEN algorithm for stock index forecasting of two well-known stock indices namely Nasdaq-100 index of Nasdaq^{SM} and the S&P CNX NIFTY stock index. We compared the test results of the GASEN algorithm and the PSOSEN algorithm on the two data sets. It is shown from the test results that the PSOSEN algorithm is more accurate when compared with the GASEN algorithm. Moreover, the improved algorithm is faster than the GASEN algorithm. Since selective neural network ensemble is shown better in many fields, more rigorous testing on more complex problems will be performed in future works.

ACKNOWLEDGMENT

This research is supported by the National Science Foundation of China under grant No. 60573065 and the Key Subject Research Foundation of Shandong Province.

REFERENCES

[1] Y. H. Chen, A. Abraham, J. Yang and B. Yang. "Hybrid Methods for Stock Index Modeling," Fuzzy Systems and Knowledge Discovery: Second International Conference, Changsha, Vol. 3614, pp. 1067-1070, Aug. 2005.

[2] Nasdaq Stock Market^{SM}. [Online]. Available: http://www.nasdaq.com

[3] National Stock Exchange of India Limited. [Online]. Available: http://www.nse-india.com

[4] W. Leigh, M. Paz and R. Purvis. "An analysis of a hybrid neural network and pattern recognition technique for predicting short-term increases in the NYSE composite index," Omega, Vol. 30, pp. 69-76, 2002.

[5] S. H. Kim and S. H. Chun. "Graded forecasting using an array of bipolar predictions: application of probabilistic neural networks to a stock market index," Int. J. Forecast, Vol. 14, pp. 323-337, 1998.

[6] J. Van den Berg, U. Kaymak and W. M. Van den Bergh. "Financial markets analysis by using a probabilistic fuzzy modelling approach," Int. J. Approximate Reasoning, Vol. 35, pp. 291-305, 2004.

[7] Y. H. Chen, B. Yang and A. Abraham. "Flexible Neural Trees Ensemble for Stock Index Modeling", Neurocomputing, Vol. 70, pp. 697-703, Oct. 2007.

[8] Z. H. Zhou, J. X. Wu, W. Tang and Z. Q. Chen. "Selectively Ensembling Neural Classifiers," Proceedings of the International Joint Conference on Neural Networks, Honolulu, Vol. 2, pp. 1411-1415, 2002.

[9] Z. H. Zhou, J. X. Wu, Y. Jiang and S. F. Chen. "Genetic Algorithm based Selective Neural Network Ensemble," Proceedings of the 17th International Joint Conference on Artificial Intelligence, Vol. 2, pp. 797-802, 2001.

[10] Z. H. Zhou, J. X. Wu and W. Tang. "Ensembling Neural Networks: Many Could Be Better Than All," Artificial Intelligence, Vol. 137, pp. 239-263, 2002.

[11] K. Nygren. "Stock Prediction - A Neural Network Approach," Master Thesis, Royal Institute of Technology, KTH, Apr. 2004.

[12] H. X. Chen, S. M. Yuan and K. Jiang. "Selective neural network ensemble based on clustering," Lecture Notes in Computer Science, Springer Verlag, Heidelberg, Vol. 3971, pp. 545-550, 2006.

[13] J. Kennedy and R. Mendes. "Population structure and particle swarm performance. Proc," the Congress on Evolutionary Computation, NanJing: IEEE Press, pp. 1671-1676, 2002.

[14] K. W. Chau. "Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River," Journal of Hydrology, Elsevier, Amsterdam, Vol. 329, pp. 363-367, Oct. 2006.

[15] Q. L. Zhang, X. Li and Q. A. Tran. "A modified particle Swarm optimization algorithm," Machine Learning and Cybernetics, Institute of Electrical and Electronics Engineers Computer Society, Piscataway, pp. 2993-2995, 2005.

[16] J. Y. Chen, Z. Qin, Y. Liu and J. Lu. "Particle Swarm Optimization with Local Search," Neural Networks and Brain, Vol. 1, pp. 481-484, 2005.

[17] A. Carlisle and G. Dozier. "An Off-The-Shelf PSO," The Workshop on Particle Swarm Optimization, Indianapolis, 2001.

[18] M. S. Arumugam, A. Chandramohan and M. V. C. Rao. "Competitive Approaches to PSO Algorithms via New Acceleration Co-efficient Variant with Mutation Operators," International Conference on Computational Intelligence and Multimedia Applications, 2005.

[19] R. Hassan, B. Cohanim, O. Weck and G. Venter. "A COMPARISON OF PARTICLE SWARM OPTIMIZATION AND THE GENETIC ALGORITHM," AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials, American Institute of Aeronautics and Astronautics, Reston, Vol. 2, pp. 1138-1150, 2005.

[20] A.J.C. Sharkey (Ed.), "Combining Artificial Neural Nets", Springer, London, 1999.

[21] S. Geman, E. Bienenstock, R. Doursat, "Neural networks and the bias/variance dilemma", Neural Comput. 4, pp. 1-58, 1992.