วิศวกรรมสาร มข. ปีที่ 32 ฉบับที่ 4 (457 - 465) กรกฎาคม - สิงหาคม 2548
KKU Engineering Journal Vol. 32 No. 4 (457 - 465) July – August 2005

Committee Networks: Many Heads Are Better Than One

Danaipong Chetchotsak [1)]

[1)] Lecturer, Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002

Email: cdanai@kku.ac.th

ABSTRACT

Committee networks are based on the proverb that "many heads are better than one." The neural networks in a committee called the committee members which have different capabilities will help each other and work together to solve a given problem. Basically, the outputs of the committee members will be combined through a certain fusion rule to produce the output of the committee network. This paper introduces the concept of committee networks along with a single neural network and reviews some of their current related literature. Here, the methods to form a committee network were classified and reviewed. This includes but not limited to the error decorrelation, linear combinations of neural networks, stacked generalization, and mixtures of local experts. All published works suggested that the committee networks perform better than a single neural network. Because of their performance, we conclude that committee networks should be used as a tool in several different fields including industrial and manufacturing engineering and be used as a replacement of a single neural network. If the computation time is a major concern however, using committee networks would not be appropriate. Further research directions in both theoretical and application are also provided in this paper.

Keywords : committee networks, committee machines, combining neural networks, neural networks

Introduction

Committee networks or committee machines have received great attention in the neural network community for more than a decade. The main idea of committee networks is to combine the outputs of several neural networks, which contain different useful information, into a single predictor. This has been shown to be better than just choosing the best single network [Breiman. 1996; Hansen & Salamon; 1990; Hashem, 1997; Krogh & Vedelsby, 1995; Parmanto et al., 1996; Perrone & Cooper. 1993; and Wolpert. 1992]. Committee networks have been used as a tool in several areas such as electric load forecasts [Abdel-Aal. 2005], manufacturing applications [Marwala. 2001], signal analysis [Das et al. 2001], medical diagnosis [Übeyli and Güler. 2005], control [Melin and Castillo. 2003], and combustion analysis [Hao et al. 2001].

The objectives of this paper are two folds: first, to present the concept of committee networks and second, to review related current literature that develop methods to form a committee network. Since the materials presented in this paper may be new to some readers, the paper begins with the concept of neural networks and committee networks in the second section. Next, several methods to form the committee networks are presented. Finally, the last section provides discussion and conclusion.

Neural networks and Committee Networks

Neural networks are information processing and computational systems, which are superficially modeled after a biological nervous system. Like human brains, neural networks can process, learn, and remember information. Because of their high-speed computation and learning capability, neural networks can be trained to perform a difficult and tedious task. They are widely used in many areas including signal processing, control, medicine, forecasting, and industrial and business applications. For the applications in industrial engineering, the following topics are a good example: solving the lot-sizing problem [Gaafar & Choueiki. 2000], heuristic scheduling algorithms [Akyol. 2004], flaw detection in ultrasonic power spectra [Thomsen & Lund. 1991], predictive quality control [Joseph & Hamatty. 1993], and redundancy allocation problem [Coit & Smith. 1996].

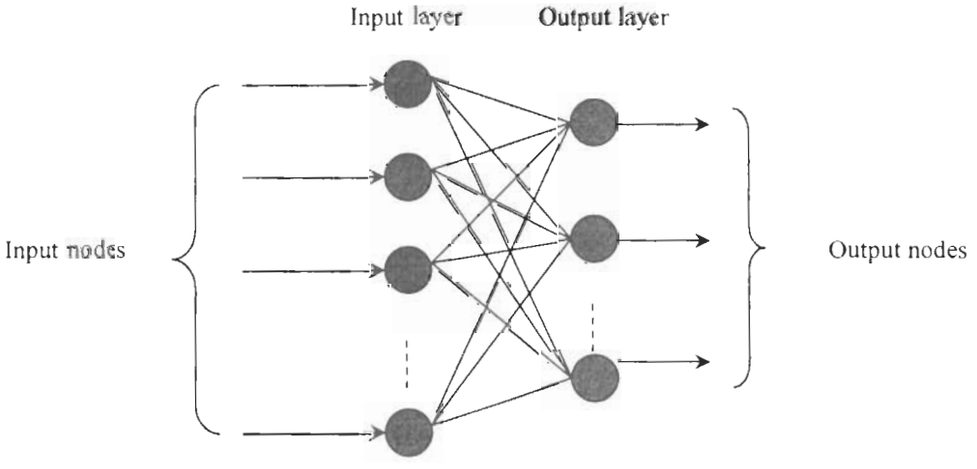

Figure 1 *A Simple Feedforward Neural Network*

Figure 1 show an example of neural networks called a feedforward neural network or multilayer perceptron (MLP). For this type of networks, the information flows in a forward direction, from the input to the output layers. The relationship between the inputs and the outputs is determined using the weights, connecting from one node to another. The weights are typically adjusted through a training process based on a given set of training data. Analogously, the weights of MLP are similar to the regression coefficients, which explain the relationship between the independent and dependent variables. Thus, using MLP for function approximation or regression problems is very common. For more detailed explanation and discussion of neural networks, the readers are referred to Haykin (1994).

Next, we explain the concept of committee networks. The main idea of committee networks is to combine several trained neural networks, which contain different useful information into a single model/predictor to solve the same problem. This concept is similar to that of the congress that several politicians who have different expertise and skills can work together and help one another in the congress to run the country. In addition, the proverb "many heads are better than one" might better describe such concept.

To build a committee network, neural networks in the committee (committee members) can be either trained using the same data set or different subsets of data. The outputs of these committee members are normally combined using a fuser (combiner), which is established from a certain rule such as the simple average and weighted average as depicted in Figure 2.

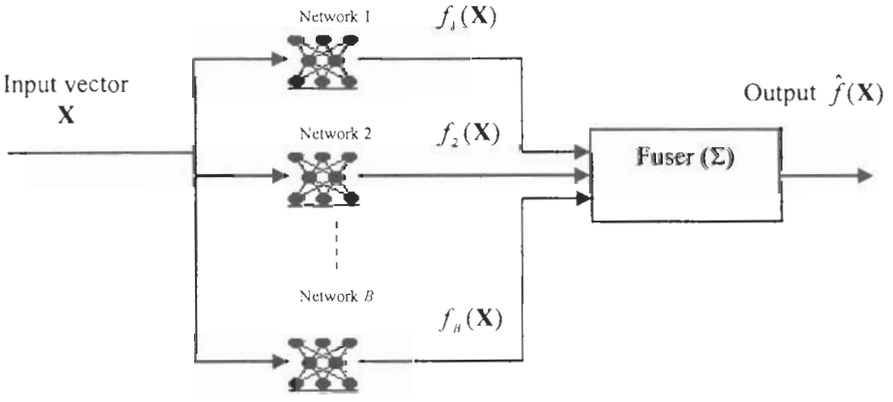

Figure 2 *Block Diagram of a Committee Network*

Some examples of the existing approaches for constructing committee networks are the error decorrelation, linear combinations of neural networks, stacked generalization, and mixtures of local experts. The linear combinations of neural networks approach mainly concerns how the committee members should be combined, while the other approaches address how the committee should be trained. These approaches are described as below.

The Error Decorrelation

In this approach, the committee members are trained to be uncorrelated with each other. When these members are uncorrelated, they will tend to make a mistake at different places, and combining these committee members tends to cancel out the mistake made by each individual

member and thus improves the generalization performance. Mathematical explanation of the error decorrelation approach is provided in Krogh & Vedelsy (1995) and Raviv & Intrator (1996).

Examples of the early works that use the error decorrelation approach are Krogh & Vedelsby (1995), Wolpert (1992), Parmanto et al. (1996), Twomey & Smith (1995), Lam (1999), Siriphala (2000), Tumer & Ghosh (1996), Raviv, & Intrator (1996). In general, they used the bootstrap and cross validation methods to decorrelate the training set. Their method is referred to as committees by resampling. Lam (1999) and Siriphala (2000) showed that the committee based on the bootstrap method (the bootstrap committee) has good generalization and outperforms the committees based on other resampling methods under sparse data conditions. Other works also showed that the committee networks based on the error decorrelation outperform a single neural network. The most recent works on the error decorreation can be seen in Grandvalet (2004), Sohn & Dagli (2004), Christensen et al. (2003), and Brown et al. (2005).

i. Let $\hat{F}$ be the empirical probability distribution where $\mathbf{T}$ with n observations is drawn.

ii. Let $t_1, t_2, ..., t_n$ be a collection of training set $\mathbf{T}$ where $t_i = (\mathbf{x}_i, y_i)$.

iii. Specify the number of bootstrap samples, B. This produces $\mathbf{T}^{*1}, \mathbf{T}^{*2}, ..., \mathbf{T}^{*j}, ..., \mathbf{T}^{*B}$

iv. For each of $\mathbf{T}^{*j}$, t_i is randomly chosen form $\mathbf{T}$ for $i = 1,....n$ with replacement and equal probability $\frac{1}{n}$.

v. Step iv is repeated B times for $j = 1,...., B$.

vi. Bootstrap training sets $\mathbf{T}^{*1}, \mathbf{T}^{*2}, ..., \mathbf{T}^{*B}$ are generated.

vii. Train Network 1, Network 2,..., Network B using $\mathbf{T}^{*1}, \mathbf{T}^{*2}, ..., \mathbf{T}^{*B}$. Given the input $\mathbf{x}_l, l = 1,..., n$ these networks produce the outputs $f_1(\mathbf{T}^{*1}, \mathbf{x}_l), f_2(\mathbf{T}^{*2}, \mathbf{x}_l), ...,$ $f_B(\mathbf{T}^{*B}, \mathbf{x}_l)$.

viii. If the simple average (SAVG) is used as a combiner, the committee's output is written as

$$\hat{f}(\mathbf{x}_l) = \frac{1}{B}\sum_{i=1}^{B} f_i(\mathbf{T}^{*i}, \mathbf{x}_l), \quad l = 1,..., n.$$

Table 1 *The Bootstrap Algorithm*

Table 1 describes the bootstrap committee algorithm. Following the bootstrap algorithm, steps i – vii produce $f_1(\mathbf{T}^{*1}, \mathbf{X})$, $f_2(\mathbf{T}^{*2}, \mathbf{X}), ..., f_B(\mathbf{T}^{*B}, \mathbf{X})$. Such notations are referred to as committee members' outputs, where each member is trained using the training data $\mathbf{T}^{*i}$ and gives the output based on the input $\mathbf{X}$. The outputs from these committee members may be fused using the simple average method, or other fusion rules such as linear regression, ridge regression, or principal component regression as described the next section.

Linear Combinations of Neural Networks

This approach is also called the weighted average. A committee is formed using a linear combination of the outputs of several networks as below:

$$\hat{f}(\mathbf{X}) = \sum_{i=1}^{m} \alpha_i f_i(\mathbf{X}), \quad (1)$$

where $f_i(\mathbf{X})$ is the output from committee member i, and α_i is the corresponding combination-weight, $i = 1,2,....,m$. The most concern here is how to choose the weights α such that the committee performance is maximized. One possibility is to use the simple average (SAVG):

$$\hat{f}(\mathbf{X}) = \frac{1}{m} \sum_{i=1}^{m} f_i(\mathbf{X}). \quad (2)$$

This method is simple but might ignore useful and unique contributions from good committee members. Another way to find the weights α is to use the least square method. Here, we run a regression model of $\mathbf{y}$ on $f_i(\mathbf{X})$, i.e., $f_i(\mathbf{X})$ are the independent variables and $\mathbf{y}$ serves as the dependent variable in a regression model. The committee's output is then written as

$$\hat{f}(\mathbf{X}) = \alpha_0 + \sum_{i=1}^{m} \alpha_i f_i(\mathbf{X}). \quad (3)$$

It should be noted that this method will minimize the mean square error of the committee. However, multicollinearity effects will arise since each committee member is trained to approximate the same problem. Multicollinearity is referred to as the phenomenon in which two or more independent variables are correlated among themselves. Hashem (1997) suggests that it could cause the committee to produce unreliable predictions. The following works introduced methods to deal with multicollinearity in the committee's model. Perrone & Cooper (1993) estimated the combination-weights using the least squares regression with the constraints $\sum \alpha_i = 1$ and $\alpha_0 = 0$. This is known as Generalized Ensemble Method (GEM). Merz & Pazzani (1999) and Chetchotsak (2003) used principal component regression (PCR) and ridge regression (RR) to determine the combination-weights. Chetchotsak (2003) introduced the r-k class estimator (a combination of RR and PCR) to find the weights α. All of these methods were shown to perform well since multicollinearity effects had been removed.

Stacked Generalization

Introduced by Wolpert (1992), stacked generalization uses the partition of available data to combine a set of generalizers or committee members (e.g., prediction models or neural networks) in a nonlinear manner. The fundamental procedures are shown in Tables 2-3.

i)	The training data T with n observations is partitioned using leave-one-out cross validation >> produces $\left((\mathbf{x}_1, y_1)\ (\mathbf{x}_2, y_2)\ \ldots\ (\mathbf{x}_{n-1}, y_{n-1})\right)$.
ii)	Then committee members $G_1, G_2, ..., G_k$ are trained using the partitioned data set.
iii)	Let $g_{1,n}, g_{2,n}, ..., g_{k,n}$ be the prediction values of $G_1, G_2, ..., G_k$ when they predict the leftover data point $(\mathbf{x}_n, y_n)$ >> these prediction values are used as a new training data set of the committee.

Table 2 *Stacked Generalization Algorithm*

iv) Let $\mathbf{s}_1$ be a row vector that consist of $[g_{1,n}\ g_{2,n} \ldots g_{k,n}]$. Here, $\mathbf{s}_1$ is the first data point of the new data set.

v) A different partition of $\mathbf{T}$ (e.g., $((x_1, y_1)\ (x_2, y_2,), \cdots, (x_{n-2}, y_{n-2}), \cdots, (x_n, y_n))$ is used as another training set and the leftover data point $(\mathbf{x}_{n-1}, y_{n-1})$ is used as a test set for $G_1, G_2, \ldots, G_k$. $G_1, G_2, \ldots, G_k$ will generate $\mathbf{s}_2 = [g_{1,n-1}\ g_{2,n-1} \ \cdots \ g_{k,n-1}]$. This process is repeated for all possible partitions of $\mathbf{T}$. The new data set to train the committee is written as

$$\mathbf{S} = \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_n \end{pmatrix} = \begin{pmatrix} g_{1,n} & g_{2,n} & \cdots & g_{k,n} \\ g_{1,n-1} & g_{2,n-1} & \cdots & g_{k,n-1} \\ \vdots & \vdots & \vdots & \vdots \\ g_{1,1} & g_{2,1} & \cdots & g_{k,1} \end{pmatrix}$$

vi) Once $\mathbf{S}$ is generated, another prediction model or neural network is used to map the relationship between $\mathbf{S}$ and $\mathbf{y}^*$, where $\mathbf{y}^* = (y_n\ y_{n-1} \ \ldots \ y_1)$.

Table 3 *Stacked Generalization Algorithm (cont'd)*

Mixtures of local experts

Mixtures of local experts are introduced by Jacob *et al.* [1991a&b]. This approach is based on the motivation that a complex task can be divided into several sets of simpler tasks as illustrated in Figure 3.

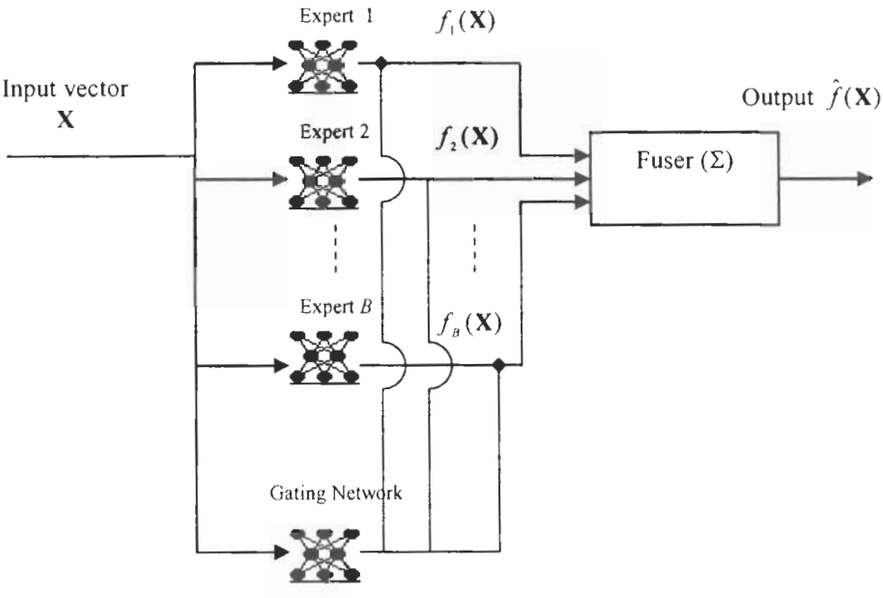

Figure 3 *Mixtures of Experts*

Each network or committee member acts as a local expert and is forced to learn a different region of data. Given a new data set, another network, called the gating network acts as a supervisor to allocate the new data to one or more experts that have learned the training data in the region close to the new data. The gating network also combines the outputs from the committee members to produce the committee's output. This allows the mixtures of experts to perform better than a single local expert.

Discussion and Conclusion

Committee networks rely on the concept that "many heads are better than one." Several neural networks in the committee which have different expertise will help one another to perform a task. This happens though the combining or forming process of the committee networks. This paper described the concept of committee networks and reviewed current related literature on the methods of committee formation to improve their performance. The methods classified here include but not limited to the error decorrelation, linear combinations of neural networks, stacked generalization, and mixtures of local experts. All the methods have shown that the committee networks outperform a single neural network. Thus, they may be applied in several different fields in engineering and other and be used as a replacement of a single neural network. However, if the computation time becomes a major constraint, committee networks may not be a good option. The computation mainly involves the training process of the committee members, which is the most time-consuming and the process of committee formation.

From the author's point of view, there are extensive research directions on committee networks in both theoretical and application. We have found that none of the methods mentioned before has been compared against each other in a methodical way. The committee networks' performance should be tested at different conditions such as the sizes of training sets, network complexity, and levels of noise in the data. Moreover, none of these methods has demonstrated their performance to be "the automatic smoothing method" where this term is used by Geman et al. (1992) to describe the properties of a model which automatically performs robustly in any kind of data and training conditions. Furthermore, we believe that committee networks should be widely used as a tool in several applications in industrial and manufacturing engineering such as predictive quality control, group technology, prediction of material properties, product classification, and flaw detection in products

Acknowledgement

The author would like to acknowledge the grace of God through this paper. All wisdoms are His. [Daniel 2: 20]. Also thanks be to Dr. Janet M. Twomey, the author's advisor, for her support and guidance during the time at Wichita State University.

References

Abdel-Aal, R.E. 2005. Improving Electric Load Forecasts using Network Committees. **Electric Power Systems Research**. 74 ,1: 83-94.

Akyol, D. E. 2004. Application of Neural Networks to Heuristic Scheduling Algorithms. **Computers & Industrial Engineering**. 46: 679–696.

Breiman, L. 1996. Stacked Regressions. **Machine Learning**. 24(1): 49-64.

Brown, G., Wyatt, J. and Tino, P. 2005. Managing Diversity in Regression Esembles. **Journal of Machine Learning Research**.

Chetchotsak, D. 2003. **Improving Generalization Capability of Neural Networks under Conditions of Sparse Data: A New Committee Formation Approach**. Ph.D. Thesis. Department of Industrial and Manufacturing Engineering. Wichita State University. Wichita, KS.

Christensen, S. W., Sinclair, I. and Reed, P. A.S. 2003. Designing Committees of Models through Deliberate Weighting of Data Points. **Journal of Machine Learning Research**. 4: 39-66.

Coit, D. W., Jackson, B.T. and Smith, A. E. 1998. Static Neural network Process Models: Considerations and Case Studies. **International Journal of Production Research**. 36,11: 2953-2967.

Das, A., Reddy, N. P. and Narayanan, J. 2001. Hybrid Fuzzy Logic Committee Neural Networks for Recognition of Swallow Acceleration Signals. **Computer Methods and Programs in Biomedicine**. 64,2: 87-99.

Gaafar, L. K., & Choueiki, M. H. 2000. A Neural Network Model for Solving the Lot-Sizing Problem. **Omega**. 28: 175-184.

Grandvalet, Y 2004. Bagging Equalizes Influence. **Machine Learning**. 55,3: 251 – 270.

Hansen, L.K. and Salamon, P. 1990. Neural Network Ensembles. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. 12: 993-1001.

Hao, Z., Kefa, C. and Jianbo, M. 2001. Combining Neural Network and Genetic Algorithms to Optimize Low NOx Pulverized Coal Combustion. **Fuel. 80,15: 2163-2169.**

Hashem, S. 1997. Optimal Linear Combinations of Neural Networks. **Neural Networks**. 10:4, 599-614.

Haykin, S. 1994. **Neural Networks: A Comprehensive Foundation**. New York, NY: McMillan College Publishing,

Jacob, R. A., Jordan, M. I., and Barto, A. G., 1991a. Task Decomposition through Competition in A Modular Connectionist Architecture: The What and Where Vision Tasks. **Cognitive Science**. 15: 219-250.

Jacob, R. A., Jordan, M.I., Nowlan, S. J., and Hilton, G. E. 1991b. Adaptive Mixture of Local Experts. **Neural Computation**. 3,1: 79-87.

Joseph, B. and Hamatty, W. 1993. Predictive Control of Quality in a Batch Manufacturing Process using Artificial Neural Networks Models. **Ind Eng. Chem. Res.** 32: 1951-1961.

Krogh, A. and Vedelsby, J. 1995. Neural Network Ensembles, Cross Validation, and Active Learning. **Advance in Neural Information Processing Systems**. 7: 1-8.

Lam, S. S.Y. 1999. **Improved Prediction and Validation Using Resampled Neural Networks: Committee Networks and Hybrid Validation**. Ph.D. Thesis, Department of Industrial Engineering, University of Pittsburgh.

Marwala, T. 2001. Scaled Conjugate Gradient and Bayesian Training of Neural Networks for Fault Identification in Cylinders. **Computers & Structures**. 79,32: 2793-2803.

Melin, P. and Castillo, O. 2003. Adaptive Intelligent Control of Aircraft Systems with a Hybrid Approach Combining Neural Networks, Fuzzy Logic and Fractal Theory. **Applied Soft Computing**. 3,4: 353-362.

Merz, C. J. and Pazzani, M. J. 1999. A Principal Components Approach to Combining Regression Estimates. **Machine Learning**. 36,1-2, 9-32.

Parmanto, B., Munro, P.W., and Doyle, H.R. 1996. Reducing Variance of Committee Prediction with Resampling Techniques. **Connection Science**. 8,3 & 4: 405-425.

Perrone, M. P. 1993. **Improving Regression Estimation: Averaging Methods for Variance Reduction with Extension to General Convex Measure Optimization**. PhD Thesis, Department of Physics, Brown University, Providence, RI.

Perrone, M.P. and Cooper, L.N. 1993. When Neural Networks Disagree: Ensemble Methods for Hybrid Neural Networks In R.J. Mammone (de.) **Artificial Neural Networks for Speech and Vision**.: 126-142.

Raviv, Y. and Intrator, N. 1996. Bootstrapping with Noise: An Alternative Regularization Technique. **Connection Science**. 8,3-4: 355-372.

Siriphala, P. 2000. **Controlling Artificial Neural Networks Overtraining When Data is Scarce**. Ph.D. Thesis. Department of Industrial and Manufacturing Engineering. Wichita State University.

Sohn, S. and Dagli, C. H. 2004. Ensemble of Evolving Neural Networks in Classification. **Neural Processing Letters**. 19,3: 191 – 203.

Thomsen, J.J. and Lund, K. 1991. Quality Control of Composite Materials by Neural Network Analysis of Ultrasonic Power Spectra. **Mater. Eval**. 594-600.

Tumer, K. and Ghosh, J. 1996. Error Correlation and Error Reduction in Ensemble Classifiers. **Connection Science**. 8,3 & 4: 385-404.

Twomey, J. M. and Smith, A. E. 1995. Committee Networks by Resampling. **Intelligent Neural Engineering Systems Through Artificial Networks**. 6: 889-894.

Übeyli, E. D. and Güler, I. 2005. Improving Medical Diagnostic Accuracy of Ultrasound Doppler Signals by Combining Neural Network Models. **Computers in Biology and Medicine**. 35,6: 533-554.

Wolpert, D. H. 1992. Stacked Generalization. **Neural Computation**. 5: 241-259.