Complementary Neural Networks For Regression Problems

Pawalai Kraipeerapun*, Sathit Nakkrasae*, Somkid Amornsamankul†‡ and Chun Che Fung§
*Department of Computer Science, Faculty of Science, Ramkhamhaeng University, Bangkok, Thailand
Email: pawalai@ru.ac.th, sathit@ru.ac.th
†Department of Mathematics, Faculty of Science, Mahidol University, Bangkok, Thailand
‡Centre of Excellence in Mathematics, CHE, Sriayudhaya Rd., Bangkok, Thailand
Email: scsam@mahidol.ac.th
§School of Information Technology, Murdoch University, Perth, Australia
Email: l.fung@murdoch.edu.au

***Abstract*— In this paper, complementary neural networks (CMTNN) are used to solve the regression problem. CMTNN consist of a pair of opposite neural networks. The first neural network is trained to predict degree of truth values and the second neural network is trained to predict degree of falsity values. Both neural networks are complementary to each other since they deal with pairs of complementary output values. In order to predict the more accurate outputs, each pair of the truth and falsity values are aggregated based on two techniques which are equal weight combination and dynamic weight combination. The first technique is just a simple averaging whereas the second technique deals with errors occurred in the prediction. We experiment our approach to the classical benchmark problems including housing, concrete compressive strength, and computer hardware from the UCI machine learning repository. It is found that complementary neural networks improve the prediction performance as compared to the traditional single backpropagation neural network and support vector regression used to predict only truth values. Furthermore, the difference between the predicted truth value and the complement of the predicted falsity value can be used as an uncertainty indicator to support the confidence in the prediction of unknown input data.**

I. Introduction

In general, a supervised neural network can be used to solve classification and regression problems. In the classification problem, each output obtained from a neural network is assigned to one of a number of classes whereas each output represents continuous value for the regression problem. This paper deals with the continuous output of neural networks.

The recent review from Paliwal and Kumar [1] showed that feed forward neural networks were found to give better classification and prediction results than the traditional statistical methods in various areas of applications such as accounting and finance, health and medicine, engineering and manufacturing, marketing, and other general applications. Furthermore, several researches have also tried to compare accuracy performance between neural networks and support vector machines (SVM), which is one of the most popular methods used for classification and regression problems. In [2], Osowski et al. argued that multilayer perceptron (MLP) is much better than SVM when applied to the regression task; however, SVM provides better accuracy results in the classification problem. Their comparative study was experimented based on two-spiral classification problem, Mackey-Glass time series prediction in chaotic mode, and the artificial nose regression problem. In [3], Msiza et al. observed that artificial neural networks perform significantly better than support vector regressions in water demand prediction. In [4], Meyer et al. concluded that neural networks, projection pursuit regression and random forests often yielded better results than support vector machine for regression tasks. Their experiment on the regression task was based on 12 data sets from the UCI machine learning database, the DELVE project, and the SLID data.

In recent years, several types of neural network methodology have been proposed by scientists in order to solve the regression problem. For example, Ma and Khorasani [5] proposed incremental constructive training schemes for one-hidden-layer feed-forward neural network. The error signal is scaled during the constructive learning process to improve the input-side training efficiency. Altun et al. [6] proposed a data treatment technique which modifies the distribution characteristics of the data set in order to improve the accuracy of a neural network estimator. They realized their technique to the suspended sediment prediction. Setiono and Thong [7] proposed rules extraction from neural networks trained to solve regression problems. The extracted rules separate the input data into groups in which a linear function of the relevant input attributes within a group estimates the network output. In [8], Chetwynd et al. claimed that applying an interval-valued neural network allows a trade-off between the model error and the interval width of the network weights, which is considered as a degree of uncertainty parameter. They used uncertainty to create a robust regression algorithm. In [9], Shen and Kong applied Generalized Regression Neural Networks (GRNN) to predict uncertainty of type error from known errors obtained from training ensemble neural networks. After that these predicted errors were used in the dynamic aggregation process in the determination results from the ensemble of neural networks.

In our previous papers [10], [11], we have applied complementary neural networks (CMTNN) to solve the problem of

978-1-4244-3703-0/09/$25.00 ©2009 IEEE

binary and multiclass classification. A pair of neural networks were created to predict degrees of truth and falsity values. The predicted truth and falsity values were then combined in order to provide the classification results. In this paper, we aim to apply the CMTNN and the combination techniques to the regression problem. In the classification problem, uncertainty in the prediction can be quantified based on the difference between the truth and falsity values [10]. However, this uncertainty quantification technique cannot be used in the regression problem since the target outputs used to train the truth and falsity neural networks are not fixed to 0 and 1. Hence, a novel technique used to quantify uncertainty in the prediction of unknown data for the regression problem is then proposed. In this work, uncertainty is quantified based on the difference between the truth and non-falsity values. We experiment our approach to the classical benchmark problems including housing, concrete compressive strength, and computer hardware from the UCI machine learning repository.

The rest of this paper is organized as follows. Section II explains the use of complementary neural networks, the quantification of uncertainty, and the combination techniques used for the regression problem. Section III describes the data set and results of our experiments. Conclusions and future work are presented in Section IV.

II. Complementary Neural Networks

In general, neural networks are used to predict only the true values. However, the prediction result may not completely true. It can be considered as degrees of truth. Instead of considering only degrees of truth, we also consider degrees of falsity. The truth table for implication can be applied to the concept of neural network. The implication "if A then B" or "$A \rightarrow B$" can be considered. It is known that if A is true and B is true then this implication rule is true. We also know that if A is true and B is false then this implication rule is false. These implication rules can be shown in table I.

TABLE I
Logical implication

Premise A	Conclusion B	Inference $A \rightarrow B$
True	True	True
True	False	False

We apply these two implication rules to the concept of neural network. Let A be the input feature vector and let B be the target output value. In traditional neural network, both A and B are set to true. They are used to train neural network to predict degree of truth values. In contrast, if we apply the second implication rule to the concept of neural network then the output of neural network should provide degree of falsity values. In order to apply the second rule, A must be true and B must be false. This implication can be written as "$A \rightarrow\sim B$". In order to set B to false, the complement of target output values are then used.

Figure 1 shows the complementary neural networks (CMTNN) model for the regression problem. Two opposite

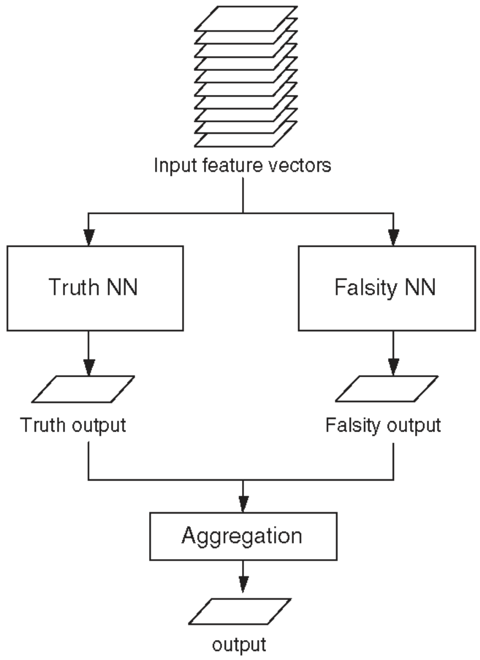

Fig. 1. The complementary neural networks model for continuous output

neural networks are trained to predict degree of truth and degree of falsity values. The truth neural network (Truth NN) is trained to predict degree of truth values. The falsity neural network (Falsity NN) is trained to predict degree of falsity values. The falsity neural network has the same architecture as the truth neural network; however, this network is trained to predict degree of falsity values using the complement of target outputs used in the training data of the truth neural network. Both truth and falsity neural network also apply the same training input data. Errors occurred in the prediction of both truth and falsity values are also quantified and will be used to estimate errors occurred in the prediction of unknown input data.

In the testing phase, the predicted truth and falsity values obtained from the truth and falsity neural networks are combined. Let $T(x_i)$ and $F(x_i)$ be the truth and falsity values predicted from the input pattern $x_i; i = 1, 2, 3, ..., n$ where n is the total number of input patterns in the testing phase. Two combination techniques can be described below.

- Equal weight combination
  The truth value and the complement of the falsity value are combined in this technique. The combined output $O(x_i)$ can be computed as follows:

$$O(x_i) = \frac{T(x_i) + (1 - F(x_i))}{2}. \quad (1)$$

- Dynamic weight combination
  In this technique, errors are quantified and used to weight the combination between the truth and complement of the

falsity values. The multidimensional interpolation method is used to estimate those errors [10]. Errors occurred in the testing phase can be estimated from known errors occurred in the training phase. Let $E_t(x_i)$ and $E_f(x_i)$ be errors estimated from the truth and falsity neural networks, respectively. The error of the falsity value is considered to be equal to the error of the non-falsity value, which is the complement of the falsity value. The dynamic combination output $O(x_i)$ can be calculated as follows:

$$O(x_i) = (W_t(x_i) \cdot T(x_i)) + (W_f(x_i) \cdot (1 - F(x_i))), \quad (2)$$

where

$$W_t(x_i) = \frac{1-E_t(x_i)}{(1-E_t(x_i))+(1-E_f(x_i))},$$

$$W_f(x_i) = \frac{1-E_f(x_i)}{(1-E_t(x_i))+(1-E_f(x_i))}.$$

The next section will show that the combined output obtained from these two techniques for the regression problem provide better performance when compared to the output obtained from a single neural network and a single support vector machine.

In general, several neural networks are created based on different parameters in order to find the most suitable neural network model for individual applications. In this paper, it is considered that no matter which model is selected, there still exists uncertainty in the prediction. We cannot avoid uncertainty. Hence, a technique used to quantify uncertainty in neural network regression is then proposed. Uncertainty in the prediction of unknown input data can be quantified using the relationship between the predicted truth and the predicted non-falsity values. If the difference between both values is high then the uncertainty is high. On the other hand, if the difference is low then the uncertainty is low. The relationship between the truth and non-falsity values can be drawn in two dimensional graphical representation. If a cluster of points is arranged in the diagonal right then the uncertainty is low since both truth and non-falsity values are very close to each other.

III. Experiments

A. Data Set

Three data sets from UCI Repository of machine learning [12] are employed. These three data sets are Housing, Concrete Compressive Strength [13], and Computer Hardware. Table II shows the characteristics of these three data sets.

TABLE II

DATA SETS USED IN THIS STUDY

Name	Feature type	No. of features	No. of samples
Housing	numeric	13	506
Concrete	numeric	8	1030
Hardware	numeric	6	209

B. Experimental Methodology and Results

Ten-fold cross validation method is performed on each data set. This paper does not focus on the optimization of the individual predictors but concentrate only on the improvement of the combined prediction. We add more works on this ten-fold cross validation method. Twenty pairs of feed-forward backpropagation neural networks (BPNN) trained with twenty different randomized training sets are created for each fold. The mean square error (MSE) obtained from twenty runs are averaged for each fold and shown in table III, IV, and V for housing, concrete, and computer data sets, respectively. Each pair of feed-forward backpropagation neural networks is trained to predict degree of truth and falsity values. For each data set, all neural networks have the same parameter values in terms of the network architecture. The number of input-nodes for each network is equal to the number of input features for each training set. Each network has one hidden layer constituting of $2n$ neurons where n is the number of input features. The only difference between each pair of networks is that the target outputs of the falsity network are equal to the complement of the target outputs used to train the truth network.

Table III, IV, and V show the average MSE obtained from a single SVM with linear kernel, a single SVM with polynomial kernel, a single SVM with radial basis function (RBF) kernel, a single BPNN (truth value), a single BPNN (falsity value), CMTNN based on equal weight combination, and CMTNN based on dynamic weight combination. SVM is run using mySVM [14] and feed-forward backpropagation neural network is run using Matlab [15]. The percent improvement between our CMTNN techniques and the existing techniques are compared and shown in table VI for housing, concrete, and computer data sets. It can be noticed that the traditional BPNN provides better accuracy performance than the traditional SVM with linear, polynomial, and RBF kernels for all three data sets. Also, it is found that our techniques provide better accuracy than the traditional SVM and BPNN. For CMTNN, the dynamic weight combination technique is found to provide better accuracy than the equal weight combination; however, there is not much difference between both techniques for the regression problem.

Furthermore, our techniques also have the advantage of providing a measure of the uncertainty in neural network regression. The relationship between the truth value ($T(x_i)$) and the non-falsity value ($1 - F(x_i)$) can be used to identify level of uncertainty in neural network regression model. The truth and non-falsity values can be drawn in two dimensional spaces. If a cluster of points is arranged in the diagonal right then the neural network model contains a very low uncertainty since both truth and non-falsity values are very close to each other. Otherwise, it contains higher uncertainty level. From our experiment, we found that uncertainty level obtained from our technique corresponds with the MSE obtained from the predicted and target values. Therefore, we can use our uncertainty quantification technique to identify uncertainty in

TABLE III
AVERAGE OF MEAN SQUARE ERROR (20 RUNS) FOR EACH FOLD OBTAINED FROM THE TEST SET OF HOUSING DATA.

Fold	SVM linear	SVM poly	SVM RBF	BPNN Truth value	BPNN Falsity value	CMTNN Eq. weight	CMTNN Dy. weight
1	0.025405	0.038817	0.033243	0.008447	0.006544	0.006115	0.006110
2	0.014315	0.012587	0.013667	0.005080	0.004649	0.004166	0.004166
3	0.028978	0.030019	0.039770	0.007620	0.009660	0.005929	0.005942
4	0.046678	0.050903	0.055616	0.026762	0.021510	0.020417	0.020395
5	0.032601	0.027560	0.036150	0.009931	0.010641	0.007822	0.007809
6	0.040773	0.043828	0.057948	0.009362	0.013838	0.006802	0.006745
7	0.014638	0.015503	0.016702	0.003625	0.004369	0.003321	0.003323
8	0.088269	0.089842	0.098378	0.044720	0.044788	0.040896	0.040843
9	0.089771	0.079819	0.100993	0.017187	0.015392	0.011275	0.011262
10	0.034173	0.048337	0.037999	0.025034	0.030735	0.024222	0.024186
Avg	0.041560	0.043721	0.049047	0.015777	0.016213	0.013097	0.013078

TABLE IV
AVERAGE OF MEAN SQUARE ERROR (20 RUNS) FOR EACH FOLD OBTAINED FROM THE TEST SET OF CONCRETE COMPRESSIVE STRENGTH DATA.

Fold	SVM linear	SVM poly	SVM RBF	BPNN Truth value	BPNN Falsity value	CMTNN Eq. weight	CMTNN Dy. weight
1	0.025150	0.024118	0.020764	0.026207	0.026541	0.022063	0.022075
2	0.076913	0.055314	0.064095	0.011691	0.013548	0.010752	0.010752
3	0.036451	0.035658	0.034371	0.005998	0.006017	0.005167	0.005173
4	0.035906	0.032145	0.038106	0.020877	0.018750	0.017793	0.017793
5	0.027747	0.021795	0.027839	0.013177	0.014644	0.012170	0.012167
6	0.068953	0.063243	0.054821	0.008954	0.009796	0.007902	0.007897
7	0.074684	0.072569	0.073884	0.004220	0.003781	0.003102	0.003103
8	0.048336	0.051942	0.048473	0.009622	0.013660	0.008489	0.008478
9	0.034691	0.034116	0.029740	0.007668	0.007551	0.006420	0.006399
10	0.033530	0.033761	0.030129	0.006897	0.006892	0.005912	0.005913
Avg	0.046236	0.042466	0.042222	0.011531	0.012118	0.009977	0.009975

TABLE V
AVERAGE OF MEAN SQUARE ERROR (20 RUNS) FOR EACH FOLD OBTAINED FROM THE TEST SET OF COMPUTER HARDWARE DATA.

Fold	SVM linear	SVM poly	SVM RBF	BPNN Truth value	BPNN Falsity value	CMTNN Eq. weight	CMTNN Dy. weight
1	0.052892	0.041773	0.058055	0.029891	0.020777	0.01883	0.018778
2	0.006530	0.006991	0.006106	0.004793	0.004519	0.004519	0.004515
3	0.004918	0.005802	0.005121	0.000488	0.000516	0.000444	0.000442
4	0.004544	0.004411	0.004806	0.001229	0.001253	0.001145	0.001142
5	0.011292	0.011604	0.011869	0.002865	0.003491	0.002241	0.002204
6	0.005267	0.005469	0.005902	0.000740	0.000417	0.000415	0.000411
7	0.003442	0.003199	0.003761	0.001266	0.000693	0.000834	0.000826
8	0.014943	0.017509	0.016882	0.007960	0.007551	0.004456	0.004290
9	0.003857	0.003834	0.004265	0.003328	0.003634	0.002941	0.002943
10	0.070298	0.089493	0.075745	0.067446	0.029364	0.027644	0.025869
Avg	0.017798	0.019009	0.019251	0.0120006	0.0072215	0.0063469	0.006142

TABLE VI
THE PERCENT IMPROVEMENT OF OUR TECHNIQUES COMPARED TO THE TRADITIONAL TECHNIQUES WHICH ARE BPNN AND SVM WITH LINEAR, POLYNOMIAL, AND RBF KERNELS.

Data set	Technique		SVM linear	SVM poly	SVM RBF	BPNN Truth value	BPNN Falsity value
Housing	CMTNN	eq. weight	68.49%	70.05%	73.30%	16.99%	19.22%
		dy. weight	68.53%	70.09%	73.34%	17.11%	19.33%
Concrete	CMTNN	eq. weight	78.42%	76.51%	76.37%	13.48%	17.67%
		dy. weight	78.43%	76.51%	76.38%	13.49%	17.68%
Computer	CMTNN	eq. weight	64.34%	66.61%	67.03%	47.11%	12.11%
		dy. weight	65.49%	67.69%	68.10%	48.82%	14.95%

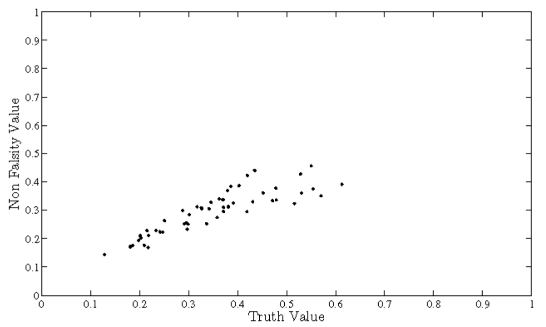

Fig. 2. The relationship between the truth and non-falsity values obtained from CMTNN that provides the results with the MSE of 0.005235 from the test set of Housing data.

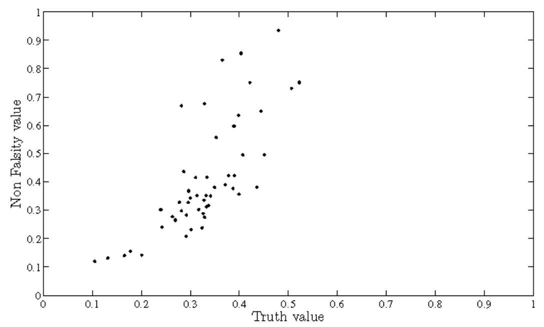

Fig. 3. The relationship between the truth and non-falsity values obtained from CMTNN that provides the results with the MSE of 0.023342 from the test set of Housing data.

the prediction of unknown data.

Figure 2, 3, and 4 show the relationship between the truth and non-falsity values obtained from three pairs of complementary neural networks. They are obtained from the test set of housing data. These three sets are picked from twenty CMTNN created in fold 10. They provide the results with the MSE of 0.005235, 0.023342, and 0.042374, respectively. It can be noticed that a cluster of points obtained from each CMTNN is corresponding to the MSE computed from the actual results and the predicted results. The average distance between points and the diagonal right shown in these three figures are 0.04063, 0.073918, and 0.130026, respectively. These three average distance values are corresponding to those three MSE. A cluster of points shown in figure 2 is arranged in the diagonal right and has a better shape when it is compared to a cluster of points shown in figure 3 and 4. Also, it provides the best MSE value when compared to the other two. A cluster of points shown in figure 4 is arranged not well in the diagonal right. Some points are dispersed in diagonal left. It provides the worst MSE value when compared to the other two. From this observation, we can argue that the relationship between the truth and non-falsity values can be used to identify uncertainty in the prediction. This technique can support the decision making in order to select a suitable neural network model.

IV. Conclusion and Future Work

In this paper, the complementary neural networks (CMTNN) are applied to the regression problem. It is found that CMTNN provide better accuracy results when compared to the traditional BPNN and SVM with linear, polynomial, and radial basis function kernels based on housing, concrete, and computer data sets from UCI machine learning database. Moreover, the difference between the truth and non-falsity values can be used to identify uncertainty level for individual complementary neural networks. In the future, we will apply our approach to ensemble neural networks. Other types of neural networks will be considered as well.

References

[1] M. Paliwal and U. A. Kumar, "Neural networks and statistical techniques: A review of applications," *Expert Systems with Applications*, vol. 36, pp. 2–17, 2009.

[2] S. Osowski, K. Siwek, and T. Markiewicz, "MLP and SVM Networks – a Comparative Study," in *Proceedings of the 6th Nordic Signal Processing Symposium*, June 2004, pp. 37–40.

[3] I. S. Msiza, F. V. Nelwamondo, and T. Marwala, "Water Demand Prediction using Artificial Neural Networks and Support Vector Regression," *Journal of Computers*, vol. 3, no. 11, pp. 1–8, November 2008.

[4] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test," *Neurocomputing*, vol. 55, pp. 169–186, 2003.

[5] L. Ma and K. Khorasani, "New training strategies for constructive neural networks with application to regression problems," *Neural Networks*, vol. 17, pp. 589–609, 2004.

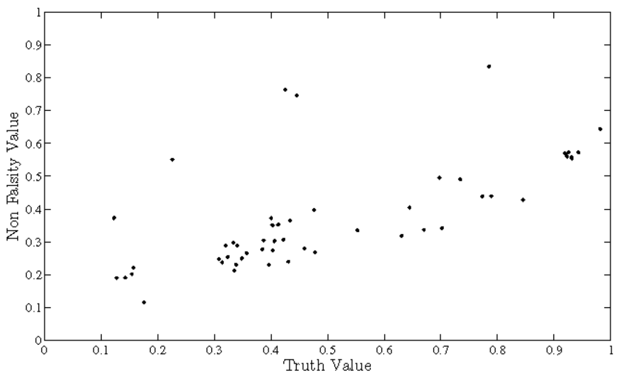

Fig. 4. The relationship between the truth and non-falsity values obtained from CMTNN that provides the results with the MSE of 0.042374 from the test set of Housing data.

[6] H. Altun, A. Bilgil, and B. Fidan, "Treatment of multi-dimensional data to enhance neural network estimators in regression problems," *Expert Systems with Applications*, vol. 32, pp. 599–605, 2007.

[7] R. Setiono and J. Y. Thong, "An approach to generate rules from neural networks for regression problems," *European Journal of Operational Research*, vol. 155, pp. 239–250, 2004.

[8] D. Chetwynd, K. Worden, and G. Manson, "An application of interval-valued neural networks to a regression problem," in *Proceedings of the Royal Society-Mathematical, Physical and Engineering Sciences (Series A)*, vol. 462, no. 2074, 2006, pp. 3097–3114.

[9] Z.-Q. Shen and F.-S. Kong, "Dynamically weighted ensemble neural networks for regression problems," in *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, vol. 6, Shanghai, August 2004, pp. 3492–3496.

[10] P. Kraipeerapun, C. C. Fung, W. Brown, K. W. Wong, and T. D. Gedeon, "Uncertainty in Mineral Prospectivity Prediction," in *ICONIP (2)*, ser. Lecture Notes in Computer Science, I. King, J. Wang, L. Chan, and D. L. Wang, Eds., vol. 4233. Springer, 2006, pp. 841–849.

[11] P. Kraipeerapun, C. C. Fung, and K. W. Wong, "Quantification of Vagueness in Multiclass Classification Based On Multiple Binary Neural Networks," in *Proceedings of the international conference on Machine Learning and Cybernetics 2007 (ICMLC07)*, Hong Kong, China, August 2007, pp. 140–144.

[12] A. Asuncion and D. Newman, "UCI Machine Learning Repository," http://www.ics.uci.edu/~mlearn/MLRepository.html, 2007, university of California, Irvine, School of Information and Computer Sciences.

[13] I.-C. Yeh, "Modeling of strength of high performance concrete using artificial neural networks," *Cement and Concrete Research*, vol. 28, no. 12, pp. 1797–1808, 1998.

[14] S. Rüping, "mySVM-Manual," http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/, 2000, university of Dortmund, Lehrstuhl Informatik 8.

[15] MATHWORKS, *MATLAB Mathematics Version 7*. Natick: The MathWorks Inc., 2004.