Linear and Nonlinear Regression for Combinatorial Optimization Problem of Multiple Transgenesis

Daisuke Tominaga[1,a) Kazuki Mori[2] Sachiyo Aburatani[1]

Abstract: Combinatorial optimization problem is a difficult class of problems from which to obtain exact solutions, but such problems often arise in biotechnology, for example, it is often necessary to find optimal combinations of genes in transgenics to improve production of a useful compound by microorganisms. In the cases of 20 candidate genes for introduction into cells, the number of possible combinations of introduced genes is approximately 10^6. Testing all of their combinations by experimental observation is impossible practically. A few combinations are observed experimentally for large numbers of possible combinations generally.

We tested two methods for the prediction of effects of transgenes: multivariate linear regression and the RBF (Radial Basis Function) network, with a simulated and an unpublished experimentally observed dataset of transgenic yeast. Results show that RBF network can detect a special gene (introduced gene) at the five percent significance level when the gene causes production values that are 1.5 times greater than other genes for the simulated dataset. Prediction by RBF network is superior than that by the linear regression model at the best condition.

Keywords: transgenics, transformation, multivariate linear regression, RBF network

1. Introduction

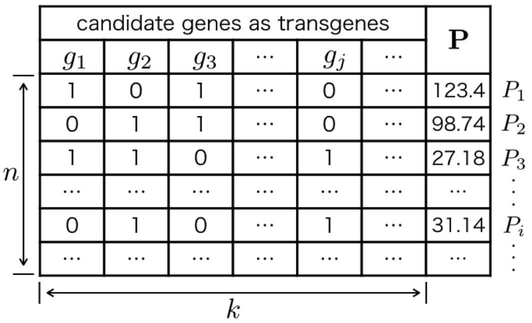
Transgenics is useful as an extremely popular experimental method to improve the production performance of microorganisms that produce a useful industrial or medical compound. Transgenics methods introduce nucleic acid chains into bodies of the microorganisms [1]. The sequence of the nucleic acid chains is generally based on those of natural plasmids, including genes to be expressed in the cell body and promoters to realize the expression [2], [3]. Plural gene sequences are known generally as homologs or orthologs for each protein expressed by the introduced gene[4]. Proteins that are of same or similar function are expressed by transgenic sequences of these genes at different efficiencies of expression level, cell mortality, production rate of useful compounds, and so on. Generally many choices of genes exist for transgenics, such as design bases of plasmids, strains of microorganisms, and substrates and bases [5], [6], to make the microorganisms produce useful compounds with better performance. Excluding quantitative conditions such as substrate concentrations, finding the best conditions for the microorganism performance is the combinatorial optimization problem [5], [7]. The hardness of the problem class that includes combinatorial optimization is widely known. The computational time to solve the problem increases very rapidly with increasing problem size, such as the number of choices to be selected ('combinatorial explosion'). No way exists to obtain the exact solution other than exploring all possible combinations exhaustively. To address that problem, many methods that find approximately optimal solutions have been proposed, such as dynamic programming [8], branch and bound methods [9], and heuristic searches that include genetic algorithms [10]. All methods for combinatorial problems include the calculation of a value from a combination. For example, a combination of a selection of objects and a value is the total weight of the selected objects at the knapsack problem [11]. The value is readily obtained through simple calculation: merely adding all weights of selected objects. For transgenics designed to improve microorganisms, however, a combination is a list of genes. The value is the production rate or amount of the useful compound. Obtaining a pair of a combination and a value necessitates observation by experimentation, which generally requires much cost and time. Predicting the best combinations from the limited number of observations using machine learning methods is a good approach to solving the combinatorial optimization problems related to transgenics.

Machine learning for transgenics entails two difficulties. One is noise in the learning data. Production values often include large noise. Statistics-based approaches such as least-squares fitting are necessary. The second is that there are nonlinear interactions between genes. Modeling these interactions is extremely difficult because their molecular mechanisms or schemes remain unclear. Multivariate linear regression (MVR) [12] is widely used because this is the simplest method to represent gene interactions. There are no other established models. We show that the Radial Basis Function network model (RBF network) [13], [14] as a simple nonlinear model is comparable to or better than the linear model

[1] Biotechnology Research Institute for Drug Discovery, National Institute of Advanced Industrial Science and Technology, 2-4-7 Aomi, Koto ward, Tokyo 135-0064, Japan

[2] Technology Research Association of Highly Efficient Gene Design, 3-17-6 Hacchobori, Chuo ward, Tokyo 104-0032, Japan

[a] tominaga-daisuke@aist.go.jp



candidate genes as transgenes						**P**	
g_1	g_2	g_3	⋯	g_j	⋯		
1	0	1	⋯	0	⋯	123.4	P_1
0	1	1	⋯	0	⋯	98.74	P_2
1	1	0	⋯	1	⋯	27.18	P_3
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋮
0	1	0	⋯	1	⋯	31.14	P_i
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋮

Fig. 1 Structure of the learning data: n, number of learning data; k, number of genes for candidates as transgenes; g_j, j-th gene for candidates as transgenes; and P_i, experimentally observed productions of the microorganism at the experiment i. The $n \times k$ matrix (experiment condition matrix) consists of elements c_{ij} with values of 1 or 0. Here, $c_{ij} = 1$ means that gene j is selected as the transgene and introduced into cells in experiment i.

(MVR) at prediction accuracies on simulated and experimentally observed datasets. The nonlinear model is useful to find the introduced gene that improves the microorganism production.

2. Method

2.1 Learning data

Let the number of learning data (the number of experimental observations) be n, the suffix of each datum be i ($i = 1, \ldots, n$), and the number of candidate genes for transgenics be k. These genes are g_j ($j = 1, \ldots, k$). The variable that means that g_j is selected for introduction into the experiment i is c_{ij} ($c_{ij} = 0$ or 1). In addition, **C** that consists of c_{ij} is called here the experiment condition matrix. Furthermore, $c_{ij} = 1$ means that g_j is introduced at the experiment i; $c_{ij} = 0$ means not to be introduced. Each row vector of $\mathbf{C} = c_{ij}$ corresponds to an experimental observation. Herein, P_i stands for the observed production of the microorganism at experiment i. **P** that consists of P_i signifies an n-dimensional real-number vector, designated here as the production vector (**Fig. 1**).

The number of possible k-dimensional binary vectors is 2^k. The n vectors out of 2^k are observed from experimentation. Machine learning methods use n pairs of these vectors and production values to build a predictor to calculate production at other $2^k - n$ experimental conditions.

2.2 Linear model

Actually, MVR is often used to represent relations between the introduced genes [6]. The total effect on cell activity by these genes is based on the assumption that the total effect can be represented by the total sum of effects of each single gene.

Linear model **a** is obtained as a solution of following linear problem of

$$\mathbf{Ca} = \mathbf{P}, \tag{1}$$

where **C** is the experiment condition matrix and **P** is the production vector. This problem is solvable when the rank of the matrix **C** is equal to the number of learning data n ($\mathrm{rank}(C) = n$). The least-squares fitting method is applied to the problem when the rank of **C** is greater than n. The model can not be solved both exactly and approximately when the rank of **C** is less than n. Prediction of the production value P^* for the condition **c** is calculated as the inner product of **a** and **c** as

$$P^* = \mathbf{ac}, \tag{2}$$

where **c** denotes the k-dimensional binary vector that represents the experimental condition where g_j ($j = 1, \ldots, k$) is introduced ($c_j = 1$) or not ($c_j = 0$).

2.3 Nonlinear model

The prediction of the production value P^* is calculated by the RBF network model as

$$f_i(\mathbf{c}|\sigma) = \exp\left(\frac{|\mathbf{c} - \mathbf{c}_i|^2}{2\sigma^2}\right) \tag{3}$$

$$P^* = \sum_{i=1}^{n} a_i f_i(\mathbf{c}), \tag{4}$$

where **c** stands for the k-dimensional binary vector that represents the experimental condition for the prediction, $\mathbf{c}_i$ signifies the i-th row of the experiment condition matrix **C**, σ denotes a model parameter, P^* represents the prediction value of the microorganism's production, and a_i is the real-number element of vector **a** and the scaling factor to f_i. Also, f_i is designated as the radial basis function (RBF).

The coefficient vector **a** is calculated as the solution of the following linear problem of

$$\mathbf{Fa} = \mathbf{P}. \tag{5}$$

In that equation, **P** is the production vector. Element F_{ij} of the real value matrix **F** is defined as

$$F_{ij} = f_i(\mathbf{c}_j) = \exp\left(\frac{|\mathbf{c}_j - \mathbf{c}_i|^2}{2\sigma^2}\right). \tag{6}$$

Therein, $|\mathbf{c}_j - \mathbf{c}_i|$ is the Euclidean distance between $\mathbf{c}_j$ and $\mathbf{c}_i$, which is equal to the Hamming distance because **C** is the binary matrix.

This calculation is similar to the calculation of the Support Vector Machine with the RBF kernel. Actually, the RBF network is similar to the Support Vector Regression model with the RBF kernel and without the smoothing (noise reduction) ability.

3. Result

3.1 Prediction accuracy on simulated data

We generate both **C** and **P** with random number simulators. Elements of the experiment condition matrix **C** are 0 or 1 of uniformly distributed random numbers. Elements P_i of the production vector **P** are first generated by normally distributed random real numbers with both the mean and the variance of 100 (standard deviation is 10). Then the values of P_i are multiplied by 1.5 when the c_{ij} of **C** is 1 for special j. We intend to simulate that gene j improves production of the microorganism, making it 50% higher than in the condition in which the gene j is not introduced. We chose this special gene as g_1. The number of genes (k) is 10 in this study. All $2^k = 1024$ values of production are generated. Some of these 1024 values are used for learning. The rest are targets of prediction.

The possible number of experimental condition is 1024: the theoretical maximum number of the learning data. In general,

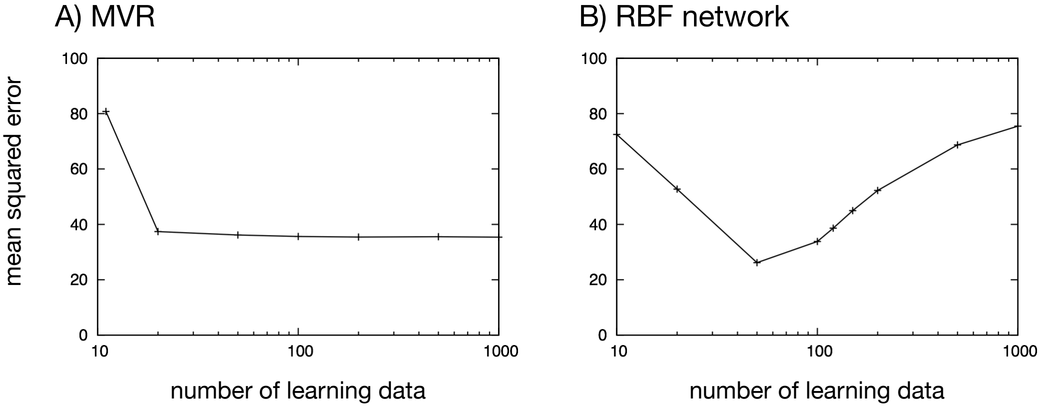


Fig. 2 Prediction accuracy of MVR and the RBF network models on simulated data other than a part of those for learning data. Vertical axes show values of mean squared errors between the prediction of production values and those of simulated values. Horizontal axes show the number of learning data out of 1024 simulated data.

a part of the 1024 pattern of the experimental conditions is observed in practice. These observed conditions are given as the learning dataset. The number of observations is n. We set n to 10 – 1000 for this simulation. Prediction is done for $1024 - n$ unobserved conditions. The prediction accuracy is calculated as the mean squared error between predicted and simulated values. This run is repeated 100 times for each n selecting learning data randomly from 1024 data.

We manually find the value of the model parameter σ to fit the mean of predicted value to 100. Then we chose $\sigma = 0.749$.

The mean values of squared errors between predicted and simulated production values are presented in **Fig. 2**.

3.2 Detection ability on simulated data

We examine MVR and the RBF network models as the detection methods for the special gene that improves production of the microorganism. Special gene g_j is regarded as detected when c_{ij} is 1 in experimental conditions that correspond to higher predicted production values. The average values of 100 detection runs are shown in **Fig. 3**. Simulated data are generated in the same way as that above ($n = 100$), but altering multiplication factors to the production vector from 1.0 to 1.5. The special gene is g_1. The number of conditions in which $c_{i1} = 1$ is counted in the top 100 conditions of high prediction values out of 924 ($= 1024 - n$) conditions.

3.3 Prediction accuracy on experimentally observed data

We examine the prediction accuracy of MVR and the RBF network on un-published experimentally observed dataset of yeast to produce a certain chemical compound from a substrate. The experiment condition matrix is shown in **Fig. 4**. The number of candidate genes (k) is 10; the number of observed data (n) is 36. Note that none of genes are intended as the special gene that is distinguished from other genes in the experimental design.

We fit the linear regression model first that represents the production values by the linear combinations of g_1 to g_{10} to see the fitness of the linear model. The fitted model is shown in **Fig. 5**. Residuals of the fitted linear model to the entire experimentally observed data distribute within a range of $[-520.80, 1002.28]$ while the range of the production in the data is $[77.1, 1837.765]$.

	Estimate
Intercept	-231.46
g_1	380.25
g_2	274.32
g_3	-41.88
g_4	-456.68
g_5	-510.65
g_6	-395.99
g_7	-43.18
g_8	179.05
g_9	342.08
g_{10}	112.17

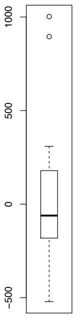


Fig. 5 The fitted linear model to the entire experimentally observed data and the distribution of residuals. Left: the fitted model. Estimate, coefficients in the linear combination of genes. Right: the distribution of 36 residual values of responses of the linear model to data.

Then we examine MVR and the RBF network models for production prediction. We set the number of learning data (n) as 10 to 30 and predict the $36 - n$ production values and calculated prediction accuracy as the average values of the mean squared error between predicted and observed production values. These average values are presented in **Fig. 6** for both MVR and the RBF network.

4. Discussion

4.1 Prediction accuracy

Fig. 2A shows that prediction accuracy by MVR does not improve for larger numbers of learning data. In the case of the RBF network, prediction accuracy gets better and then worse by increasing the number of learning data. This V-letter profile may be the ‘over-learning’ or ‘over-fitting’ [15]. Although the best accuracy is better than MVR (Fig. 2B), the effect of the degree of freedom of the RBF network model on prediction accuracy shall be examined.

RBF network model shows better performances for small numbers of learning data (Fig. 6) than those of MVR. Increasing the number of learning data does not improve prediction accuracy also in the experimentally observed data by both MVR and the RBF network models. It is generally known that the more data used for learning, the better the prediction that can be done. How-

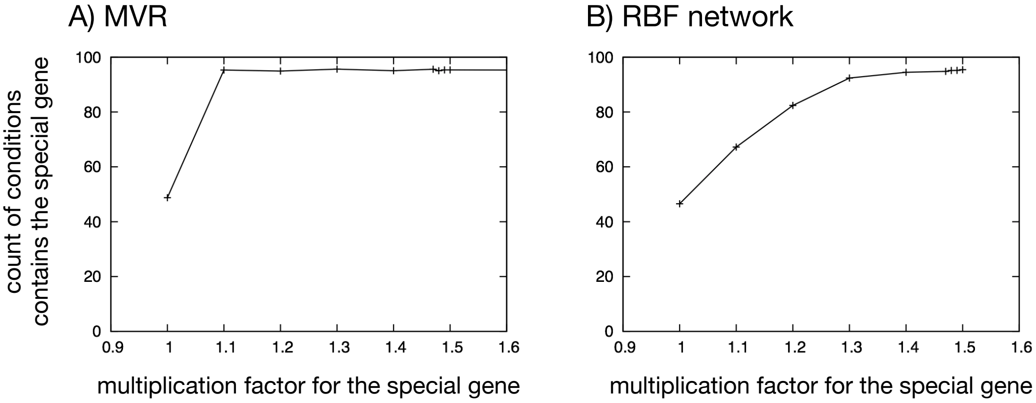


Fig. 3 Detection ability of MVR and the RBF network models on simulated data. The vertical axes show the number of conditions (rows of the experiment condition matrix) in which $c_{i1} = 1$. The horizontal axes are the multiplication factor.

g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	Production
0	0	0	0	0	0	0	1	0	0	63.3
0	0	0	0	0	0	0	1	0	1	98.76
0	0	0	0	0	0	0	1	1	0	109.44
0	0	0	0	0	0	0	1	1	1	341.38
0	0	0	0	0	0	1	0	1	1	104.95
0	0	0	0	0	0	1	1	1	1	579.21
0	0	0	0	0	1	0	1	1	1	305.9
0	0	0	0	1	0	0	1	1	1	111.49
0	0	0	1	0	0	0	1	1	1	163.94
0	0	1	0	0	0	0	0	1	1	126.62
0	0	1	0	0	0	0	1	1	1	147.24
0	1	0	0	0	0	1	0	1	1	180.22
0	1	1	0	0	0	0	0	1	1	196.88
1	0	0	0	0	0	0	0	1	0	776.81
1	0	0	0	0	0	0	0	1	1	82.24
1	0	0	0	0	0	0	1	1	0	809.224
1	0	0	0	0	0	0	1	1	1	474.425
1	0	0	0	0	0	1	0	1	1	791.34
1	0	0	0	0	0	1	1	1	1	527.61
1	0	0	0	0	1	0	0	1	1	107.05
1	0	0	0	0	1	0	1	1	1	368.63
1	0	0	0	1	0	0	0	1	1	77.12
1	0	0	0	1	0	0	1	1	1	212.41
1	0	0	1	0	0	0	0	1	1	84.92
1	0	0	1	0	0	0	1	1	1	282.175
1	0	1	0	0	0	0	0	1	1	870.18
1	0	1	0	0	0	0	1	1	1	578.62
1	1	0	0	0	0	0	0	1	0	404.45
1	1	0	0	0	0	1	0	0	1	408.5
1	1	0	0	0	0	1	0	1	1	1729.32
1	1	0	0	0	1	1	0	1	1	255.62
1	1	0	1	0	0	1	0	1	1	263.41
1	1	1	0	0	0	0	0	0	1	422.3
1	1	1	0	0	0	0	0	1	1	1837.765
1	1	1	0	0	0	1	0	1	1	385.06
1	1	1	0	1	0	0	0	1	1	178.81

Fig. 4 The experiment condition matrix of the un-published experimentally observed data. n and k in Fig. 1 are 36 and 10 respectively.

ever, that pattern does not fit the results obtained for this case study. The optimal number of learning data for prediction accuracy maybe exsist.

4.2 Detection ability

The RBF network model detected special conditions (combinations of transgenes) obtained at the five percent significance level when the condition causes predicted values that were 40–50% greater than other conditions on the simulated dataset (Fig. 3B).

On the other hand, MVR can find the special conditions when the special gene improves the production values 10% greater than other conditions (Fig. 3A). This result seems very nice, however, may be unnatural because noise of biological observations is often greater than 10%. MVR may be too sensitive for small experimentally observed data. Statistical examination for both MVR and RBF network models on small numbers of learning data with noise shall be done.

4.3 Consideration in the order of genes

When introducing plural genes by a single nucleotide chain (the sequence of the chain is constructed of genes and promoter sequences in tandem), expression efficiencies of each gene are generally different depending on its position in the chain sequence. Neither prediction method explained in this paper is relevant to these differences. These expression differences are easily modeled in the two methods if the differences (ratios of expression strengths to the lowest strength, or 'efficiency vector') are known quantitatively and if they do not change at all through the learning data. For the MVR model, effects of the position can be eliminated from the linear model $\mathbf{a}$ in eq. (1) by dividing each element of $\mathbf{a}$ by the element of the efficiency vector in the corresponding position.

For the RBF network model, multiplication of each element of the efficiency vector to the corresponding element of the experiment condition vector $\mathbf{c}_i$ (each row of the experiment condition matrix $\mathbf{C}$), and replacing the original $\mathbf{c}_i$ with the calculated vector can model the expression efficiency differences by position. The distance between two experimental conditions $|\mathbf{c}_j - \mathbf{c}_i|$ in eq. (6) is the real-value Euclidean distance when the elements of matrix $\mathbf{C}$ are real. Freedom of experiment planning is improved by this replacement because alternation of the order of the tandem genes is represented by the order of elements in the efficiency vector. Expression efficiency differences by some other cause, such as gene length, can therefore be represented by the efficiency vector.

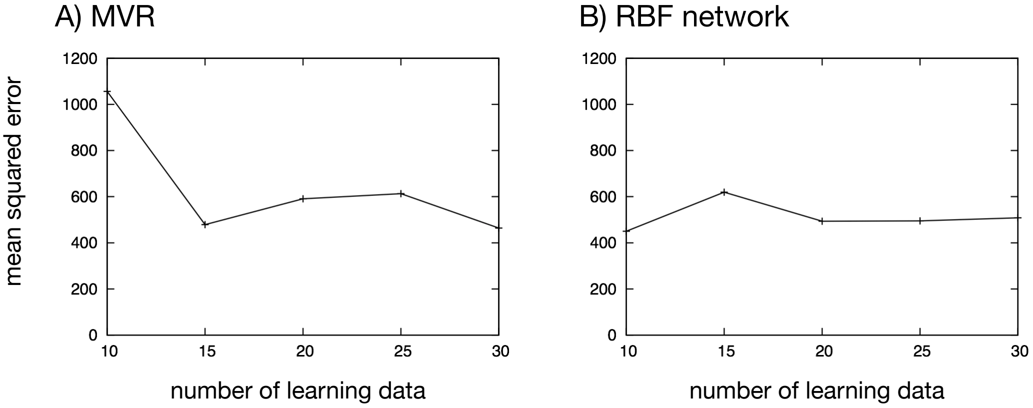


Fig. 6 Prediction accuracy of MVR and the RBF network models on experimentally observed data other than a part of those for learning data. Vertical axes show mean squared errors between the prediction of production values and those of observed values. Horizontal axes show the number of learning data out of 36 observed data.

Actual gene expression efficiencies are often adjusted to similar levels among positions by choices of promoter sequences and the repeat numbers of promoter sequences. Actually, the MVR and RBF network models of this paper are directly applicable for these cases.

Acknowledgments This work was supported by the Commission for Development of Artificial Gene Synthesis Technology for Creating Innovative Biomaterial from the Ministry of Economy, Trade and Industry (METI), Japan.

References

[1] Sambrook, J., Russell, D.: *Molecular Cloning: A Laboratory Manual*, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY(2001).

[2] Tsuge, K., Matsui, K., Itaya, M.: One step assembly of multiple DNA fragments with a designed order and orientation in Bacillus subtilis plasmid *Nucleic Acids Res.*, Vol., 31, No. 21, e133 (2003)

[3] Itaya, M., Fujita, K., Kuroki, A., Tsuge, K.: Bottom-up genome assembly using the Bacillus subtilis genome vector, *Nature Methods*, Vol., 5, pp. 41–43 (2008)

[4] Alberts, B., Johnson, A., Lewis, J., Morgan, D., Raff, M., Roberts, K., Walter, P.: *Molecular biology of the cell, 6th ed.*, Garland Science, New York, NY (2014).

[5] Latimer, L.N., Lee, M.E., Medina-Cleghorn, D., Kohnz, R.A., Nomura, D.K., Dueber, J.E: Employing a combinatorial expression approach to characterize xylose utilization in Saccharomyces cerevisiae, *Metabolic Eng.*, Vol. 25, pp. 20-29 (2014).

[6] Lee, M.E., Aswani, A., Han, A.S., Tomlin, C.J., Dueber, J.E.: Expression-level optimization of a multi-enzyme pathway in the absence of a high-throughput assay, *Nucleic Acids Res.*, Vol. 41, No. 22, pp. 10668-10678 (2013).

[7] Papadimitriou, C. H.: *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications (1998).

[8] Andonov, R., Poirriez, V., Rajopadhye, S.: Unbounded Knapsack Problem : dynamic programming revisited, *J. Operational Res.*, Vol. 123, No. 2, pp.168–181 (2000)

[9] Martello, S., Toth, P.: *Knapsack Problems: Algorithms and Computer Implementation*, John Wiley and Sons, Hoboken, NJ (1990).

[10] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional (1985).

[11] Kellerer, H., Pferschy, U., Pisinger, D.: *Knapsack problems*, Springer, New York, NY (2004).

[12] Cohen, J., Cohen P., West, S.G., Aiken, L.S.: *Applied multiple regression/correlation analysis for the behavioral sciences*, Lawrence Erlbaum Associates, Hillsdale, NJ (2003).

[13] Chen, S., Cowan, C.F.N., Grant, P.M.: Orthogonal least square learning for radial basis function networks, *IEEE Transactions on Neural Networks*, Vol. 2, No. 2, pp. 302-309 (1991).

[14] Langari, R.: Radial basis function networks, regression weights, and the expectation-maximization algorithm, *Systems, Man and Cybernetics*, Vol. 27, No. 5 (1997).

[15] Burnham, K. P., Anderson, D. R.: *Model selection and multimoldel inference: a practical information-theoretic approach*, 2nd Ed., Springer, New York, NY (1998).