2005 International Symposium on Nonlinear
Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

Cellular analysis of covariance structure for Data Mining by Backward Euler method

Yuko Zennyoji, Nao Ohashi, Masayuki Yamauchi and Mamoru Tanaka

Department of Electrical and Elecrtonics Engineering, Sophia University
7-1 Kioi-cho, Chiyoda-ku, Tokyo 102-8554, Japan
Email: [yuko, nao1980, masa, tanaka]@mamoru.ee.sophia.ac.jp

Abstract—This paper describes a cellular analysis of covariance structure for data mining using Back Euler method. It is the implicit method which is the most practical method for solving stiff systems. It is difficult to solve these systems with conventional methods. Davidon-Fletcher-Powell (DFP) method, one of quasi-Newton methods, is utilized to modify the next step solution. At each iteration step for quasi-Newton method, the approximation to the matrix including second-order partial derivatives is updated by using new gradient information. By using both of Back Euler method and DFP method, the solution for stiff systems in the parameter space can be obtained.

1. Introduction

Recently, machine learning methods have been used as data mining to acquire the important information from massive amount of data and to predict future. That is, the data mining is to find 'rule' as classification, prediction, I/O mapping and association by using machine learning algorithm. Conventionally, decision tree methods based on so-called 'if-then' rule have been often used. For example, MLC++ [1] which is developed by Stanford University and Silicon Graphics, Inc is a famous software tools including top-down decision tree algorithms such as C4.5 [2] or bottom-up method like OODGs.

However, the decision tree methods are not very suitable for the information which includes continuous data. Back propagation neural network is very popular as the way of learning, and its application is being expected to solve various problems in many fields. One major drawback to the back propagation algorithm is that interpreting a model is difficult. So it is not easy to detect meaningful information.

We propose a cellular analysis of covariance structure to predict the meaningful information including continuous data. For massive amount of data all sparse matrices corresponding to express cellular signal flow graph (SFG) are constructed by using sparse matrix technique which has been used in the circuit simulation such as SPICE. To understand estimator of population parameters brings to grasp main internal processing. In this paper, Backward Euler method, one of methods for solving an differential equation, is used. The important point is that this method can solve stiff systems which have large difference among eigenvalues. The solution is obtained for a model which is difficult to solve by using conventional methods. The resulted parameters are used as weights on edges in the cellular SFG which works as a prediction model for unknown input data. Our simulation for the model of "Purchase of a Car" shows good result.

2. Cellular Structural State and Measurement Equations

The observed variable $\boldsymbol{x}$ is a visible information data which has been obtained from real human behavior, natural environment and so on. The average $\boldsymbol{\mu_x}$ and the real covariance matrix $\boldsymbol{S}$ is calculated by using the observed variable $\boldsymbol{x}$.

Let $\boldsymbol{\eta} \in \boldsymbol{R}^n$ be the latent variable of covariance structure method. The cellular structural equation is expressed by

$$\boldsymbol{\eta} = \boldsymbol{B}_{\#}\boldsymbol{\eta} + \boldsymbol{\Gamma}\boldsymbol{\xi} + \boldsymbol{\zeta} \tag{1}$$

where $\boldsymbol{B}_{\#} \in \boldsymbol{R}^{n\times n}$ and $\boldsymbol{\Gamma} \in \boldsymbol{R}^{n\times m}$ are coefficient weight matrices which express connection between the variables, and $\boldsymbol{\zeta} \in \boldsymbol{R}^n$ is the error variable for the latent variables.

The cellular measurement equation should be used to express the casual relations among the observed variables $\boldsymbol{x} \in \boldsymbol{R}^l$ and the latent variable $\boldsymbol{\eta}$. The measurement equation is given by

$$\boldsymbol{x} = \boldsymbol{\mu_x} + \boldsymbol{K}\boldsymbol{\eta} + \boldsymbol{\Lambda}\boldsymbol{\xi} + \boldsymbol{e} \tag{2}$$

where $\boldsymbol{K} \in \boldsymbol{R}^{l\times n}$ and $\boldsymbol{\Lambda} \in \boldsymbol{R}^{l\times m}$ are coefficient weight matrices which express connection between the variables $\boldsymbol{\eta}, \boldsymbol{\xi}$ and observed variables $\boldsymbol{x}$, and $\boldsymbol{e} \in \boldsymbol{R}^l$ is the error variables for the input.

Each i-th row vector of the matrix $\boldsymbol{B}_{\#}, \boldsymbol{\Gamma}, \boldsymbol{K}$ and $\boldsymbol{\Lambda}$ is including a weight element w_{ij} on the edge from a cell C_j to C_i. Generally, the number of elements is also very few and then the matrices $\boldsymbol{B}_{\#}, \boldsymbol{\Gamma}, \boldsymbol{K}$ and $\boldsymbol{\Lambda}$ are sparse.

The matrices $\boldsymbol{B}_{\#}, \boldsymbol{\Gamma}, \boldsymbol{K}$ and $\boldsymbol{\Lambda}$ are cellular (sparse) matrices in the case of large SFG.

In this paper, our purpose is to determine the parameters of $\boldsymbol{B}_{\#}, \boldsymbol{\Gamma}, \boldsymbol{K}, \boldsymbol{\Lambda}, \boldsymbol{e}$ and $\boldsymbol{\zeta}$ by proposed learning method.

Table 1: Multivariate data

	Observed Item	S_1	S_2	$\dots$	S_{50}
u_1	Color	5	3	$\dots$	4
u_2	Style	4	4	$\dots$	4
u_3	Power performance	4	4	$\dots$	4
u_4	Suspension setting	3	3	$\dots$	3

3. Optimization

3.1. Fit Function

Let $\boldsymbol{z}$ be the model standardized vector, then it is given by

$$\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{\mu_x}. \tag{3}$$

Let $\boldsymbol{C}_u \in \boldsymbol{R}^{l \times l}$ be the covariance matrix of the state variable, then it is derived from the cellular structural equation and the measurement equation as follow:

$$\begin{aligned} \boldsymbol{C}_u &= E(\boldsymbol{z}\boldsymbol{z}') \\ &= \boldsymbol{G}\boldsymbol{B}_0\boldsymbol{\Gamma}_0\boldsymbol{\Phi}_0\boldsymbol{\Gamma}_0'\boldsymbol{B}_0'\boldsymbol{G}' \end{aligned} \tag{4}$$

where $\boldsymbol{G} = (\mathbf{I}\ \mathbf{0}), \boldsymbol{B}_0(\boldsymbol{B}_{\#}, \boldsymbol{K}), \boldsymbol{\Gamma}_0(\boldsymbol{\Lambda}, \boldsymbol{\Gamma}), \boldsymbol{\Phi}_0(\boldsymbol{\Delta}, \boldsymbol{\Psi}, \boldsymbol{\Phi})$. Here these $\boldsymbol{\Delta}$, $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ are covariance matrices of $\boldsymbol{\xi}, \boldsymbol{\zeta}$ and $\boldsymbol{e}$. $\mathbf{X}'$ is a transposed matrix of $\mathbf{X}$.
Let $\boldsymbol{\theta} \in \boldsymbol{R}^p$ be vector of population parameters which is elements of the matrices $\boldsymbol{B}_{\#}, \boldsymbol{K}, \boldsymbol{\Lambda}, \boldsymbol{\Gamma}, \boldsymbol{\Delta}, \boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$, then the Generalized Least Squares (GLS) method is applied to the fit function as

$$f_{GLS}(\boldsymbol{\theta}) = \frac{1}{2} tr((\boldsymbol{S} - \boldsymbol{C}_u)\boldsymbol{S}^{-1})^2 \tag{5}$$

where $\boldsymbol{S} \in \boldsymbol{R}^{l \times l}$ is the real sample covariance matrix given by

$$\boldsymbol{S} = \frac{1}{N}\boldsymbol{Z}\boldsymbol{Z}', \tag{6}$$

$\boldsymbol{Z} \in \boldsymbol{R}^{l \times l}$ is the data matrix standardized by expected value from Table 1 and N is the number of samples and $tr(\mathbf{X})$ means a trace of the matrix $\mathbf{X}$.

The matrix $\boldsymbol{C}_u$ is approached to $\boldsymbol{S}$ by using optimization calculation to obtain all parameters of sparse matrices and errors. In this paper, we use both of Backward Euler method and quasi-Newton method.

3.2. New Algorithm

The purpose of this paper is that function (5) is minimized, and the parameter at that time is determined. So we want to solve the equation:

$$\mathbf{g}(\boldsymbol{\theta}) = 0 \tag{7}$$

where $\mathbf{g}(\boldsymbol{\theta}) = \frac{\partial f_{GLS}}{\partial \boldsymbol{\theta}}$.

However, the convergence depend on initial value when a nonlinear equation is solved by an iterative solution method. In ordre to achieve the purpose, we solve the following equation by using Backward Euler method.

$$\dot{\boldsymbol{\theta}} = \mathbf{g}(\boldsymbol{\theta}) \tag{8}$$

3.2.1. Backward Euler Method

The Backward Euler method is implicit, in that it uses the differentiation at the next time step, instead of the current one. Implicit methods are the most practical method for solving stiff systems. This method approximates the solution $f_{GLS}(\boldsymbol{\theta})$ at virtual time $t_{k+1} = t_k + h$ by solving the implicit equation:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + h\mathbf{g}(\boldsymbol{\theta}_{k+1}) \tag{9}$$

where the gradient vector $\mathbf{g}(\boldsymbol{\theta}_k)$ is evaluated at $\boldsymbol{\theta}_k$.

Since this equation(9) may be nonlinear, solving it in general requires an iterative solution method. In this paper, quasi-Newton method is provided for solving the implicit equation.

3.2.2. Quasi-Newton Method

The given function is approximated in each iteration by a truncated Taylor series.

$$F(\boldsymbol{\theta}) \approx F(\boldsymbol{\theta}_n) + \mathbf{G}(\boldsymbol{\theta}_n)'(\boldsymbol{\theta} - \boldsymbol{\theta}_n) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_n)'\mathbf{H}(\boldsymbol{\theta}_n)(\boldsymbol{\theta} - \boldsymbol{\theta}_n) \tag{10}$$

where the gradient vector $\mathbf{G}(\boldsymbol{\theta}_n)$ is evaluated at $\boldsymbol{\theta}_n$. $\mathbf{H}(\boldsymbol{\theta}_n) \in \mathbf{R}^{p \times p}$ is the matrix of second-order partial derivatives of function with respect to $\boldsymbol{\theta}_n$. This is called Hessian. If it assume that $F(\boldsymbol{\theta})$ takes its minimum at $\boldsymbol{\theta} = \boldsymbol{\theta}_n$, the gradient is zero.

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \alpha\mathbf{H}(\boldsymbol{\theta}_n)^{-1}\mathbf{G}(\boldsymbol{\theta}_n) \tag{11}$$

where α is the step size, setting to 1. However, as the Hessian leads to algorithmic and computational complexities, an approximation technique of the inverse Hessian is often used. We use Davidon-Fletcher-Powell (DFP) method which is one of quasi-Newton methods. The update formula is as follows:

$$\mathbf{H}_{n+1} = \mathbf{H}_n + \frac{\mathbf{z}\mathbf{z}'}{\mathbf{z}'\mathbf{u}} - \frac{\mathbf{H}_n'\mathbf{u}\mathbf{u}'\mathbf{H}_n}{\mathbf{u}'\mathbf{H}_n\mathbf{u}} \tag{12}$$

where

$$\mathbf{z} = -\alpha\mathbf{H}_n\mathbf{G}(\boldsymbol{\theta}_n) \quad \mathbf{u} = \mathbf{G}(\boldsymbol{\theta}_{n+1}) - \mathbf{G}(\boldsymbol{\theta}_n)$$

. Because of conjugate property of direction vector, $\mathbf{H}(\boldsymbol{\theta}_{n+1})$ is set to $\mathbf{H}(\boldsymbol{\theta}_1)$ after p-iterations.

An initial matrix $\mathbf{H}(\boldsymbol{\theta}_1)$ is unit matrix. Then the approximation at first follows the line of steepest descent, and later follows the estimated Hessian more closely.

3.2.3. Proposed Method

An implicit method requires the solution of a nonlinear equation at each step. For one step of Backward Euler method, we use the quasi-Newton method.

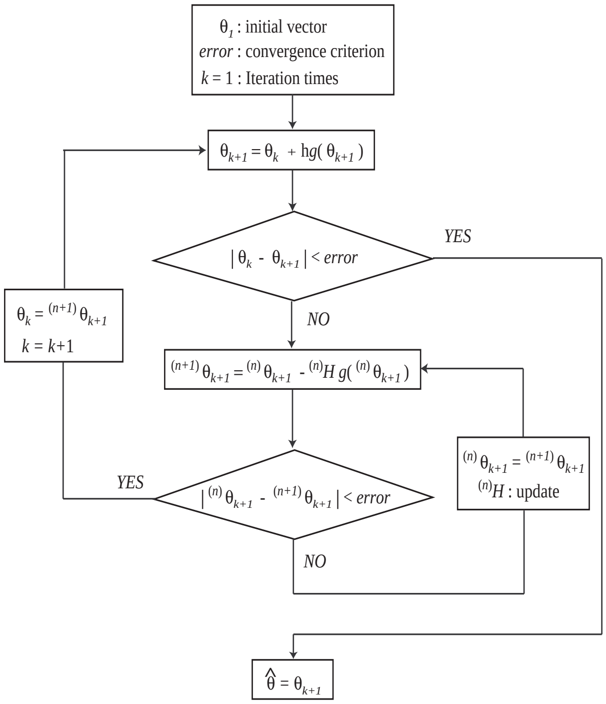
$$\mathbf{F}(\boldsymbol{\theta}_{k+1}) = \boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k - h\mathbf{g}(\boldsymbol{\theta}_{k+1}) \tag{13}$$

In order to satisfy the equation(9), $\mathbf{F}(\boldsymbol{\theta}_{k+1})$ is minimized by Newton method.

$${}^{(n+1)}\boldsymbol{\theta}_{k+1} = {}^{(n)}\boldsymbol{\theta}_{k+1} - \left(\frac{\partial \mathbf{F}({}^{(n)}\boldsymbol{\theta}_{k+1})}{\partial {}^{(n)}\boldsymbol{\theta}_{k+1}}\right)^{-1}\mathbf{F}({}^{(n)}\boldsymbol{\theta}_{k+1}) \tag{14}$$

$$= {}^{(n)}\boldsymbol{\theta}_{k+1} - \left(\mathbf{I} - h\frac{\partial \mathbf{g}({}^{(n)}\boldsymbol{\theta}_{k+1})}{\partial {}^{(n)}\boldsymbol{\theta}_{k+1}}\right)^{-1}\mathbf{F}({}^{(n)}\boldsymbol{\theta}_{k+1}) \tag{15}$$

The computation of the matrix $\left(\frac{\partial \mathbf{g}({}^{(n)}\boldsymbol{\theta}_{k+1})}{\partial {}^{(n)}\boldsymbol{\theta}_{k+1}}\right)$ is not available or expensive. Then the approximation technique (DFP) is used. We replace $\left(\mathbf{I} - h\frac{\partial \mathbf{g}({}^{(n)}\boldsymbol{\theta}_{k+1})}{\partial {}^{(n)}\boldsymbol{\theta}_{k+1}}\right)$ by the approximation (equation(12)). Fig 1 shows flowchart of this method.



θ_1 : initial vector
error : convergence criterion
$k = 1$: Iteration times
$\theta_{k+1} = \theta_k + hg(\theta_{k+1})$
YES
$|\theta_k - \theta_{k+1}| < error$
NO
$\theta_k = {}^{(n+1)}\theta_{k+1}$
$k = k+1$
${}^{(n+1)}\theta_{k+1} = {}^{(n)}\theta_{k+1} - {}^{(n)}H\, g({}^{(n)}\theta_{k+1})$
${}^{(n)}\theta_{k+1} = {}^{(n+1)}\theta_{k+1}$
${}^{(n)}H$: update
YES
$|{}^{(n)}\theta_{k+1} - {}^{(n+1)}\theta_{k+1}| < error$
NO
$\hat{\theta} = \theta_{k+1}$

Figure 1: flowchart of proposed method

4. Model of "Purchase of a Car"

The model of "Purchase of a Car" is used as an example of analysis of the cellular covariance structure. A part of the observed data is showed in Table 1. This data is collected from survey to 50 people. Four points considered at the purchase of a private car were evaluated by five stages. The four variables $[x_1, x_2, \ldots, x_4]$ are defined as observed variables in Table 1.

The state variable of η_1 means a design, the state variable of η_2 means a performance, and the state variable of ξ_1 means a value of a car. The parameters of ζ_1 and ζ_2 are the error variables.

A design and a performance are determined by a value of a car. A user can set the parameters in advance. Some parameters of the matrices are set to 0 before learning. It is useful to set some parameters previously.

It is very important that the coefficient matrices are sparse and its SFG is cellular network. The weights on the edges incident to a cell C_j are corresponding to the template like that of cellular neural network.

In the simulation, we use two models.

4.1. Model 1

The cellular structural state equation of the model for "Purchase of a Car" is given by

$$\begin{pmatrix}\eta_1\\ \eta_2\end{pmatrix} = \begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}\begin{pmatrix}\eta_1\\ \eta_2\end{pmatrix} + \begin{pmatrix}\gamma_{11}\\ \gamma_{21}\end{pmatrix}(\xi_1) + \begin{pmatrix}\zeta_1\\ \zeta_2\end{pmatrix} \tag{16}$$

The cellular measurement equation can be also defined by the user as follows

$$\begin{pmatrix}x_1\\ x_2\\ x_3\\ x_4\end{pmatrix} = \begin{pmatrix}\mu_{x1}\\ \mu_{x2}\\ \mu_{x3}\\ \mu_{x4}\end{pmatrix} + \begin{pmatrix}\kappa_{11} & 0\\ \kappa_{21} & 0\\ 0 & \kappa_{32}\\ 0 & \kappa_{42}\end{pmatrix}\begin{pmatrix}\eta_1\\ \eta_2\end{pmatrix} + \begin{pmatrix}0\\ 0\\ 0\\ 0\end{pmatrix}(\xi_1) + \begin{pmatrix}e_1\\ e_2\\ e_3\\ e_4\end{pmatrix} \tag{17}$$

The SFG corresponding to the equations is given in Fig 2.

4.2. Model 2

In the second model, a pass is added. The design is expressed by the value of car and the performance. Then the equations are as follows

$$\begin{pmatrix}\eta_1\\ \eta_2\end{pmatrix} = \begin{pmatrix}0 & \beta_{12}\\ 0 & 0\end{pmatrix}\begin{pmatrix}\eta_1\\ \eta_2\end{pmatrix} + \begin{pmatrix}\gamma_{11}\\ \gamma_{21}\end{pmatrix}(\xi_1) + \begin{pmatrix}\zeta_1\\ \zeta_2\end{pmatrix} \tag{18}$$

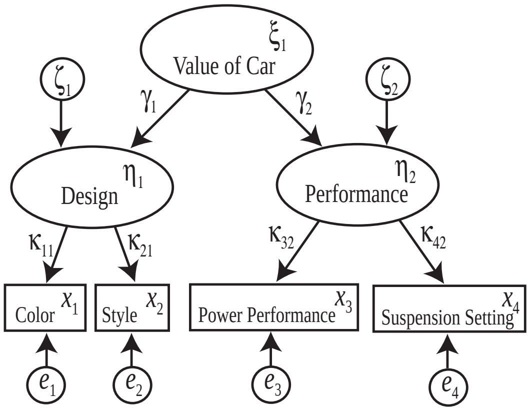


Figure 2: SFG of "Purchase of a Car" model1

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} \mu_{x1} \\ \mu_{x2} \\ \mu_{x3} \\ \mu_{x4} \end{pmatrix} + \begin{pmatrix} \kappa_{11} & 0 \\ \kappa_{21} & 0 \\ 0 & \kappa_{32} \\ 0 & \kappa_{42} \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} \xi_1 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} \tag{19}$$

The SFG corresponding to the equations is given in Fig 3.

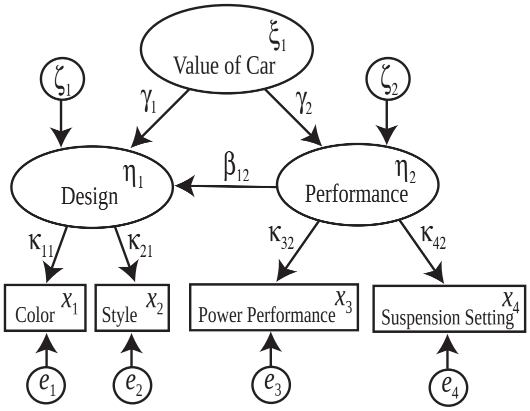


Figure 3: SFG of "Purchase of a Car" model2

5. Simulation Results

In the simulation, the initial value $\boldsymbol{\theta}_0$ was set to random parameters from -1 to 1, the value of convergence criterion was equal to 10^{-6}, and maximum number of iterations was 2000.

We simulated conventional method (software) about same models. Then in the model 2, we could not get result. But the parameters were determined by our proposed method. Fig 4 shows the learning curves for the model 2 of "Purchase of a Car". The number of steps is shown on the horizontal axis, and a value of fit function is shown on the vertical axis.

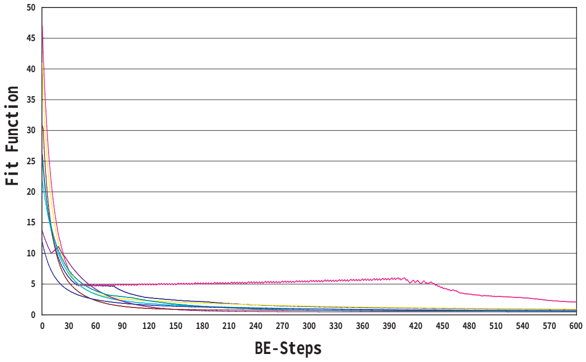


Figure 4: The simulation results of model 2

Conventional methods have identifiability problem. Therefore setting some conditions is required. However our proposed method could get the parameters.

6. Conclusion

In this paper, a novel cellular analysis method of covariance structure for data mining was proposed. We used Back Euler method for solving stiff systems. Then the quasi-Newton method was utilized to modify the next step solution. Since this method can approximate the inverse matrix including second-order partial derivatives, which is arduous to compute. Experimental results show that the performance of our proposed method has better than that of conventional methods. In future, the nonlinear CNN differential equations for (1) and (2) will be used and automatic model decision will be found for data mining.

Acknowledgments

This research is supported by the fund of Open Research Center Project from MEXT of Japanese Government (2002-2006).

References

[1] Kohavi, Ron, Dan Sommerfield, and James Dougherty, "Data Mining using MLC++", Tool with AI, IEEE, pp.234-245, 1996.

[2] Quinlan, J. Ross, "C4.5: Programs for machine learning", Morgan Kaufmann Publishers, 1993.