Efficient Input Variable Selection for Soft-senor Design based on Nearest Correlation Spectral Clustering and Group Lasso

Koichi Fujiwara*, Manabu Kano

Dept. of Systems Science, Kyoto University,
Yoshida-Honmachi, Sakyoku, Kyoto 606-8501, JAPAN

Abstract

Soft sensors have been widely used for estimating product quality or other key variables, and appropriate input variables have to be selected for building highly accurate soft sensor. A novel input variable selection method based on nearest correlation spectral clustering (NCSC), which is a correlation-based clustering method, has been proposed, and it is referred to as NCSC-based variable selection (NCSC-VS). In NCSC-VS, NCSC is used for variable group construction, and a few variable groups are selected by their contribution to estimates. Although NCSC-VS can select appropriate input variables and construct an accurate soft sensor, a lot of parameters have to be tuned carefully for selecting proper variables because its parameter tuning affects the estimation performance of a soft sensor greatly. The present work proposes a new methodologies for efficient input variable selection by integrating NCSC and group Lasso, which is an extension of Lasso that can select variable groups. The proposed method, referred to as NCSC-based group Lasso (NCSC-GL), can not only reduce the number of tuning parameters but also achieve almost the same performance as NCSC-VS. The usefulness of the proposed NCSC-GL is demonstrated through applications to soft sensor design for a pharmaceutical process and a chemical process.

Keywords: Soft-sensor design, Input variable selection, Group Lasso, Spectral clustering, Near infrared spectroscopy

1. Introduction

Soft sensors, or calibration models, are key techniques for estimating product quality or other important variables when online analyzers are not available. In chemical processes, for example, soft sensors have been widely used to estimate

*Corresponding author. Tel.: +81-75-753-3369
Email address: fujiwara.koichi@i.kyoto-u.ac.jp (Koichi Fujiwara)

product quality of distillation columns, reactors, etc. In addition, near infrared spectroscopy (NIRS) has been used as a powerful online monitoring tool because of its noninvasiveness and short measuring time. Many kinds of material attributes such as water content, blend uniformity and content uniformity have been estimated using calibration models with NIR spectra [1, 2]. Partial least squares (PLS), in particular, has been used to build an accurate soft sensor with a small number of latent variables [3, 4, 5].

In general, a soft sensor can be fitted to model construction samples when the number of input variables is large. However, its estimation performance may deteriorate when input variables that do not physically relate to the objective property are used. Although appropriate input variables have to be selected when a soft sensor is built, the computational load increases greatly if all possible combinations of variables are tested. When the past information is used as inputs in addition to the present information to take process dynamics into account in soft sensor design, variable selection becomes difficult because the number of candidate input variables increases. In addition, when a calibration model with NIR spectra is constructed, its input variables are wavelengths in NIR spectra and the number of them is large.

Therefore a systematic methodology for selecting appropriate input variables is required for improving the estimation performance as well as the efficiency of soft sensor design [6, 7].

Although a genetic algorithm (GA) can be applied to variable selection [8], its computational load is still heavy. On the other hand, PLS-based variable selection methods, such as PLS-Beta and variable influence on projection (VIP), have been proposed [9]. In addition, stepwise and least absolute shrinkage and selection operator (Lasso) are well-known variable selection methods for linear regression [10, 11]. These methods evaluate each candidate variable independently as to whether or not it should be used as an input variable; however such an evaluation is not appropriate because it has a correlation with other variables. In particular, wavelengths of NIR spectra are strongly correlated with each other.

Recently, a novel variable selection method based on nearest correlation spectral clustering (NCSC) [12, 13], which can cluster samples according to the correlation among them, has been proposed, and it is referred to as NCSC-based variable selection (NCSC-VS) [14]. In NCSC-VS, some variable groups are constructed based on the correlation among variables using NCSC, because it is important to consider the correlation appropriately when building a good regression model [15]. After variable group construction, each variable group is examined as to whether or not it should be used as input variables according to their contribution to the estimates. However, since multiple parameters have to be tuned at the same time in NCSC-VS for good variable selection, its parameter tuning is time consuming. Therefore the number of tuning parameters in NCSC-VS should be reduced for its practical use.

Group Lasso has been proposed as an extension of Lasso for selecting variables from predefined variable groups and it has just one tuning parameter [16, 17]. Although variable groups have to be made before using group Lasso, ap-

propriate variable grouping is not always clear.

The present work proposes a new correlation-based variable selection method by integrating NCSC and group Lasso. In the proposed method, referred to as NCSC-based group Lasso (NCSC-GL), NCSC constructs variable groups in the same manner as NCSC-VS and group Lasso selects a few variable groups as input variables of a soft sensor. Although the number of tuning parameters in NCSC-GL is less than that of NCSC-VS, it can achieve almost the same performance as NCSC-VS. That is, the proposed NCSC-GL can design an accurate soft sensor efficiently.

This paper reports application results of the proposed NCSC-GL to soft sensor design for a pharmaceutical process and a chemical process.

2. Partial least squares (PLS)

PLS is widely used linear regression method for building a soft sensor. In PLS, the input $\boldsymbol{X} \in \Re^{N \times M}$ and the output $\boldsymbol{y} \in \Re^N$ are decomposed as follows:

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{E} && (1) \\ \boldsymbol{y} &= \boldsymbol{T}\boldsymbol{b} + \boldsymbol{f} && (2) \end{aligned}$$

where $\boldsymbol{T} \in \Re^{N \times K}$ is the latent variable matrix whose columns are the latent variable $\boldsymbol{t}_k \in \Re^N$ $(k = 1, \cdots, K)$, $\boldsymbol{P} \in \Re^{M \times K}$ is the loading matrix of $\boldsymbol{X}$ whose columns are the loading vectors $\boldsymbol{p}_k \in \Re^M$ and $\boldsymbol{b} = [b_1, \cdots, b_K]^T$ is the loading vector of $\boldsymbol{y}$. K denotes the number of adopted latent variables. $\boldsymbol{E} \in \Re^{N \times M}$ and $\boldsymbol{f} \in \Re^N$ are errors.

The nonlinear iterative partial least squares (NIPALS) algorithm can be used to construct a PLS model [3]. Suppose that the first to kth latent variables $\boldsymbol{t}_1, \cdots, \boldsymbol{t}_k$, loading vectors $\boldsymbol{p}_1, \cdots, \boldsymbol{p}_k$ and loading $b_1, \cdots, b_k$ are given. The $k+1$th residual input and output can be written as follows:

$$\begin{aligned} \boldsymbol{X}_{k+1} &= \boldsymbol{X}_k - \boldsymbol{t}_k \boldsymbol{p}_k^T && (3) \\ \boldsymbol{y}_{k+1} &= \boldsymbol{y}_k - b_k \boldsymbol{t}_k. && (4) \end{aligned}$$

where $\boldsymbol{X}_1 \equiv \boldsymbol{X}$, $\boldsymbol{y}_1 \equiv \boldsymbol{y}$, and $\boldsymbol{t}_k$ is a linear combination of the columns of $\boldsymbol{X}_k$, that is, $\boldsymbol{t}_k = \boldsymbol{X}_k \boldsymbol{w}_k$ where $\boldsymbol{w}_k \in \Re^M$ is the kth weighting vector. $\boldsymbol{w}_k$ is defined so that the covariance between $\boldsymbol{y}_k$ and $\boldsymbol{t}_k$ is maximized under $||\boldsymbol{w}_k|| = 1$. Using the Lagrange multipliers method, the function to be maximized can be defined as

$$G_k = \boldsymbol{y}_k^T \boldsymbol{t}_k - \mu(||\boldsymbol{w}_k||^2 - 1) = \boldsymbol{y}_k^T \boldsymbol{X}_k \boldsymbol{w}_k - \mu(||\boldsymbol{w}_k||^2 - 1) \quad (5)$$

where μ is the Lagrange multiplier. Solving $\partial G_k / \partial \boldsymbol{w} = 0$, $\boldsymbol{w}_k$ is derived as

$$\boldsymbol{w}_k = \frac{\boldsymbol{X}_k^T \boldsymbol{y}_k}{||\boldsymbol{X}_k^T \boldsymbol{y}_k||}. \quad (6)$$

The kth loading vector $\boldsymbol{p}_k$ and the kth loading b_k are as follows:

$$\boldsymbol{p}_k = \frac{\boldsymbol{X}_k^T \boldsymbol{t}_k}{\boldsymbol{t}_k^T \boldsymbol{t}_k},\ b_k = \frac{\boldsymbol{y}_k^T \boldsymbol{t}_k}{\boldsymbol{t}_k^T \boldsymbol{t}_k}. \tag{7}$$

Finally, the above procedure is repeated until the number of adopted latent variables K is achieved; K can be determined by cross validation.

The estimation performance of the PLS model can be improved by taking into account process dynamics. For this purpose, the past information is used as inputs in addition to the present information. In such a case, the input variable vector $\boldsymbol{z}_i$ is written as

$$\boldsymbol{z}_i = [\boldsymbol{x}_i^{\mathrm{T}}, \boldsymbol{x}_{i-k_1}^{\mathrm{T}}, \boldsymbol{x}_{i-k_2}^{\mathrm{T}}, \cdots]^{\mathrm{T}} \tag{8}$$

where $k_1, k_2, \cdots$ are i or less arbitrary natural numbers, and which past samples are added to the present information can be determined according to process knowledge such as a time constant. This method is referred to as dynamic PLS [18, 19].

3. Conventional input variable selection methods

This section explains several conventional input variable selection methods briefly.

3.1. PLS-Beta

PLS-Beta translates a PLS model, Eqs. (1) and (2), into a multiple linear regression (MLR) model and selects input variables based on the magnitude of its regression coefficients [9]. The estimate $\hat{\boldsymbol{y}}$ is expressed as follows:

$$\hat{\boldsymbol{y}} = \boldsymbol{T}(\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{T}\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}_{pls}. \tag{9}$$

The regression coefficients $\boldsymbol{\beta}_{pls}$ are described as

$$\boldsymbol{\beta}_{pls} = \boldsymbol{W}(\boldsymbol{P}^T\boldsymbol{W})^{-1}(\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{y} \tag{10}$$

where $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K] \in \Re^{M\times K}$.

The input variables can be selected individually in descending order of the magnitude of $\boldsymbol{\beta}_{pls}$ until

$$\frac{||\boldsymbol{\beta}_{select}||}{||\boldsymbol{\beta}_{pls}||} > \nu\ (0 < \nu \leq 1) \tag{11}$$

is achieved, where $\boldsymbol{\beta}_{select}$ denotes the vector of the regression coefficients corresponding to the selected variables.

3.2. Variable influence on projection (VIP)

The VIP, which expresses the contribution of the input variable to the output variable, can be used for variable selection [9]. The VIP score of the jth input variable is defined as

$$V_j = \sqrt{M \sum_{k=1}^{K} \left(w_{jk}^2 b_k^2 (\boldsymbol{t}_k^T \boldsymbol{t}_k) / ||\boldsymbol{w}_k||^2 \right) \Big/ \sum_{k=1}^{K} b_k^2 (\boldsymbol{t}_k^T \boldsymbol{t}_k)} \tag{12}$$

where w_{jk} is the jth element of $\boldsymbol{w}_k$. The variables satisfying $V_j > \eta$ (> 0) should be selected as input variables, and the default values of η can be determined as one since the average of all the VIP scores is one [9].

3.3. Selectivity ratio (SR)

The SR score is a variable selection index based on PLS [20]. It is defined as the ratio of the variance of the input variables explained by the latent variables v_j^{expl} to the variance of the errors v_j^{err}, that is, $s_j = v_j^{expl}/v_j^{err}$, because the input variables can be expressed as the sum of the latent variables and the errors as written in Eq. (1). In the SR method, the input variables should be selected according to the magnitude of s_j.

The input variables can be individually selected in descending order of the SR scores s_j until

$$\frac{||\boldsymbol{s}_{select}||}{||\boldsymbol{s}_{all}||} > \xi \ (0 < \xi \leq 1) \tag{13}$$

is achieved, where $\boldsymbol{s}_{all}$ and $\boldsymbol{s}_{select}$ denote the SR score vectors of all the input variables and the selected input variables.

3.4. Stepwise

Stepwise selects the input variables of an MLR model on the basis of a hypothesis test. It repeats model construction by adding and eliminating a variable step by step, and tests whether the true value of the regression coefficient of the added variable is zero using the F-test [10].

3.5. Least absolute shrinkage and selection operator (Lasso)

Lasso minimizes the sum of the squared errors with L_1 regularization [11]. The objective function is formulated as:

$$\boldsymbol{\beta}_{lasso} = \arg\min_{\boldsymbol{\beta}} \ \left(||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_1 \right) \tag{14}$$

where λ (> 0) is a parameter. In Lasso, some regression coefficients tend to be zero, and this characteristic can be used for variable selection. Lasso can be solved by an efficient algorithm called least angle regressio (LARS) whose computational load is the same order as ordinal least squares (OLS) [21].

4. Nearest correlation spectral custering (NCSC)

Although NCSC was originally proposed for sample clustering based on the correlation between variables by not assuming any distribution [12, 13], it can be used for variable selection.

NCSC integrates the nearest correlation (NC) method [22] that can detect samples whose correlation is similar to the query and spectral clustering (SC) [23, 24] that can partition a weighted graph. The NC method constructs the weighted graph that expresses the correlation-based similarities between samples, and SC partitions the constructed graph.

4.1. Spectral clustering (SC)

SC is a clustering method based on graph theory. It can partition a weighted graph whose weights express affinities between nodes into subgraphs through cutting some of their arcs. Although some spectral clustering algorithms have been proposed, the max-min cut (Mcut) method [23] is described in this subsection.

Given a weighted graph G and its affinity matrix (adjacency matrix) $\boldsymbol{W}$, G is partitioned into two subgraphs A and B. The affinity between A and B is defined as

$$\begin{aligned} \mathrm{cut}(A,B) &\equiv W(A,B) && (15)\\ W(A,B) &= \sum_{u\in A, v\in B} W_{u,v} && (16)\\ W(A) &\equiv W(A,A). && (17) \end{aligned}$$

where u and v denote nodes of subgraphs A and B, respectively. That is, the affinity between subgraphs $\mathrm{cut}(A,B)$ is the sum of the weights of the arcs between subgraphs. The Mcut method searches subgraphs A and B that can minimize $\mathrm{cut}(A,B)$ and maximize $W(A)$ and $W(B)$, simultaneously. The objective function of the Mcut method is as follows:

$$\min\ J = \frac{\mathrm{cut}(A,B)}{W(A)} + \frac{\mathrm{cut}(A,B)}{W(B)}. \tag{18}$$

Since indices of the nodes are interchangeable, the affinity matrix $\boldsymbol{W}$ can be defined as follows:

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_A & \boldsymbol{W}_{A,B} \\ \boldsymbol{W}_{B,A} & \boldsymbol{W}_B \end{bmatrix} \tag{19}$$

where $\boldsymbol{W}_A$ and $\boldsymbol{W}_B$ are the affinity matrices within the subgraphs A and B, respectively, and $\boldsymbol{W}_{A,B}$ and $\boldsymbol{W}_{B,A}$ are the affinity matrices between A and B. The affinity $\boldsymbol{W}$ is described using vectors $\boldsymbol{x} = [1,\cdots,1,0,\cdots,0]^T$ and $\boldsymbol{y} = [0,\cdots,0,1,\cdots,1]^T$ that express partition into subgraphs.

$$\begin{aligned} W(A,B) &= \boldsymbol{x}^T(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{x} = \boldsymbol{y}^T(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{y} && (20)\\ W(A) &= \boldsymbol{x}^T\boldsymbol{W}\boldsymbol{x} && (21)\\ W(B) &= \boldsymbol{y}^T\boldsymbol{W}\boldsymbol{y} && (22) \end{aligned}$$

where $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{W}\boldsymbol{e})$ and $\boldsymbol{e} = [1, \cdots, 1]^T$. Using these equations, Eq. (18) can be described as

$$\min_{\boldsymbol{x},\boldsymbol{y}} \ J = \frac{\boldsymbol{x}^T(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{W}\boldsymbol{x}} + \frac{\boldsymbol{y}^T(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{y}}{\boldsymbol{y}^T\boldsymbol{W}\boldsymbol{y}}. \tag{23}$$

Only the first term of Eq. (23) needs to be analyzed.

The minimization problem of Eq. (23) is hard to solve, since the optimizing variables $\boldsymbol{x}$ and $\boldsymbol{y}$ are binary variables. Therefore, these binary variables should be relaxed. The new index vector $\boldsymbol{q} = \{a, -b\}$ $(a, b > 0)$ is introduced:

$$q_u = \begin{cases} a & (u \in A) \\ -b & (u \in B) \end{cases} \tag{24}$$

where q_u is the uth element of $\boldsymbol{q}$. Finally, the problem can be written as

$$\min_{\boldsymbol{q}} \ J_q = \frac{\boldsymbol{q}^T(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{q}}{\boldsymbol{q}^T\boldsymbol{W}\boldsymbol{q}}. \tag{25}$$

This minimization problem results in an eigenvalue problem.

$$(\boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2})\boldsymbol{z} = \lambda\boldsymbol{z} \tag{26}$$

The solution $\boldsymbol{q}^*$ of Eq. (25) is expressed as $\boldsymbol{q}^* = \boldsymbol{D}^{-1/2}\boldsymbol{z}_2$, where $\boldsymbol{z}_2$ is the eigenvector corresponding to the second smallest eigenvalue λ_2. The detailed analysis of the Mcut method is described in Appendix A.

Through the above procedure, the Mcut method does not need parameter tuning. Although the Mcut method partitions a weighted graph into two subgraphs, an extended algorithm that can partition the graph into multiple subgraphs simultaneously has been proposed [24]. The algorithm integrates the eigenvalue problem and the k-means method for clustering, and it can be used for multiclass clustering.

In SC, the definition of an affinity and the number of clusters J are arbitrary and affects results.

4.2. Nearest correlation (NC) method

The NC method can detect samples whose correlation is similar to the query without any correct labels [22].

The concept of the NC method is as follows. Suppose that the affine subspace P in Fig. 1 (left) shows the correlation among variables and all the samples on P have the same correlation. Although $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_5$ have the same correlation, $\boldsymbol{x}_6$ and $\boldsymbol{x}_7$ have a different correlation from the others. The NC method aims to detect samples whose correlation is similar to the query $\boldsymbol{x}_1$. In this example, $\boldsymbol{x}_2, \cdots, \boldsymbol{x}_5$ on P should be detected.

First, the whole space is translated so that the query becomes the origin as shown in Fig. 1 (right). That is, $\boldsymbol{x}_1$ is subtracted from all other samples $\boldsymbol{x}_i$ $(i = 2, \cdots, 7)$. Since the translated affine subspace contains the origin, it becomes the linear subspace V.

Next, a line connecting each sample and the origin is drawn. Then, one must check whether another sample can be found on this line. In this case, $\boldsymbol{x}_2$-$\boldsymbol{x}_5$ and $\boldsymbol{x}_3$-$\boldsymbol{x}_4$ satisfy such a relationship. The correlation coefficients of these pairs must be 1 or -1. On the other hand, $\boldsymbol{x}_6$ and $\boldsymbol{x}_7$, which are not the elements of V, cannot make such pairs. The pairs whose correlation coefficients are ± 1 are thought to have the same correlation as $\boldsymbol{x}_1$.

In practice, the threshold of the correlation coefficient γ $(0 < \gamma \leq 1)$ has to be used, since there are no pairs whose correlation coefficient is strictly ± 1. Therefore, the pairs should be selected when the absolute values of their correlation coefficients are larger than γ.

The pairs whose correlation is similar to the query can be detected through the above procedure.

4.3. Nearest correlation spectral clustering (NCSC)

The correlation-based affinity matrix for spectral clustering can be constructed using the NC method. Assume that samples $\boldsymbol{x}_n \in \Re^M$ $(n = 1, 2, \cdots, N)$ are stored in the database. The NCSC procedure is given in Algorithm 1. In this algorithm, steps 4-9 correspond to the NC method.

In NCSC, the threshold in the NC method γ and the number of clusters partitioned by SC J are tuning parameters. γ should be close to one, such as 0.99, however it is difficult to determine the appropriate J and it should be determined by trial and error unless physical knowledges of a process can be used.

According to Algorithm 1, the computational load of affinity matrix construction by the NC method is $O(N^3)$ where N is the number of samples. In addition, SC results in the eigenvalue problem and it can be solved by QR decomposition whose computational load is usually $O(N^3)$. Therefore the the computational load of NCSC is $O(N^3)$.

5. NCSC-based variable selection (NCSC-VS)

An input variable selection method based on NCSC, referred to as NCSC-based variable selection (NCSC-VS), has been proposed [14]. In NCSC-VS, NCSC constructs variable groups based on the correlation, and evaluates each variable group as to whether or not it should be used as input variables.

First, NCSC-VS classifies variables into J variable groups $\boldsymbol{v}_j = \{x_m \mid m \subset \mathcal{V}_j\}$ $(j = 1, \cdots, J)$ based on their correlation, where $\mathcal{V}_j$ is the subset of the index of variables and $\mathcal{V} = \cup_j \mathcal{V}_j$. To construct variable groups by NCSC, an affinity matrix is constructed from the transposed variable matrix $\boldsymbol{X}^T$ using the NC method. Next, the jth PLS model with the number of latent variable P, f_j^P is constructed from the jth variable group matrix $\boldsymbol{X}_j$, whose columns consist of variables belonging to the jth variable group $\boldsymbol{v}_j$. The validity of the constructed PLS model f_j^P is examined using the contribution ratio to the estimates:

$$C_j^P = 1 - \frac{||\hat{\boldsymbol{y}}_j^P||^2}{||\boldsymbol{y}||^2} \tag{27}$$

where $\hat{\boldsymbol{y}}_j^P$ denotes the estimates by f_j^P.

Finally, D ($\leq J$) variable groups are selected in descending order of C_j^P and the final PLS model is constructed from the variables belonging to the selected variable groups. The procedure for NCSC-VS is shown in Algorithm 2.

In NCSC-VS, NCSC becomes $O(M^3)$ because the input matrix $\boldsymbol{X} \in \Re^{N \times M}$ is transposed for variable clustering, while the computational load of PLS remains approximately $O(N)$ [25]. Thus, most of the computational time in NCSC-VS is occupied by NCSC when M becomes large.

NCSC-VS has four tuning parameters; the parameter in the NC method γ, the number of variable groups partitioned by SC, J, those of latent variables in PLS models P and final selected variable groups D. In order to select good input variables by NCSC-VS, these four parameters should be tuned and this task is time consuming.

Therefore an efficient variable selection method should be developed for the practical use.

6. NCSC-based group Lasso (NCSC-GL)

The present work proposes a new NCSC-based variable selection method for efficient soft sensor design. The proposed method integrates NCSC and group Lasso, which selects some groups consisting of multiple variables [16], and it is referred to as NCSC-based group Lasso (NCSC-GL). Since the total number of tuning parameters in NCSC-GL is smaller than that of NCSC-VS, it can design a soft sensor efficiently.

6.1. Group Lasso

Group Lasso is an extension of Lasso for selecting some input variable groups from predefined multiple variable groups for a linear regression model [16, 17]. Suppose that M variables are divided into J groups and $\boldsymbol{X}_j$ and $\boldsymbol{\beta}_j$ denote the input data matrix and the regression coefficient vector corresponding to the jth group, respectively. The number of variables in the jth group is M_j, that is, $M = \sum_{j=1}^{J} M_j$. The objective function of group Lasso is formulated as:

$$\boldsymbol{\beta}_{glasso} = \arg\min_{\boldsymbol{\beta}} \left(||\boldsymbol{y} - \sum_{j=1}^{J} \boldsymbol{X}_j \boldsymbol{\beta}_j||_2^2 + \lambda \sum_{j=1}^{J} \sqrt{M_j} ||\boldsymbol{\beta}_j||_2 \right) \tag{28}$$

where $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^T, \cdots, \boldsymbol{\beta}_J^T]^T$, and λ is a parameter.

In group Lasso, an entire variable group can be dropped out and this results in Lasso when the number of variable groups $J = M$. Group Lasso can be solved using the same algorithm used for ordinal Lasso like LARS [16]. In order to select input variables by group Lasso, variable groups have to be constructed in advance, however how to construct variable groups is not clear.

6.2. NCSC-based group Lasso (NCSC-GL)

The proposed NCSC-GL forms multiple variable groups by NCSC in the same manner as NCSC-VS and selects some variable groups by group Lasso as input variables of a soft sensor. This integration will be useful because variable groups have to be formed before using group Lasso. The procedure for the proposed NCSC-GL is shown in Algorithm 3.

Since group Lasso can be solved by LARS whose computational load is the same order of OLS, NCSC occupies most of the computational time in NCSC-GL like NCSC-VS. However, the number of tuning parameters in NCSC-GL is three; the parameter in the NC method γ, the number of variable groups J formed by SC and the parameter in group Lasso λ. That is, the number of tuning parameters in NCSC-GL is one less than that of NCSC-VS.

7. Case study 1: pharmaceutical process

The input wavelength selection result of the proposed NCSC-GL was compared with those of the conventional methods through an application to the pharmaceutical data provided by Daiichi Sankyo Co., Ltd. [26].

7.1. Objective data

The target drug products consist of six components. Some blending experiments were conducted with different active pharmaceutical ingredient (API) content. After each blending experiment, the granules for tableting were taken out, and NIR spectra (2203 points in 800−2500 nm) and the API content were measured. The objective of this case study is to select appropriate input wavelengths of NIR spectra for constructing a precise calibration model that can estimate the API content.

The calibration data and the validation data consisted of 576 and 20 samples, respectively.

7.2. Wavelength selection and model construction

Before modeling, a first order differential Savitzky-Golay smoothing filter [27] was applied to the spectra. A PLS model, called PLS-All, employing all the wavelengths was constructed as a benchmark, and the number of its adopted latent variables was determined by cross validation. Input wavelengths were selected using PLS-Beta, VIP, SR, stepwise, Lasso, group Lasso, NCSC-VS and the proposed NCSC-GL.

In PLS-Beta, VIP and SR, their parameters ν, η and ξ were selected as $\nu = \{0.70,\ 0.75,\ 0.80,\ 0.85,\ 0.90,\ 0.95\}$, $\eta = \{0.6,\ 0.7,\ 0.8,\ 0.9,\ 1.0,\ 1.1\}$ and $\xi = \{0.70,\ 0.75,\ 0.80,\ 0.85,\ 0.90,\ 0.95\}$, respectively.

The threshold of the p-value in stepwise was $\bar{p} = \{0.005,\ 0.05,\ 0.08,\ 0.1,\ 0.12,\ 0.15\}$. In Lasso, the parameter was $\lambda = \{0.1,\ 0.2,\ 0.4,\ 0.5,\ 0.8,\ 1.0\}$. Before using group Lasso, the full-spectrum region was divided into several wavelength groups so that they consisted of the same number of successive wavelengths [28]. The

number of constructed wavelength groups was $J = \{5,\ 6,\ 7,\ 8,\ 9,\ 10\}$ and the parameter in Eq. (28) was $\lambda = \{20,\ 25\}$.

In NCSC-based variable selection methods, the parameter of the NC method was $\gamma = 0.99$ and the number of wavelength groups obtained with NCSC was $J = \{5,\ 6,\ 7,\ 8,\ 9,\ 10\}$. In NCSC-VS, the number of latent variables in a PLS model was $P = \{9,\ 10,\ 11\}$ and that of final selected wavelength groups was $D = \{2,\ 3\}$. On the other hand, the parameter of Eq. (28) in NCSC-GL was $\lambda = \{20,\ 25\}$. Parameters used in each method were selected by trial and error.

PLS models were constructed with the wavelengths selected by each method. Usually, Lasso and group Lasso can not only select input variables but also derive regression coefficients. However, PLS models were constructed based on the wavelength selection results by Lasso, group Lasso and the proposed NCSC-GL in this case study, since the number of selected wavelengths were still large and there was the correlation among them. The optimal numbers of latent variables were determined through cross validation, and the API content was estimated by the constructed PLS models.

Figure 2 shows the number of selected wavelengths and the root-mean-square error (RMSE) for the calibration data in each parameter. In this figure, #WL denotes the number of selected wavelengths. The desirable parameters and the input variables in each method were selected by consulting Fig. 2. Finally, the PLS models were constructed with the selected input variables, and the API content was estimated by the constructed PLS models.

Table 1 summarizes the selected parameters, the numbers of selected latent variables (#LV), RMSE, the determination coefficient R^2 and the average CPU times [s] in each method. The computer configuration was as follows: OS: CentOS 6.4 (64bit), CPU: Intel Core i7-3930K (3.2GHz×6), RAM: 64G bytes, and MATLAB 2014b. In addition, Figs. 3 - 11 shows the detailed estimation results.

These results show that the estimation performance of stepwise was worse than PLS-All; its performance was not improved whichever parameter was selected. PLS-Beta, VIP, SR, Lasso and group Lasso improved the estimation performance. On the other hand, both NCSC-based methods achieved higher performance than the conventional methods. NCSC-GL, in particular, achieved the highest performance of all methods, and RMSE was improved by about 40% in comparison with PLS-All.

The average computational loads of NCSC-VS and NCSC-GL were heavier than those of other methods, because NCSC uses iteration for similarity calculation, and it occupied about 98% of their CPU time. However, the CPU time is not important since input variables can be selected offline. On the other hand, it is difficult to find the optimal parameter combination when number of the number of tuning parameters is large. The actual variable selection by NCSC-GL was more efficient than NCSC-VS. The total number of computation for parameter tuning in NCSC-GL was 12 and that of NCSC-VS was 36 in this case study. In addition, according to Fig. 2, the parameters in NCSC-VS D and P affected model construction results greatly, while the parameters in NCSC-GL λ did not affect results so much. This means that variable selection

in NCSC-GL is easier than NCSC-VS.

7.3. Discussion

To validate the wavelength selection results, the wavelengths selected by PLS-Beta, VIP, SR, Lasso, group Lasso are shown in Fig. 12. The colored bands denote the selected wavelengths. PLS-Beta, stepwise and Lasso selected discontinuous wavelengths. VIP, SR and group Lasso selected continuous wavelengths; however, they contained not only the spectra peaks but also other regions.

Figure 13 shows the wavelengths selected by NCSC-VS and NCSC-GL and the wavelength groups constructed by NCSC in each method, and the same color bands denote the same group. The optimal number of wavelength groups of NCSC-VS was $J = 6$ and that of NCSC-GL was $J = 8$. The wavelength groups generated by NCSC consisted of some successive spectra regions, and the wavelength groups selected by both NCSC-VS and NCSC-GL contained almost only specific peaks. These wavelength selection results are consistent with physicochemical knowledge that peaks in spectra contain much information about compounds. Therefore it is concluded that NCSC-VS and NCSC-GL can select meaningful wavelengths for soft sensor design.

8. Case study 2: chemical process

The usefulness of the proposed NCSC-GL is shown through another case study. A soft sensor for estimating ethane concentration was constructed to realize the highly efficient operation of the ethylene fractionator at the Showa Denko K.K. (SDK) Oita plant in Japan [14, 29].

8.1. Ethylene fractionator

A schematic diagram of the ethylene fractionator, referred to as T431/2, is shown in Fig. 14. This ethylene fractionator consists of the bottom column T431 and the top column T432. The feed stream enters the bottom column, and the product ethylene is drawn from the top column. The main specification is the ethane concentration in the ethylene product. Although the ethane concentration must not exceed its upper bound, it should be kept as high as possible to keep the operating cost as low as possible.

This fractionator is controlled by applying multivariable model predictive control. The number of controlled variables, manipulated variables, and disturbance variables is seven, four, and three, respectively. The controlled variables are the ethane concentration and methane concentration in the ethylene product, T431 tray #29 temperature, T431 differential pressure, T432 differential pressure, condenser pot level, and reboiler pot level. Manipulated variables are the T431 reboiler flow rate, T432 internal reflux flow rate, T432 purge flow rate, and T432 top pressure. The disturbance variables are T431 feed flow rate, T431 feed ethane concentration, and C351 #4 discharge pressure. C351 is a propylene

compressor. Its #4 discharge pressure affects propylene refrigerant temperature and then reboiler heat duty.

To realize the highly efficient operation, a soft sensor that can estimate the ethane concentration accurately in real time needs to be developed.

8.2. Operation data

In the ethylene fractionator, 33 variables listed in Table 2 were measured and stored in the database every 5 minutes. To take account of the dynamics of the ethylene fractionator, the candidate input variables consisted of the present sample and the past samples measured 5, 10, 15, 20, 25, 30, 35, 40 and 45 minutes earlier since it was empirically known to engineers and operators that the ethylene fractionator achieved a new steady state within less than an hour. As a result, the total number of candidate variables was 330 (33×10) and appropriate input variables had to be selected to design a precise soft sensor using the input variable selection methods.

Operation data obtained from November to December 2001 was used to build a soft sensor, and the ethane concentration was estimated from January to February 2002.

8.3. Variable selection and model construction

A PLS model, called PLS-All, employing all the candidate variables was constructed as a benchmark, and input variables were selected using the same manner as Sec. 7. However, group Lasso was not used in this case study because how to construct variable groups was not clear in this process. The desirable parameter in each method was selected by trial and error, and the numbers of latent variables in the final PLS models were determined by cross validation. In addition, the SDK engineers selected the input variables on the basis of physical process knowledge [29]. The selected variables are indicated by asterisks in Table 2. Since the number of SDK selection variables was 18, the total number of input variables of the soft sensor was 180 ($= 18 \times 10$).

Figures 16-25 show the final ethane concentration estimation results in each methods and Table 3 summarizes the determined parameters, the numbers of latent variables, the average CPU times [s] and the final estimation results. The computer configuration was the same as Sec. 7.

SDK selection, PLS-Beta, SR, Lasso and stepwise achieved almost the same performance as PLS-All while the number of input variables of the soft sensors could be reduced. However, the performance VIP deteriorated and it did not improve whatever the threshold η was selected. On the other hand, NCSC-VS and NCSC-GL achieved almost the same estimation performance and they were the highest of all the methods. Their RMSEs were improved by about 36% in comparison with PLS-All.

In addition, Fig. 26 shows the input variable selected by the conventional methods and the proposed NCSC-GL. Each cell denotes whether or not a variable was selected as input of the soft sensor. The horizontal line denotes the measured variables corresponding to those in Table 2, and the vertical lines

shows measurement time. Although the conventional methods did not select the specific variables or the specific measurement time, NCSC-VS and the proposed NCSC-GL selected variables with all different measurement time together because NCSC classified the same variables with different measurement time into the same variable classes due to autocorrelation between them.

In this case study, the average computational load of stepwise was heavier than NCSC-VS and NCSC-GL, this indicates that their computational loads are not so heavy when the number of candidate input variables is small, because the number of iterations for similarity calculation in NCSC is proportional to the square of the number of candidate variables.

These case studies show that parameter tuning in NCSC-GL is more efficient than NCSC-VS although they achieve almost the same estimation performance.

9. Conclusions

This work proposed a new input variable selection method for efficient soft sensor design. In the proposed NCSC-GL, NCSC forms variable groups based on the correlation and variable group are selected using group Lasso. The proposed NCSC-GL can realize efficient input variable selection and improve estimation performance of a soft sensor at the same time. The usefulness of NCSC-GL was demonstrated through case studies of soft sensor design for real processes. In the pharmaceutical process, the proposed NCSC-GL improved RMSE by about 40% in comparison with PLS-All while reduced the number of input wavelengths by about half. In the chemical process, RMSE was improved by about 36% in comparison with PLS-All although the number of input variable was reduced from 330 to 290 by the proposed method. Therefore NCSC-GL has the potential for realizing efficient soft sensor design.

Acknowledgements

The authors thank Daiichi-Sankyo Co., Ltd. and Showa Denko K.K. for providing real operation data used in case studies. This work was partially supported by Japan Society for the Promotion of Science (JSPS), Grant in-Aid for Scientific Research (C) 24560940.

References

[1] Y. Roggo, P. Chalus, L. Maurer, C. Lema-Martinez, A. Edmond, N. Jent, A review of near infrared spectroscopy and chemometrics in pharmaceutical technologies, J. Pharm. Biomed. Anal 44 (2007) 683–700.

[2] T. Miyano, M. Kano, H. Tanabe, H. Nakagawa, T. Watanabe, H. Minami, Spectral fluctuation dividing for efficient wavenumber selection: Application to estimation of water and drug content in granules using near infrared spectroscopy, Int. J. Pharm. 475 (2014) 504–513.

[3] S. Wold, M. Sjostroma, L. Erikssonb, Pls-regression: a basic tool of chemometrics, Chemom. Intell. Lab. Syst. 58 (2001) 109–130.

[4] M. Kano, M. Ogawa, The state of the art in chemical process control in japan: Good practice and questionnaire survey, J. Process Control 20 (2010) 969–982.

[5] M. Kano, K. Fujiwara, Virtual sensing technology in process industries: Trends and challenges revealed by recent industrial applications, J. Chem. Eng. Jpn. 46 (2013) 1–17.

[6] C. M. Andersen, R. Bro, Variable selection in regression – a tutorial, J. Chemometrics 24 (2010) 728–737.

[7] T. Mehmood, K. H. Liland, L. Snipen, S. Sæbø, A review of variable selection methods in partial least squares regression, Chemom. Intell. Lab. Syst. 118 (2012) 62–69.

[8] M. Arakawa, Y. Yamashita, K. Funatsu, Genetic algorithm-based wavelength selection method for spectral calibration, J. Chemometrics 25 (2011) 10–19.

[9] H. Kubinyi, 3D QSAR in Drug Design; Theory, Methods, and Applications, ESCOM, Leiden, Holland, 1993.

[10] R. R. Hocking, The analysis and selection of variables in linear regression, Biometrics 32 (1976) 1–49.

[11] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Series B Stat. Methodol. 58 (1996) 267–288.

[12] K. Fujiwara, M. Kano, S. Hasebe, Development of correlation-based clustering method and its application to software sensing, Chemom. Intell. Lab. Syst. 101 (2010) 130–138.

[13] K. Fujiwara, M. Kano, S.Hasebe, Correlation-based spectral clustering for flexible process monitoring, J. Process Control 21 (2011) 1348–1448.

[14] K. Fujiwara, H. Sawada, M. Kano, Input variable selection for pls modeling using nearest correlation spectral clustering, Chemom. Intell. Lab. Syst. 118 (2012) 109–119.

[15] K. Fujiwara, M. Kano, S. Hasebe, Soft-sensor development using correlation-based just-in-time modeling, AIChE J. 55 (2009) 1754–1765.

[16] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, J. R. Stat. Soc. Series B Stat. Methodol. 68 (2006) 49–67.

[17] F. Bach, Consistency of group lasso and multiple kernel learning, J. Mach. Learn. Res. 9 (2008) 1179–1225.

[18] M. Kano, K. Miyazaki, S. Hasebe, I. Hashimoto, Inferential control system of distillation compositions using dynamic partial least squares regression, J. Process Control 10 (2000) 157–166.

[19] N. L. Ricker, Use of biased least-squares estimators for parameters in discrete-time pulse response models, Ind. Eng. Chem. Res. 27 (1998) 343–350.

[20] T. Rajalahtia, R. Arnebergc, F. S. Bervend, K. Myhra, R. J. Ulvikd, O. M. Kvalheim, Biomarker discovery in mass spectral profiles by means of selectivity ratio plot, Chemom. Intell. Lab. Syst. 95 (2009) 35–48.

[21] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Statist. 32 (2004) 407–499.

[22] K. Fujiwara, M. Kano, S. Hasebe, Development of correlation-based pattern recognition algorithm and adaptive soft-sensor design, Control Eng. Pract. 20 (2012) 371–378.

[23] C. H. Q. Ding, X. He, H. Zha, M. Gu, H. D. Simon, A min-max cut algorithm for graph partitioning and data clustering, in: IEEE Int'l Conf. Data Min. (ICDM), 2001, pp. 107 – 114.

[24] A. N. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: NIPS, 2002, pp. 849–856.

[25] L. Qin, H. Snoussi, F. Abdallah, Online learning partial least squares regression model for univariate response data, in: Eur. Signal Process. Conf. (EUSIPCO), Lisbon, 2014, pp. 1073 – 1077.

[26] S. Kim, M. Kano, H. Nakagawa, S. Hasebe, Estimation of active pharmaceutical ingredients content using locally weighted partial least squares and statistical wavelength selection, Int. J. Pharm. 421 (2011) 269–274.

[27] A. Savitzky, M. J. E. Golay, Smoothing and differentiation of data by simplified least squares procedures, Anal. Chem. 36 (1964) 627–1639.

[28] L. Norgaard, A. Saudland, J. Wagner, J. P. Nielsen, L. Munck, S. B. Engelsen, Interval partial least-squares regression (ipls): A comparative chemometric study with an example from near-infrared spectroscopy, Appl. Spectrosc. 54 (2000) 413–419.

[29] H. Kamohara, A. Takinami, M. Takeda, M. Kano, S. Hasebe, I. Hashimoto, Product quality estimation and operating condition monitoring for industrial ethylene fractionator, J. Chem. Eng. Jpn. 37 (2004) 422–428.

[30] M. Fiedler, Algebraic connectivity of graphs, Czech. Math. J. 23 (1976) 298–305.

Table 1: Selected parameters, estimation performance and CPU times needed for wavelength selection in API content estimation

	#WL	#LV	Parameters	RMSE	R^2	CPU time [s]
PLS-All	2203	37	–	1.28	0.83	–
PLS-Beta	928	36	$\nu = 0.75$	1.06	0.81	1.73
VIP	1133	19	$\eta = 0.8$	1.01	0.83	0.55
SR	612	20	$\xi = 0.75$	1.20	0.82	1.46
Lasso	1138	39	$\lambda = 0.2$	0.98	0.87	0.89
group Lasso	1457	20	$J = 7$, $\lambda = 20$	1.03	0.90	0.11
stepwise	561	24	$\bar{p} = 0.15$	1.42	0.72	28.1
NCSC-VS	843	25	$J = 6$, $D = 2$	0.77	0.92	281.0
NCSC-GL	1059	18	$J = 8$, $\lambda = 25$	0.71	0.93	278.5

Table 2: Measured variables of the ethylene fractionator (asterisks indicate the SDK selection variables)

No.	Variables	No.	Variables
1*	T431 tray #29 temperature	18*	T432 reflux ratio
2	T432 outlet temperature	19	T431 differential pressure
3	T432 top temperature	20	T432 differential pressure
4	T432 reflux temperature	21*	T432 top pressure
5	T431 feed temperature	22	T432 outlet methane concentration
6*	T431 bottom temperature	23	outlet acetylene concentration
7*	T431 top temperature	24*	T431 feed ethane concentration
8*	T431 tray #37 temperature	25	C351 #1 intake pressure
9*	T432 tray #129 temperature	26	C351 #1 intake temperature
10	T432 tray #90 temperature	27	C351 #1 discharge pressure
11*	T431 reflux flow rate	28	C351 #1 discharge temperature
12*	T431 reboiler flow rate	29*	C351 #2 discharge pressure
13*	Product ethylene flow rate	30*	C351 #2 discharge temperature
14*	T432 reflux flow rate	31*	C351 #3 discharge pressure
15*	T432 internal reflux flow rate	32	C351 #3 discharge temperature
16	T431 reboiler ethane flow rate	33*	V359 level (cooling propylene)
17*	T432 purge flow rate		

Table 3: Selected parameters, estimation performance and CPU time needed for variable selection in ethane concentration estimation

	#Var.	#LV	Parameters	RMSE	R^2	CPU time [s]
PLS-All	330	47	–	28.7	0.77	–
SDK selection	180	37	–	23.8	0.77	–
PLS-Beta	117	25	$\mu = 0.75$	25.7	0.75	0.51
VIP	293	26	$\eta = 0.6$	31.5	0.75	0.82
SR	271	45	$\mu = 0.95$	26.0	0.72	0.51
Lasso	217	30	$\lambda = 20$	28.1	0.77	5.71
stepwise	151	40	$\alpha = 0.01$	26.6	0.79	8.18
NCSC-VS	200	35	$J = 9, D = 3$	18.0	0.86	5.40
NCSC-GL	290	29	$J = 8$, $\lambda = 2.5$	18.1	0.86	6.36

Algorithm 1 Nearest correlation spectral clustering (NCSC)

Set parameters γ and J.
Set $\boldsymbol{S} \in \Re^{N \times N} \leftarrow \boldsymbol{O}_{N,N}$ and $L = 1$.
for $L = 1$ to N **do**
  Set $\boldsymbol{S}_L \in \Re^{N \times N} \leftarrow \boldsymbol{O}_{N,N}$.
  for all $n = 1, 2, \cdots, N$ $(n \neq L)$ **do**
    $\boldsymbol{x}'_n = \boldsymbol{x}_n - \boldsymbol{x}_L$.
  end for
  for all k, l $(k \neq l)$ such that $|C'_{k,l}| \geq \gamma$ **do**
    $(\boldsymbol{S}_L)_{k,l} = (\boldsymbol{S}_L)_{l,k} = 1$.
  end for
  $\boldsymbol{S} = \boldsymbol{S} + \boldsymbol{S}_L$.
end for
Partition $\boldsymbol{S}$ by SC.

Algorithm 2 NCSC-based Variable Selection (NCSC-VS)

Set parameters γ, J, P and D.
Apply NCSC with γ and J to $\boldsymbol{X}^T$; $\mathcal{V} \rightarrow \{\mathcal{V}_j\}$ $(j = 1, \cdots, J)$.
for $j = 1$ to J **do**
  Construct a PLS model f_j^P with the number of latent variables P.
  Evaluate a contribution ratio C_j^P.
end for
Sort $\mathcal{V}_j$ in descending order of C_j^P.
Select $\mathcal{V}_s \leftarrow \cup_{j=1}^{D} \mathcal{V}_j as input variable of a soft sensor$.

Algorithm 3 NCSC-based group Lasso (NCSC-GL)

Set parameters γ, J and λ.
Apply NCSC with γ and J to $\boldsymbol{X}^T$; $\mathcal{V} \rightarrow \{\mathcal{V}_j\}$ $(j = 1, \cdots, J)$.
Apply group Lasso with λ to the variable groups constructed in step 2.

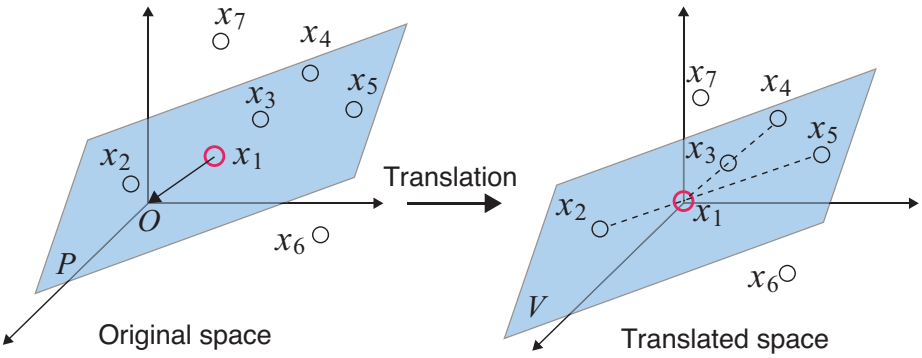

Figure 1: An example of the procedure of the NC method: The red circle denotes the query ($\boldsymbol{x}_1$) and the whole space is translated so that the query becomes the origin (left). The dished lines show pairs of samples that have the same correlation as the query (left).

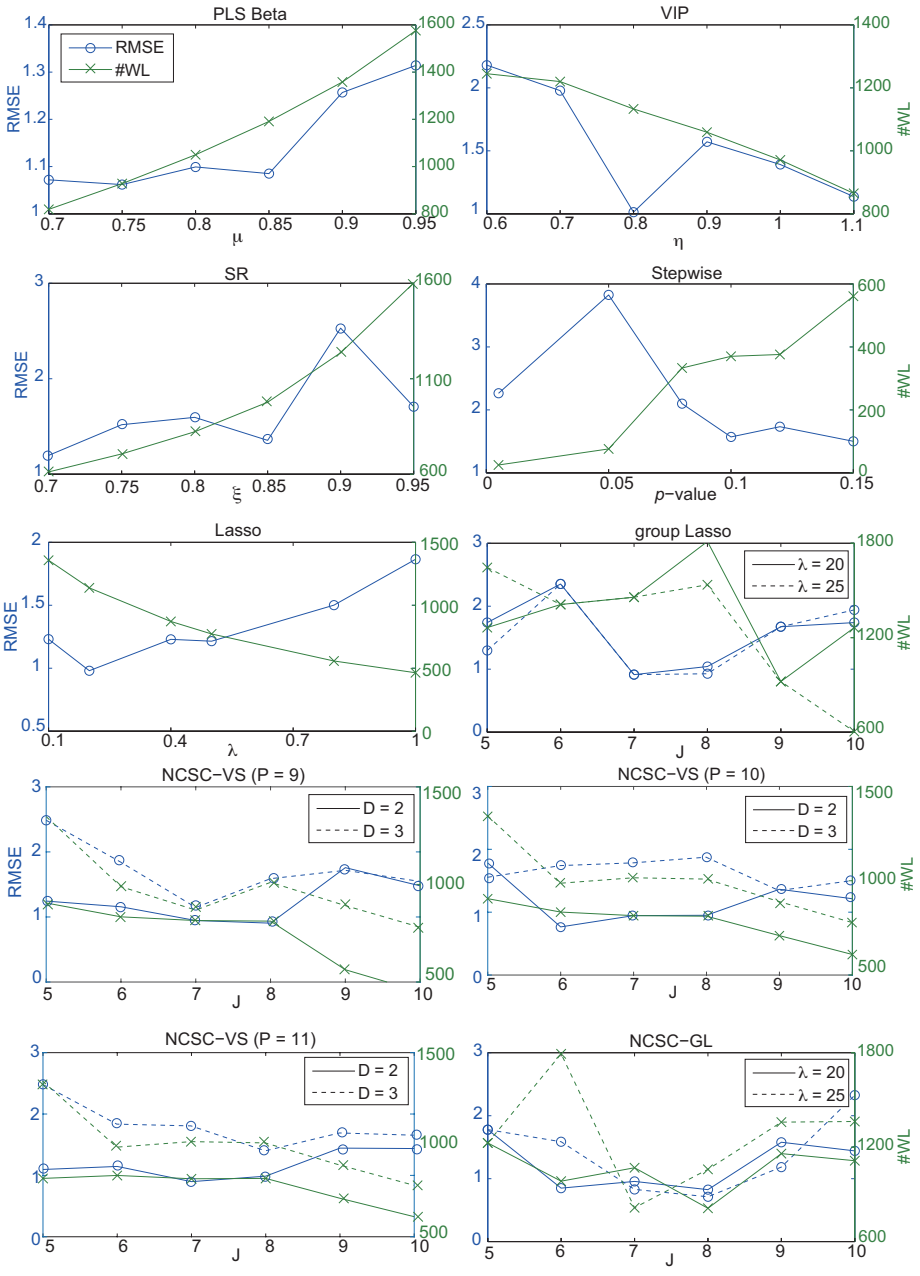

Figure 2: Root-mean-square error (∘) and the number of selected wavelengths (×) as a function of the tuning parameters for each of the methods.

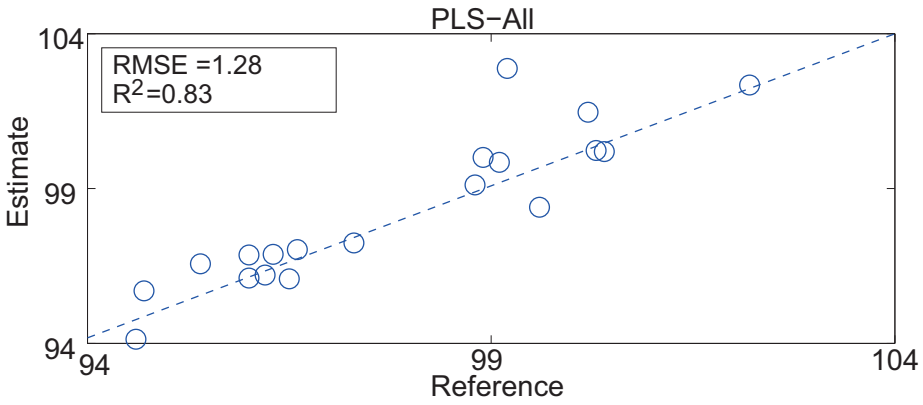

Figure 3: API content estimation result by PLS-All

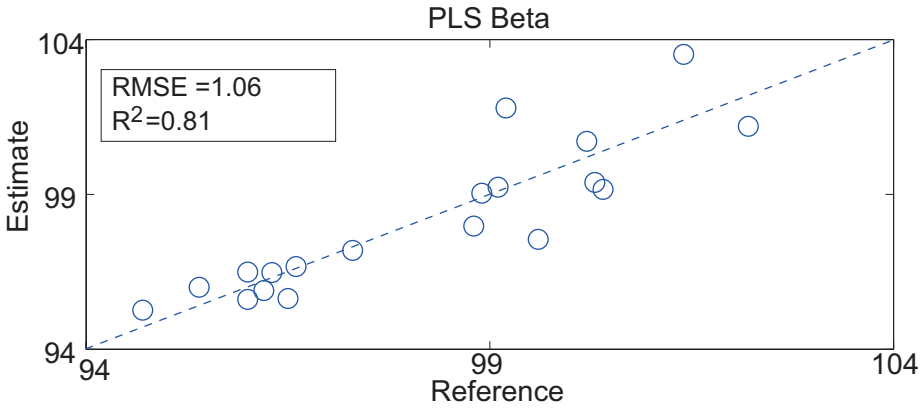

Figure 4: API content estimation result by PLS-Beta

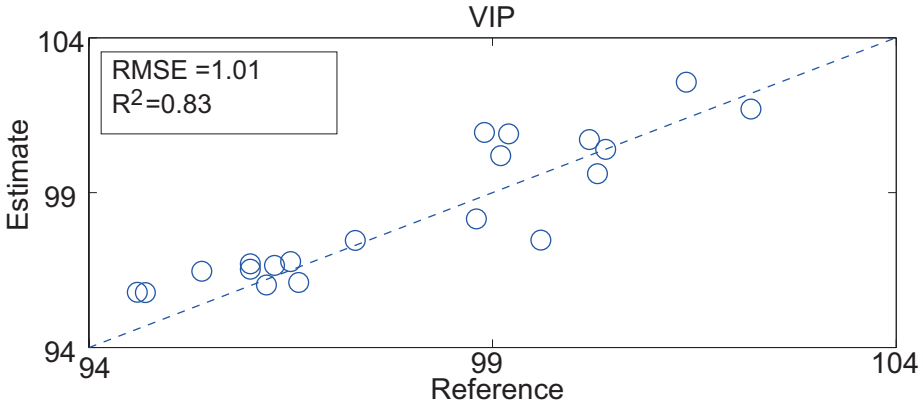

Figure 5: API content estimation result by VIP

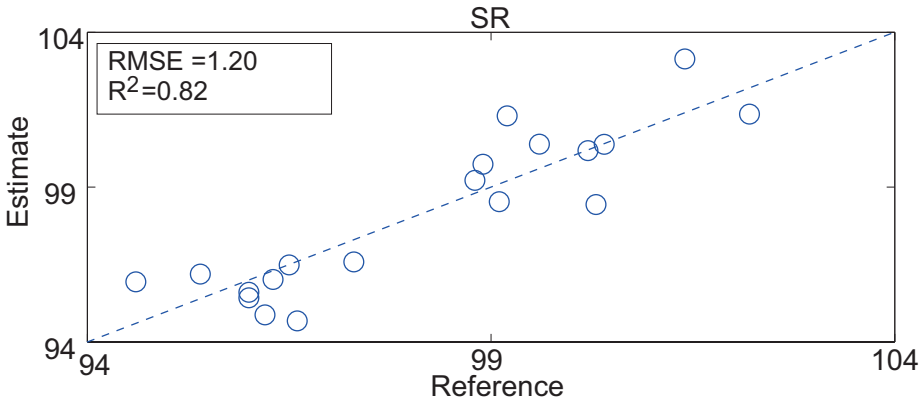

Figure 6: API content estimation result by SR

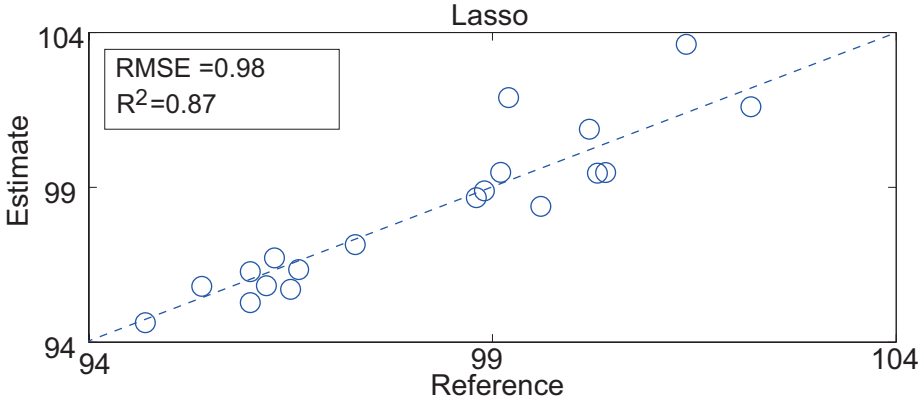

Figure 7: API content estimation result by Lasso

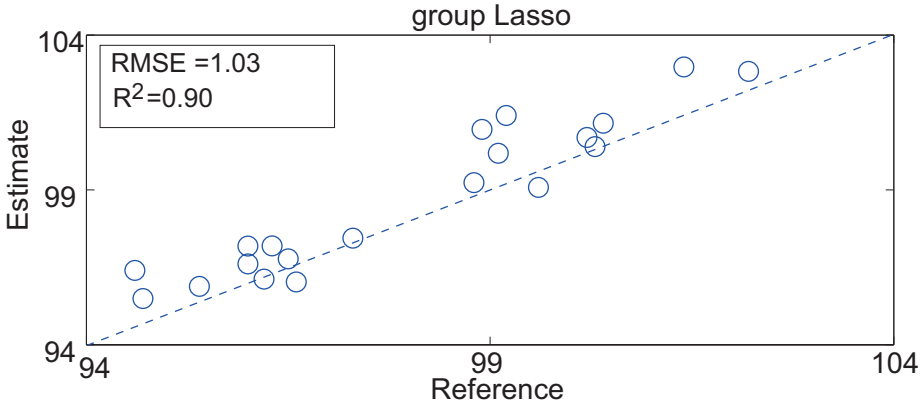

Figure 8: API content estimation result by group Lasso

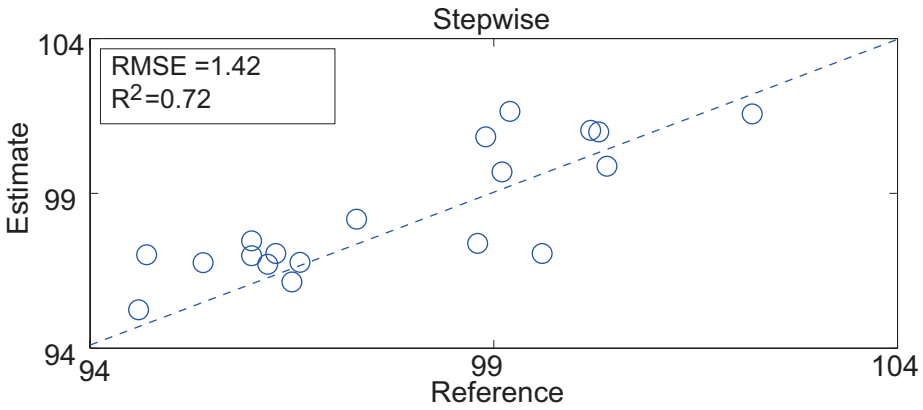

Figure 9: API content estimation result by stepwise

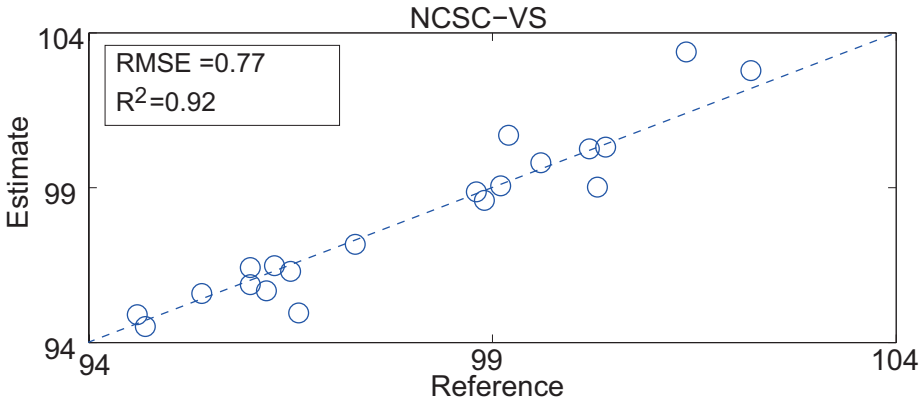

Figure 10: API content estimation result by NCSC-VS

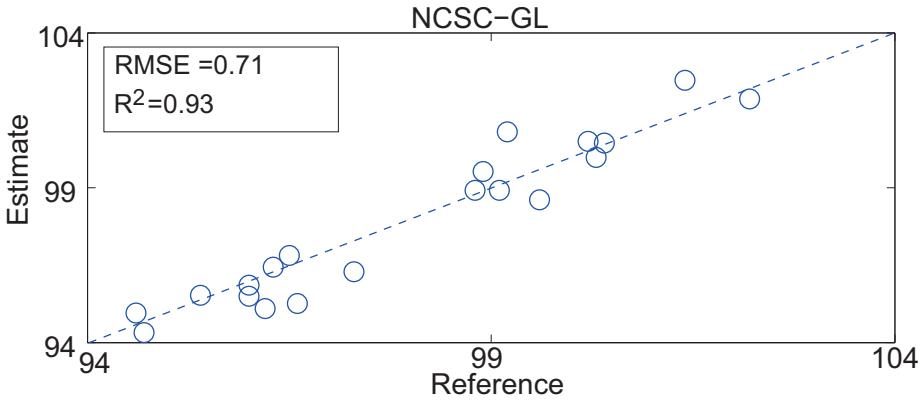

Figure 11: API content estimation result by NCSC-GL

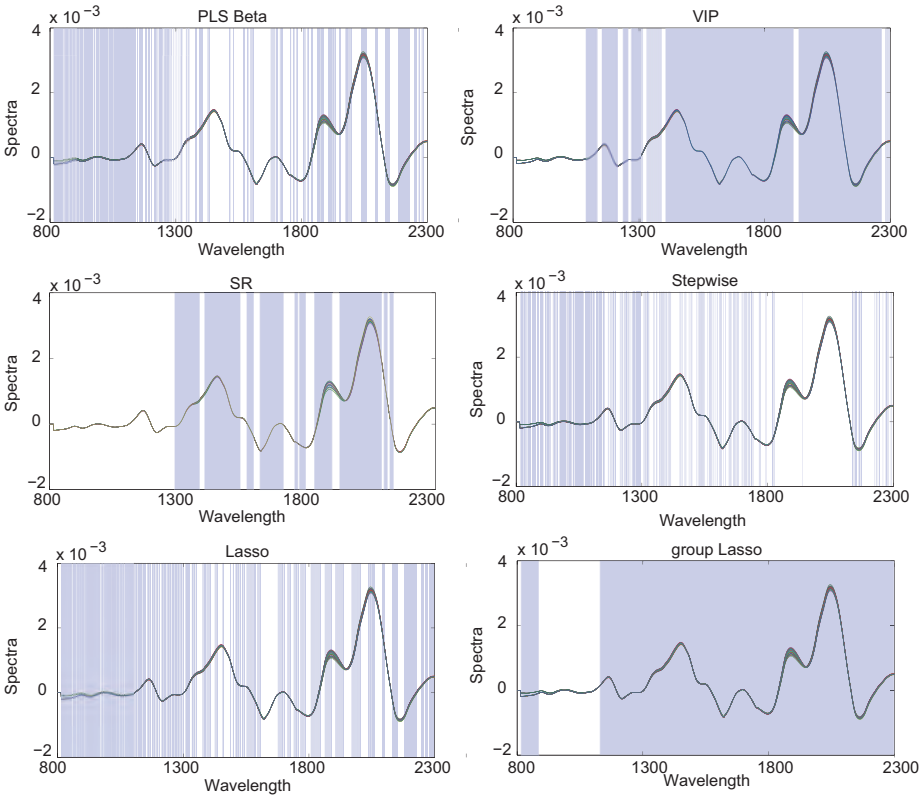

Figure 12: The wavelengths selected by PLS-Beta, VIP, SR, stepwise, Lasso and group Lasso: Colored bands denote the selected wavelengths.

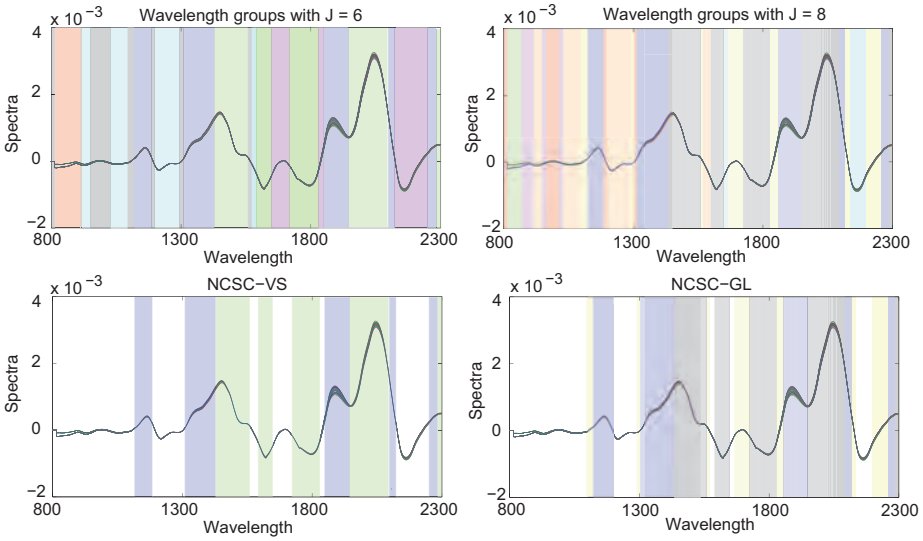

Figure 13: The wavelength groups constructed by NCSC with $J = 6$ (top-left) and $J = 8$ (top-right), and the wavelengths selected by NCSC-VS (bottom-left) and NCSC-GL (bottom-right): The same color bands denote the same wavelength group.

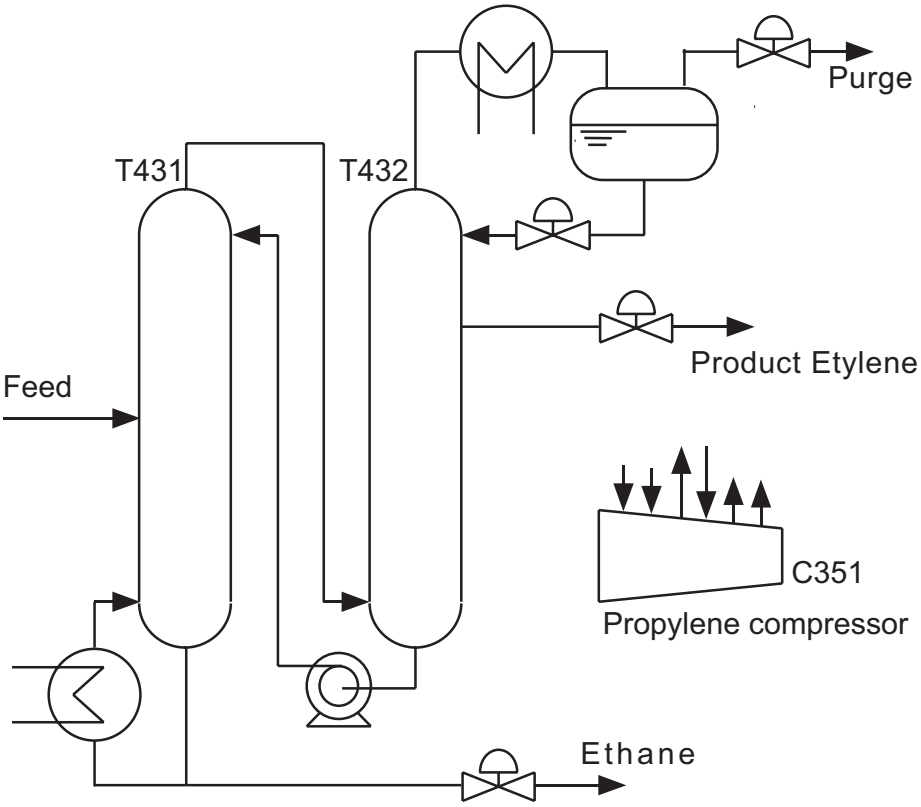

Figure 14: Schematic diagram of the ethylene fractionator at the Showa Denko K.K. (SDK) Oita plant

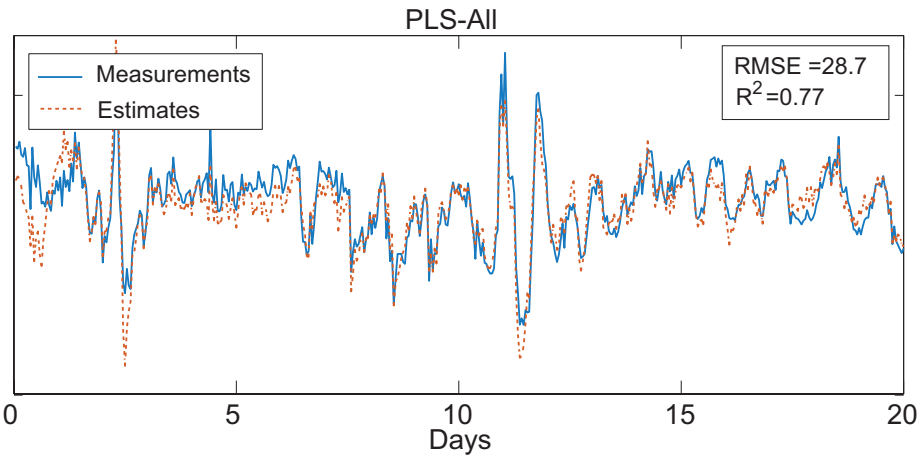

Figure 15: Ethane concentration estimation by PLS-All

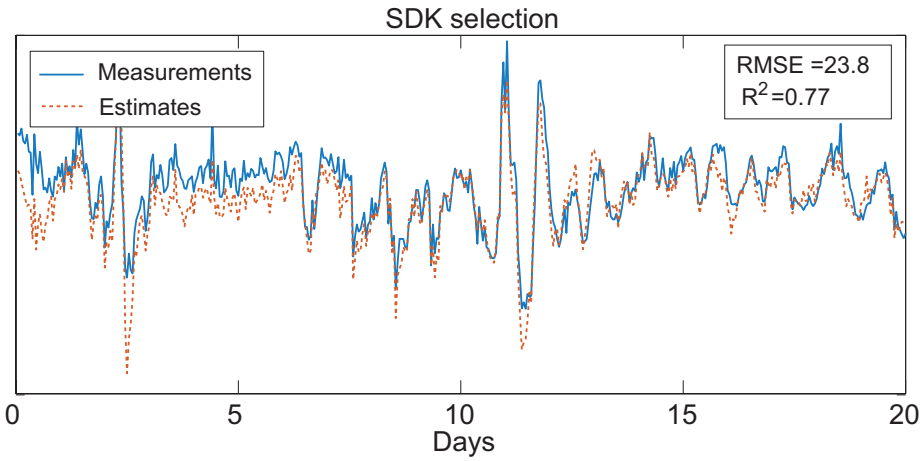

Figure 16: Ethane concentration estimation by SDK selection

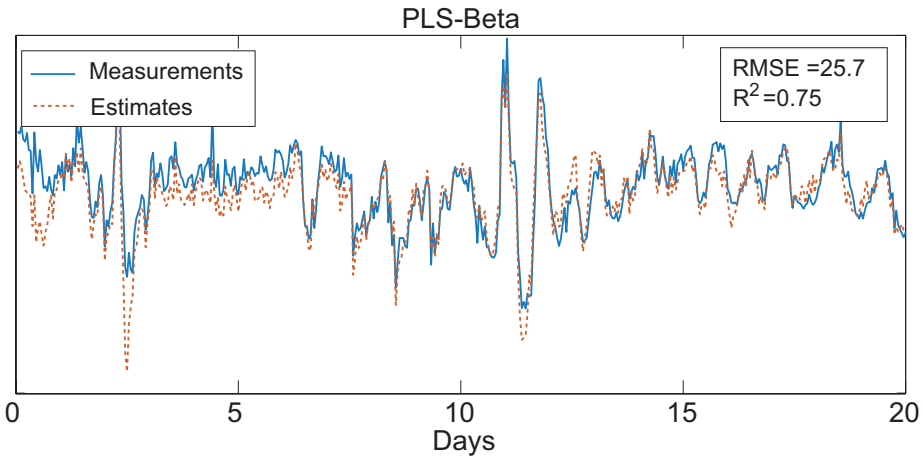

Figure 17: Ethane concentration estimation by PLS-Beta

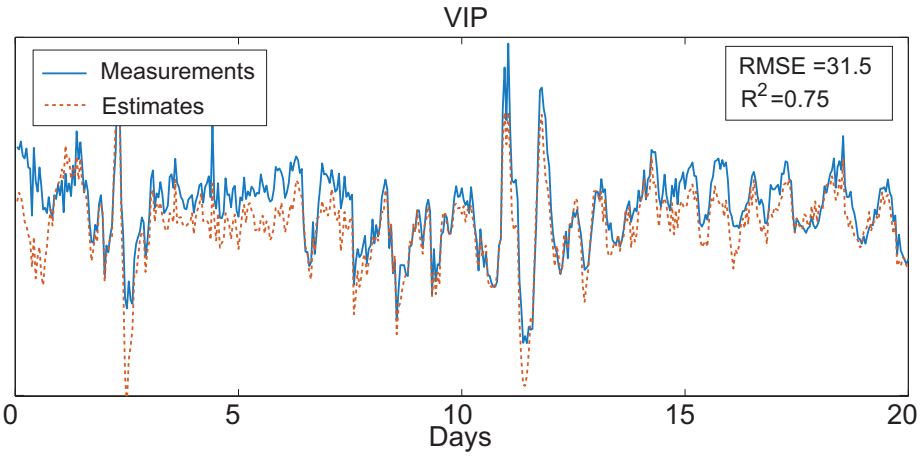

Figure 18: Ethane concentration estimation by VIP

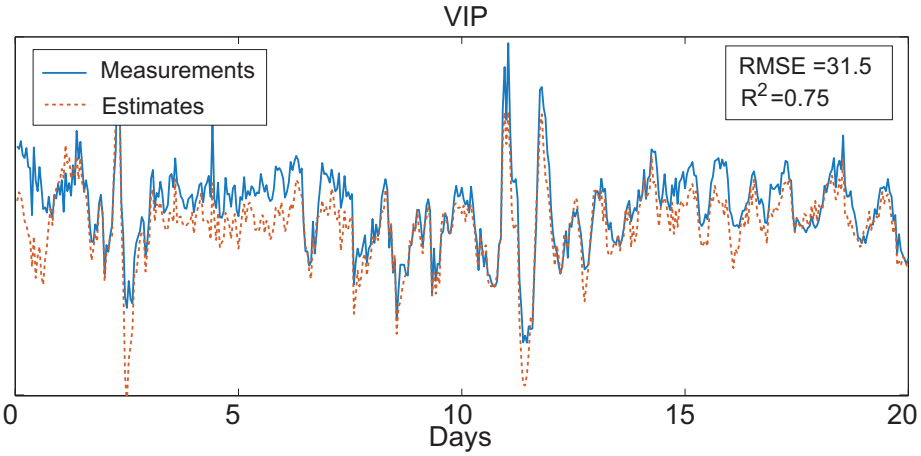

Figure 19: Ethane concentration estimation by VIP

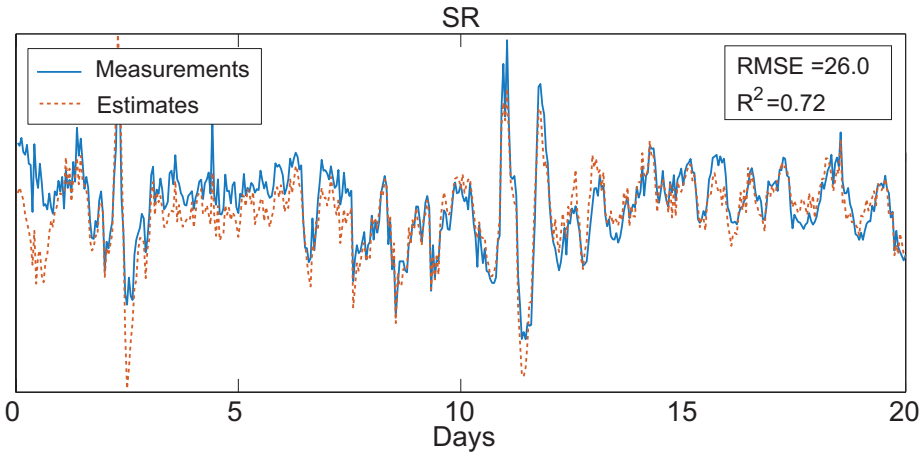

Figure 20: Ethane concentration estimation by SR

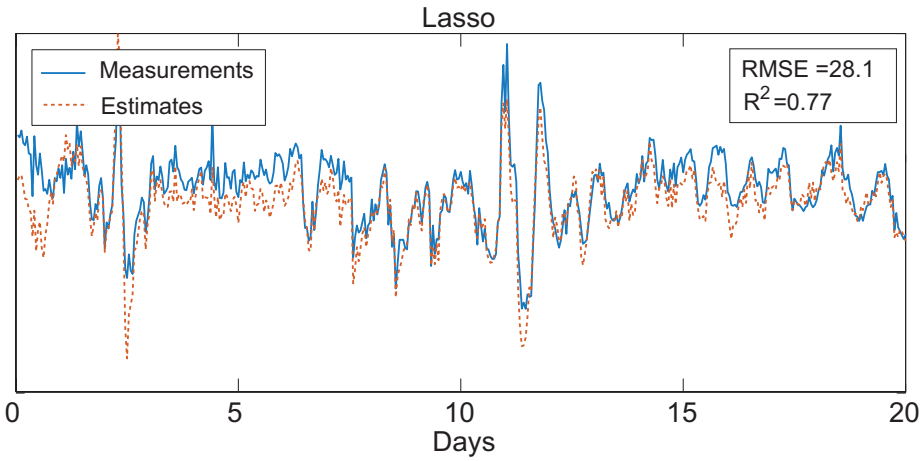

Figure 21: Ethane concentration estimation by Lasso

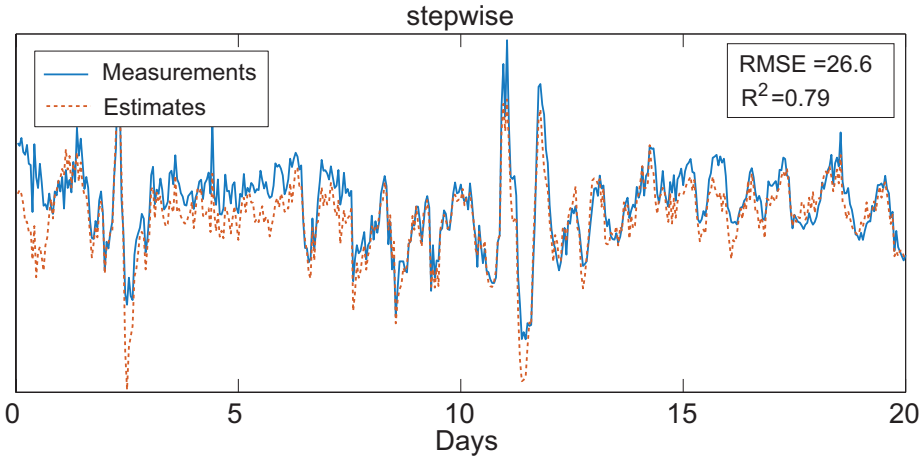

Figure 22: Ethane concentration estimation by stepwise

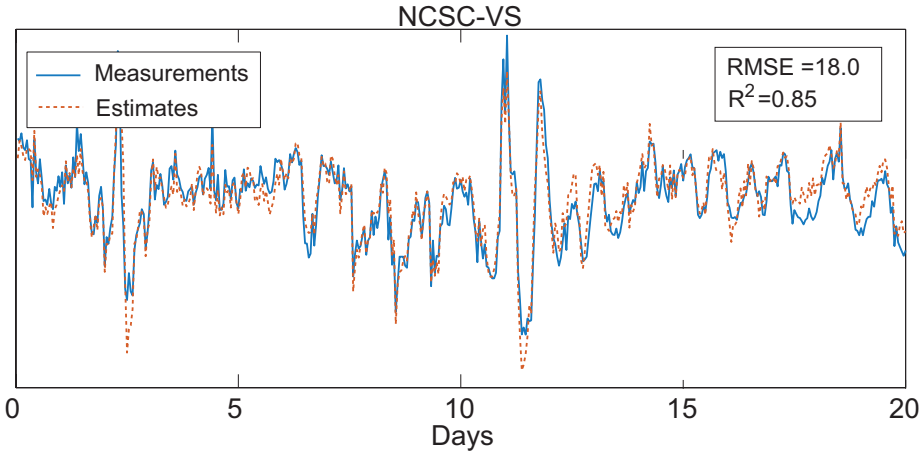

Figure 23: Ethane concentration estimation by NCSC-VS

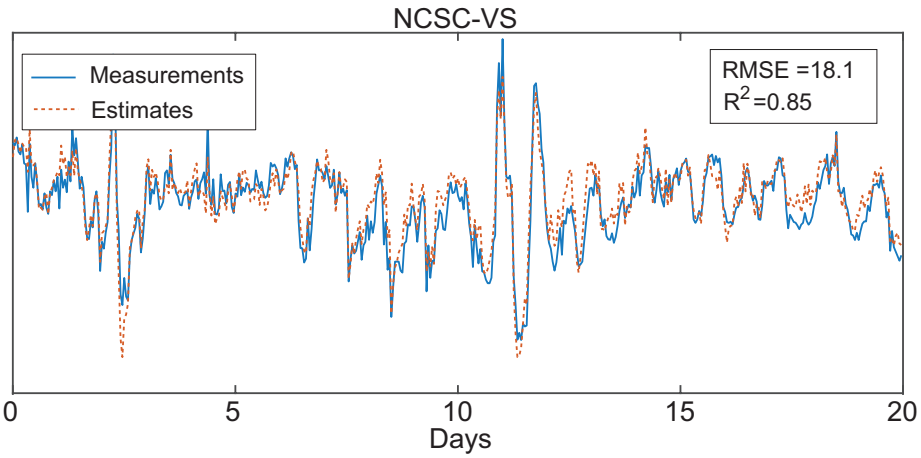

Figure 24: Ethane concentration estimation by NCSC-GL

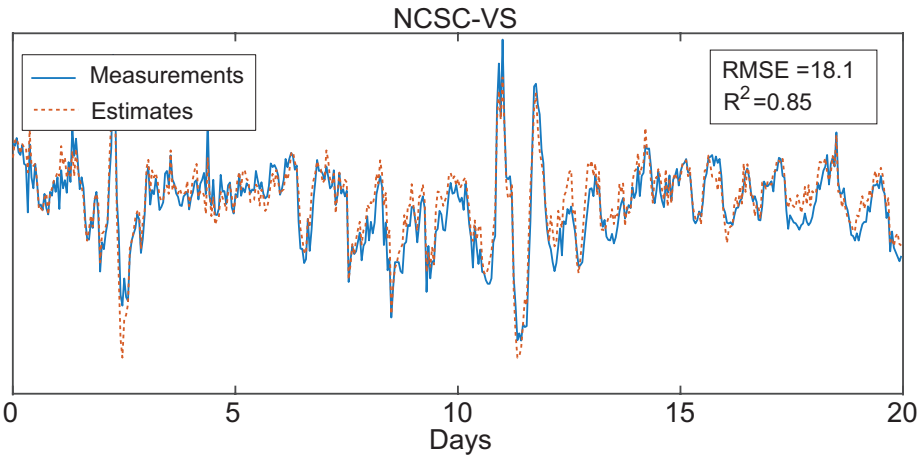

Figure 25: Ethane concentration estimation by NCSC-GL

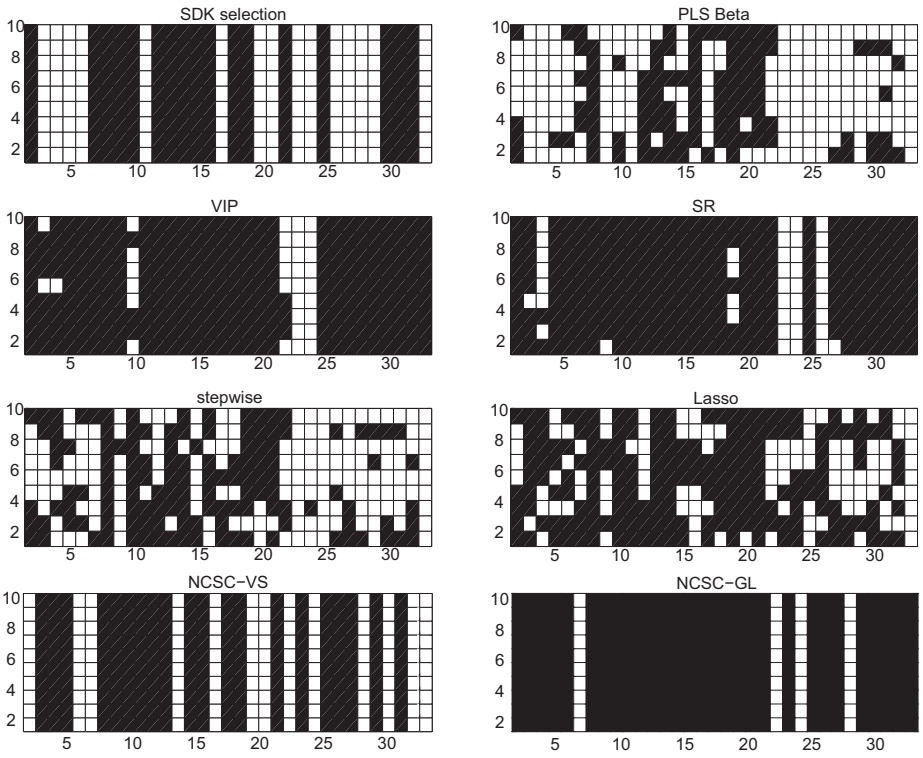

Figure 26: Selected input variables (black: selected and white: not selected)

Appendix A. Analysis of the relaxed problem in the Mcut method

The detailed analysis of the relaxed problem in the Mcut method is described. The objective function of the Mcut method is as follows:

$$\min_{\boldsymbol{q}} \ J_q = \frac{\boldsymbol{q}^T(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{q}}{\boldsymbol{q}^T\boldsymbol{W}\boldsymbol{q}}. \tag{A.1}$$

where $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{W}\boldsymbol{e})$, $\boldsymbol{W}$ is an affinity matrix and $\boldsymbol{e} = [1, \cdots, 1]^T$. $\boldsymbol{q}$ in Eq. (A.1) is scaled as

$$\min_{\tilde{\boldsymbol{q}}} \ \tilde{J}_q = \frac{\tilde{\boldsymbol{q}}^T(\boldsymbol{I}-\tilde{\boldsymbol{W}})\tilde{\boldsymbol{q}}}{\tilde{\boldsymbol{q}}^T\tilde{\boldsymbol{q}}} \tag{A.2}$$

where $\tilde{\boldsymbol{W}} = \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$, $\tilde{\boldsymbol{q}} = \boldsymbol{D}^{1/2}\boldsymbol{q}/|\boldsymbol{D}|^{1/2}$ and $\tilde{\boldsymbol{q}}^T\tilde{\boldsymbol{W}}\tilde{\boldsymbol{q}} > 0$. $\tilde{\boldsymbol{W}}$ is called a probability transition matrix because $0 \leq (\tilde{W})_{i,j} \leq 1$ and $\sum_j (\tilde{W})_{i,j} = 1$ where $(\tilde{W})_{i,j}$ is the element of $\tilde{\boldsymbol{W}}$.

The objective function $\tilde{J}_q$ can be expressed as Rayleigh quotient when $\boldsymbol{P} \equiv \boldsymbol{I} - \tilde{\boldsymbol{W}}$. From Rayleigh's theorem, a quotient $R(\boldsymbol{x})$ is minimized by the eigenvector $\boldsymbol{x}_1$ corresponding to the smallest eigenvalue λ_1, and the minimum value of $R(\boldsymbol{x})$ is λ_1. Using this theorem, the relaxed minimization problem Eq. (A.2) results in the eigenvalue problem.

Suppose that an eigenvector of $\boldsymbol{P}$ is $\boldsymbol{D}^{1/2}\boldsymbol{e}$ and $\boldsymbol{w}_i$ denotes the ith row of $\boldsymbol{W}$, then $(\boldsymbol{D})_{i,i} = d_i = \boldsymbol{w}_i\boldsymbol{e}$ and

$$\begin{aligned} \boldsymbol{P}\boldsymbol{D}^{1/2}\boldsymbol{e} &= \boldsymbol{I}\boldsymbol{D}^{1/2}\boldsymbol{e} - \tilde{\boldsymbol{W}}\boldsymbol{D}^{1/2}\boldsymbol{e} \\ &= \boldsymbol{D}^{1/2}\boldsymbol{e} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}\boldsymbol{D}^{1/2}\boldsymbol{e} = \boldsymbol{0}. \end{aligned} \tag{A.3}$$

Therefore, one of the eigenvectors of $\boldsymbol{P}$ is $\boldsymbol{D}^{1/2}\boldsymbol{e}$ and its eigenvalue is 0. In addition, the largest eigenvalue of $\tilde{\boldsymbol{W}}$ is 1 because it is the probability transition matrix, and $\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x} = \boldsymbol{x}^T(\boldsymbol{I}-\tilde{\boldsymbol{W}})\boldsymbol{x} \geq 0$. That is, $\boldsymbol{P}$ is a positive semidefinite matrix, and $\boldsymbol{D}^{1/2}\boldsymbol{e}$ is the eigenvector $\boldsymbol{z}_1$ corresponding to the smallest eigenvalue $\lambda_1 = 0$.

Although $\boldsymbol{z}_1$ can minimize $\tilde{\boldsymbol{q}}$, all elements of $\boldsymbol{z}_1$ are positive and Eq. (24) is not satisfied. From the max-min theory, the second smallest solution is the second smallest eigenvalue λ_2. The eigenvector $\boldsymbol{z}_2$ corresponding to λ_2 satisfies $\boldsymbol{z}_1^T\boldsymbol{z}_2 = \boldsymbol{0}$ because $\boldsymbol{P}$ is a symmetric matrix. Therefore, the eigenvector $\boldsymbol{z}_2$ has both positive and negative elements, and it is the optimal solution $\tilde{\boldsymbol{q}}^*$ of Eq. (A.2).

From the above analysis, Eq. (A.2) can be rewritten as

$$\min_{\tilde{\boldsymbol{q}}} \tilde{J}_q = \frac{\tilde{\boldsymbol{q}}^T(\boldsymbol{I}-\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2})\tilde{\boldsymbol{q}}}{\tilde{\boldsymbol{q}}^T\tilde{\boldsymbol{q}}} \tag{A.4}$$

and this results in the eigenvalue problem.

$$(\boldsymbol{I}-\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2})\boldsymbol{z} = \lambda\boldsymbol{z} \tag{A.5}$$

The solution $\boldsymbol{q}^*$ is expressed as $\boldsymbol{q}^* = \boldsymbol{D}^{-1/2}\boldsymbol{z}_2$; λ_2 and $\boldsymbol{z}_2$ are called Fiedler value and Fiedler vector, respectively [30].