frontiers in Chemistry

ORIGINAL RESEARCH
published: 22 May 2018
doi: 10.3389/fchem.2018.00171

Nearest Correlation-Based Input Variable Weighting for Soft-Sensor Design

Koichi Fujiwara and Manabu Kano*

Department of Systems Science, Kyoto University, Kyoto, Japan

In recent years, soft-sensors have been widely used for estimating product quality or other important variables when online analyzers are not available. In order to construct a highly accurate soft-sensor, appropriate data preprocessing is required. In particular, the selection of input variables or input features is one of the most important techniques for improving estimation performance. Fujiwara et al. proposed a variable selection method, in which variables are clustered into variable groups based on the correlation between variables by nearest correlation spectral clustering (NCSC), and each variable group is examined as to whether or not it should be used as input variables. This method is called NCSC-based variable selection (NCSC-VS). However, these NCSC-based methods have a lot of parameters to be tuned, and their joint optimization is burdensome. The present work proposes an effective input variable weighting method to be used instead of variable selection to conserve labor required for parameter tuning. The proposed method, referred to herein as NC-based variable weighting (NCVW), searches input variables that have the correlation with the output variable by using the NC method and calculates the correlation similarity between the input variables and output variable. The input variables are weighted based on the calculated correlation similarities, and the weighted input variables are used for model construction. There is only one parameter in the proposed NCVW since the NC method has one tuning parameter. Thus, it is easy for NCVW to develop a soft-sensor. The usefulness of the proposed NCVW is demonstrated through an application to calibration model design in a pharmaceutical process.

Keywords: soft-sensor, calibration model, variable weighting, partial least squares, near infrared spectroscopy

OPEN ACCESS

Edited by:
Hoang Vu Dang,
Hanoi University of Pharmacy, Vietnam

Reviewed by:
Daniel Cozzolino,
Central Queensland University, Australia
Larisa Lvova,
Università degli Studi di Roma Tor Vergata, Italy

****Correspondence:***
Koichi Fujiwara
fujiwara.koichi@i.kyoto-u.ac.jp

Specialty section:
This article was submitted to Analytical Chemistry, a section of the journal Frontiers in Chemistry

Received: *28 February 2018*
Accepted: *30 April 2018*
Published: *22 May 2018*

Citation:
Fujiwara K and Kano M (2018) Nearest Correlation-Based Input Variable Weighting for Soft-Sensor Design. Front. Chem. 6:171. doi: 10.3389/fchem.2018.00171

1. INTRODUCTION

It is important in terms of process safety and quality control to estimate product quality or other process variables, particularly when online analyzers are not available. Soft-sensors are mathematical models for estimating variables that are difficult to measure by hard sensors in real-time from other variables that are easy to measure. They have been used in various industries, for example, measurement of product composition at distillation columns in chemical processes, silicon wafer surface flatness in semiconductor processes, and active ingredient content of drugs in pharmaceutical processes. There are three methodologies for constructing soft-sensors: (i) first-principal modeling based on physicochemical knowledge of processes, (ii) statistical modeling based on process data, and (iii) a combination of the two. These methodologies also are called white-box, black-box, and gray-box modeling, respectively (Ahmad et al., 2014). In particular,

statistical modeling has attracted wide attention due to recent advances in machine learning. Although we can utilize various machine learning techniques for soft-sensor development, partial least squares (PLS) is still widely used in chemometrics as well as soft-sensor design. This is because it is possible to construct an accurate linear regression model even when the multicollinearity problem occurs (Wold et al., 2001; Kano and Ogawa, 2010; Kano and Fujiwara, 2013).

One of the major issues in developing a precise soft-sensor is input variable selection. Although soft-sensors are well-fitted to modeling data when numerous variables are used as the input, their performance may deteriorate when unimportant variables are used for estimation. In particular, input variable selection is a key when a calibration model is constructed from Near-infrared spectroscopy (NIRS) which is a powerful online measurement technology due to its short measuring time and non-invasiveness (Roggo et al., 2007; Miyano et al., 2014). The number of measured wavelengths of an NIR spectrum is usually more than 100.

If all of the possible variable combinations are tested, the computational load increases exponentially as the candidate variables increase. Appropriate variables must be selected in a systematic manner, which is referred to as input variable selection in soft-sensors, and feature selection in machine learning. A technique for input variable selection should be developed for improving the efficiency of soft-sensor design (Andersen and Bro, 2010; Mehmood et al., 2012).

In linear regression, stepwise and least absolute shrinkage and selection operator (Lasso) are widely used as input variable selection methods (Hocking, 1976; Tibshirani, 1996). In addition, PLS-Beta and variable influence on projection (VIP) are available for selecting input variables of PLS (Kubinyi, 1993).

Methods of selecting variables on the basis of correlation have been proposed because the correlation between variables should be considered when building a good regression model (Fujiwara et al., 2009). In correlation-based variable selection methods, variable groups are constructed according to the correlation, some of which are selected as the input variables. Nearest correlation spectral clustering (NCSC) (Fujiwara et al., 2010, 2011) is used for variable grouping. In NCSC-based variable selection (NCSC-VS), variable groups are constructed by NCSC, and it is examined whether or not they should be used as the input variables according to their contribution to the estimates (Fujiwara et al., 2012b). In addition, NCSC-based group Lasso (NCSC-GL) uses group Lasso (Yuan and Lin, 2006; Bach, 2008) for variable group selection after NCSC (Fujiwara and Kano, 2015). Although both NCSC-VS and NCSC-GL can build highly-accurate soft-sensors, tuning their parameters is complicated and time-consuming because they have multiple parameters to be tuned. Therefore, the number of their tuning parameters should be reduced for efficient variable selection.

Another approach is input variable weighting or input variable scaling, which multiplies each input variable by weights according to its importance from the viewpoint of estimation (Kim et al., 2014). The present work proposes an effective input variable weighting method to replace variable selection in order to conserve labor required for parameter tuning. The proposed method, referred to herein as NC-based variable weighting (NCVW), searches input variables that have the correlation with the output variable by using the NC method and calculates the correlation similarity between each input variable and the output variable. The input variables are weighted based on the calculated correlation similarities, and the weighted input variables are used for modeling. Since there is only one parameter in the proposed NCVW, an efficient soft-sensor design is realized. In this work, the usefulness of the proposed NCVW is demonstrated through application to calibration model design for estimating active pharmaceutical ingredient (API) content.

This paper is organized as follows. Section 2 introduces conventional variable selection methods for PLS modeling, and NCVW is proposed in section 3. Section 4 reports on application results of the proposed method to pharmaceutical data. The conclusion and future work are described in section 5.

2. CONVENTIONAL METHODS

This section introduces PLS and conventional input variable selection methods.

2.1. PLS

PLS is a widely used linear regression method in chemometrics as well as soft-sensor design. Given an input data matrix $\boldsymbol{X} \in \Re^{N\times M}$ whose nth row is the nth input sample $\boldsymbol{x}_n \in \Re^M$ and an output data vector $\boldsymbol{y} \in \Re^N$ whose nth element is the nth output sample $y_n \in \Re$, $\boldsymbol{X}$ and $\boldsymbol{y}$ are mean-centered and appropriately scaled. The input $\boldsymbol{X} \in \Re^{N\times M}$ and the output $\boldsymbol{y} \in \Re^N$ are broken down as follows:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{E} \tag{1}$$

$$\boldsymbol{y} = \boldsymbol{T}\boldsymbol{b} + \boldsymbol{f} \tag{2}$$

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 regression coefficient 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.

A PLS model can be constructed by the non-linear iterative partial least squares (NIPALS) algorithm. Let the first to kth latent variables be $\boldsymbol{t}_1, \cdots, \boldsymbol{t}_k$, the loading vectors be $\boldsymbol{p}_1, \cdots, \boldsymbol{p}_k$ and the loading be $b_1, \cdots, b_k$. The $(k+1)$th residual input and output are as follows:

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k - \boldsymbol{t}_r\boldsymbol{p}_k^T \tag{3}$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k - b_k\boldsymbol{t}_k. \tag{4}$$

$\boldsymbol{t}_k$ 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 the eigenvector corresponding the maximum eigenvalue of the following eigenvalue problem:

$$\boldsymbol{X}_{k-1}^T\boldsymbol{y}_{k-1}^T\boldsymbol{y}_{k-1}\boldsymbol{X}_{k-1}\boldsymbol{w}_k = \lambda\boldsymbol{w}_k \tag{5}$$

where λ is an eigenvalue. The kth loading vector $\boldsymbol{p}_k$ and the kth loading b_k are $\boldsymbol{p}_k = \boldsymbol{X}_k^T\boldsymbol{t}_k/\boldsymbol{t}_k^T\boldsymbol{t}_k$ and $b_k = \boldsymbol{y}_k^T\boldsymbol{t}_k/\boldsymbol{t}_k^T\boldsymbol{b}_k$.

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

2.2. PLS-Beta

PLS-Beta translates a PLS model, Equations (1, 2), into a multiple linear regression (MLR) model and selects input variables based on the magnitude of its regression coefficients (Kubinyi, 1993). The translated model is expressed as

$$\hat{\boldsymbol{y}} = \boldsymbol{T}(\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{T}\boldsymbol{y} = \boldsymbol{X}\beta_{pls} \tag{6}$$

where $\beta_{pls} = \boldsymbol{W}(\boldsymbol{P}^T\boldsymbol{W})^{-1}(\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{y}$, and $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K] \in \Re^{M\times K}$. The evaluation index of PLS-Beta ν is defined as

$$\nu = \frac{||\beta_{select}||}{||\beta_{pls}||} \ (0 < \nu \leq 1) \tag{7}$$

where β_{select} is the regression coefficient vector of the selected input variables. We select individual input variables in descending order of the magnitude of β_{pls} until ν achieves a predefined threshold.

2.3. Variable Influence on Projection (VIP)

The VIP evaluates the contribution of each input variable to the output (Kubinyi, 1993). The VIP score of the jth input variable is

$$V_j = \sqrt{M\sum_{k=1}^{K}\left(w_{jk}^2 b_k^2(\boldsymbol{t}_k^T\boldsymbol{t}_k)/||\boldsymbol{w}_k||^2\right)\Big/\sum_{k=1}^{K} b_k^2(\boldsymbol{t}_k^T\boldsymbol{t}_k)} \tag{8}$$

where w_{jk} is the jth element of $\boldsymbol{w}_k$. Variables satisfying $V_j > \eta$ (> 0) are selected.

2.4. Stepwise

Stepwise is an input variable selection method for the MLR model based on a statistical test which checks whether or not the true value of the regression coefficient of a newly added candidate variable is zero (Hocking, 1976).

2.5. Least Absolute Shrinkage and Selection Operator (Lasso)

Lasso is least squares with L_1 regularization so that some regression coefficients approach zero (Tibshirani, 1996). The objective function of Lasso is as follows:

$$\beta_{lasso} = \arg\min_{\beta}\left(||\boldsymbol{y} - \boldsymbol{X}\beta||_2^2 + \lambda||\beta||_1\right), \ \lambda \ (> 0) \tag{9}$$

Least angle regression (LARS) solves the problem of Equation (9) efficiently (Efron et al., 2004).

3. NEAREST CORRELATION BASED VARIABLE WEIGHTING (NCVW)

The present work proposes a new method for weighting input variables for PLS modeling to be used instead of variable selection. Since the proposed method uses the nearest correlation (NC) method for calculating correlation-based variable weights, this section explains the NC method and variable selection methods based on the NC method before the proposed method is described.

3.1. NC Method

The NC method was originally developed as an unsupervised learning technique for detecting samples whose correlation is similar to the query (Fujiwara et al., 2012a). The procedure of the NC method is described in Algorithm 1.

Algorithm 1 Nearest correlation (NC) method

1: Prepare $\boldsymbol{x}_n (n = 1, \cdots N)$ and $\boldsymbol{x}_q$.
2: Set γ.
3: **for all** $n = 1, 2, \cdots, N$ $(n \neq q)$ **do**
4: $\quad \boldsymbol{x}'_n = \boldsymbol{x}_n - \boldsymbol{x}_q$.
5: **end for**
6: **for all** k, l $(k \neq l)$ **do**
7: $\quad$ Calculate $C'_{k,l}$ from $\boldsymbol{x}'_k$ and $\boldsymbol{x}'_l$.
8: $\quad$ **if** $|C'_{k,l}| \geq \gamma$ **then**
9: $\quad\quad$ Output $\boldsymbol{x}_k$ and $\boldsymbol{x}_l$ as similar samples to $\boldsymbol{x}_q$
10: $\quad$ **end if**
11: **end for**

The concept of Algorithm 1 is explained through a simple example. In **Figure 1** (left), there are seven samples $\boldsymbol{x}_q, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_6$, of which five $\boldsymbol{x}_q$ and $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_4$ are on the same plane P. That is, plane P expresses the hidden correlation between the five samples and $\boldsymbol{x}_5$ and $\boldsymbol{x}_6$ have a different correlation. The aim of the NC method here is to detect samples whose correlation is similar to the query $\boldsymbol{x}_q$, that is, to detect $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_4$ on P.

In steps 3–5, the entire space is translated so that $\boldsymbol{x}_q$ becomes the origin by subtracting $\boldsymbol{x}_q$ from all other samples $\boldsymbol{x}_n$ as shown in **Figure 1** (right). The translated plane P becomes the linear subspace V since it contains the origin.

Draw lines connecting each sample and the origin, and check whether another sample is on the line in steps 6–8. In this example, pairs $\boldsymbol{x}_1$-$\boldsymbol{x}_4$ and $\boldsymbol{x}_2$-$\boldsymbol{x}_3$ satisfy such a relationship, and $\boldsymbol{x}_5$ and $\boldsymbol{x}_6$, which are not on V, cannot make pairs. At this time, the correlation coefficients of these pairs must be 1 or -1. Thus, the pairs whose correlation coefficients are ± 1 are thought to have a correlation similar to $\boldsymbol{x}_q$. The threshold of the correlation coefficient γ $(0 < \gamma \leq 1)$ is used for constraint relaxation. Steps 6–8 correspond to the above procedure.

Finally, the pairs whose correlations are similar to the query $\boldsymbol{x}_q$ are output in step 9.

3.2. NCSC

NCSC was originally proposed for sample clustering based on correlation between variables (Fujiwara et al., 2010, 2011), in which the NC method and spectral clustering (SC) (Ding et al., 2001; Ng et al., 2002) are integrated. SC is a graph theory-based clustering method, which can partition a weighted graph, whose weights express affinities between nodes, into subgraphs by cutting some of their arcs. In NCSC, the NC method is

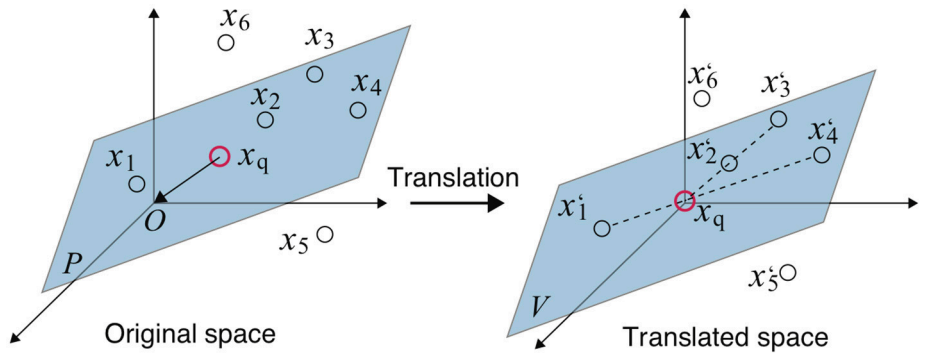

FIGURE 1 | Example of the NC method: A red circle is the query $\boldsymbol{x}_q$. Dashed lines denote pairs of samples, whose correlation is similar to $\boldsymbol{x}_q$ (left) (Fujiwara et al., 2010, 2012a,b).

used for building an affinity graph expressing the correlation-based similarities between samples, and SC partitions the graph constructed by the NC method.

Algorithm 2 shows an affinity matrix construction procedure in NCSC. Steps 6–13 correspond to the NC method, and the weighted graph constructed by the NC method is expressed as an affinity matrix $\boldsymbol{S}$. Although some SC algorithms have been proposed, the max-min cut (Mcut) algorithm (Ding et al., 2001) or its extended method (Ng et al., 2002) is used herein.

Algorithm 2 Affinity matrix construction

1: Set γ and J.
2: $\boldsymbol{S} \in \Re^{N\times N} \leftarrow \boldsymbol{O}_{N,N}$.
3: $L = 1$.
4: **for** $L = 1$ to N **do**
5: $\quad \boldsymbol{S}_L \in \Re^{N\times N} \leftarrow \boldsymbol{O}_{N,N}$.
6: $\quad$ **for all** $n = 1, 2, \cdots, N$ $(n \neq L)$ **do**
7: $\quad\quad \boldsymbol{x}'_n = \boldsymbol{x}_n - \boldsymbol{x}_L$.
8: $\quad$ **end for**
9: $\quad$ **for all** k, l $(k \neq l)$ **do**
10: $\quad\quad$ Calculate $C'_{k,l}$ from $\boldsymbol{x}'_k$ and $\boldsymbol{x}'_l$.
11: $\quad\quad$ **if** $|C'_{k,l}| \geq \gamma$ **then**
12: $\quad\quad\quad (\boldsymbol{S}_L)_{k,l} = (\boldsymbol{S}_L)_{l,k} = 1$.
13: $\quad\quad$ **end if**
14: $\quad$ **end for**
15: $\quad \boldsymbol{S} = \boldsymbol{S} + \boldsymbol{S}_L$.
16: **end for**

NCSC has two parameters: the threshold in the NC method γ and the number of clusters partitioned by SC, J. Previous studies have suggested the default value of γ to be 0.99 (Fujiwara et al., 2010, 2011), and that J needs to be determined by trial and error.

3.3. NCSC-VS and NCSC-GL

NCSC has been utilized for variable selection in soft-sensor design. In these methods, multiple variable groups are constructed by NCSC, of which some are selected as the input variables of a soft-sensor. NCSC classifies variables into J variable groups $\boldsymbol{v}_j = \{x_m \mid m \subset \mathcal{V}_j\}$ $(j = 1, \cdots, J)$, where $\mathcal{V}_j$ is the subset of variable indexes and $\mathcal{V} = \cup\mathcal{V}_j$. An affinity matrix is derived from the transposed input variable matrix $\boldsymbol{X}^T$ by the NC method for variable grouping.

NCSC-VS evaluates each variable group as to whether or not its members should be used as input variables from the viewpoint of contribution to the output (Fujiwara et al., 2012b). The jth PLS model with the number of latent variables P, f_j^P, is built from the jth variable group matrix $\boldsymbol{X}_j$, and its contribution is evaluated by

$$C_j^P = 1 - \frac{||\hat{\boldsymbol{y}}_j^P||^2}{||\boldsymbol{y}||^2} \tag{10}$$

where $\hat{\boldsymbol{y}}_j^P$ is the estimate of f_j^P. We select D $(\leq J)$ variable groups in descending order of C_j^P and construct the final PLS model from the selected input variables.

NCSC-GL selects variable groups by using group Lasso instead of contribution evaluation in NCSC-VS. Group Lasso is an extension of Lasso for selecting some input variable groups from predefined multiple variable groups (Yuan and Lin, 2006; Bach, 2008).

Suppose that M variables are divided into J groups; and $\boldsymbol{X}_j$ and β_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 regression coefficients of group Lasso is derived as:

$$\beta_{glasso} = \arg\min_{\beta} \left(||\boldsymbol{y} - \sum_{j=1}^{J} \boldsymbol{X}_j\beta_j||_2^2 + \lambda \sum_{j=1}^{J} \sqrt{M_j}||\beta_j||_2 \right) \tag{11}$$

where $\beta = [\beta_1^T, \cdots, \beta_J^T]^T$, and λ is a parameter. Variable groups must be constructed in advance in group Lasso. Thus, NCSC-GL uses variable groups formed by NCSC as the input of group Lasso.

NCSC-VS has four tuning parameters: γ in the NC method, the number of variable groups partitioned by SC, J, latent variables in the PLS models for variable group evaluation, P, and selected variable groups, D. On the other hand, there are three tuning parameters in NCSC-GL: γ in the NC method, the number of variable groups J formed by SC and λ in group Lasso. These three or four parameters need to be tuned for appropriate

input variable selection. However, their joint optimization is burdensome and time-consuming. For more efficient soft-sensor design, the number of tuning parameters should be reduced.

3.4. NCVW

A new input variable weighting method, referred to as NC-based variable weighting (NCVW), is proposed to be used instead of variable selection for conserving labor required for parameter tuning. The proposed method applies the NC method to the input variables and output variable together for calculating similarities based on the correlation between the input variables and output variable, and uses the input variables weighted by the calculated similarities for modeling.

Let the nth input sample and the nth output sample are $\boldsymbol{x}_n \in \Re^M$ and y_n, where M denotes the number of input variables. In NCVW, the NC method is applied to extended samples

$$\boldsymbol{x}'_n = [x_n^{[1]}, \cdots, x_n^{[M]}, y_n]^T \ (n = 1, \cdots, N) \tag{12}$$

and the affinity matrix $\boldsymbol{S}'$ is constructed. Next, the 1st to Mth element in the $(M+1)$th column of $\boldsymbol{S}$ which corresponds to the output variable is extracted as a weighting vector $\boldsymbol{w} = [w^{[1]}, \cdots, w^{[M]}]$. Finally, a new input variable for PLS modeling is formed as

$$\boldsymbol{z}_n = \boldsymbol{w} \circ \boldsymbol{x} = [w^{[1]}x^{[1]}, \cdots, w^{[M]}x^{[M]}]^T. \tag{13}$$

where $\boldsymbol{a} \circ \boldsymbol{b}$ denotes an element-wise product between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. Algorithm 3 summarizes the procedure of the proposed NCVW.

Algorithm 3 Nearest correlation based variable weighting (NCVW)

1: Prepare $\boldsymbol{x}_n$ and y_n $(n = 1, \cdots N)$.
2: $\boldsymbol{x}_n \longleftarrow [x_n^{[1]}, \cdots, x_n^{[M]}, y_n]^T \ (n = 1, \cdots, N)$
3: Get $\boldsymbol{S} \in \Re^{(M+1)\times(M+1)}$ by applying Algorithm 2 to $\boldsymbol{x}_n$.
4: Extract the 1st to Mth element in the $M+1$th column of $\boldsymbol{S}$ as $\boldsymbol{w} = [w^{[1]}, \cdots, w^{[M]}]$.
5: $\boldsymbol{z}_n = \boldsymbol{w} \circ \boldsymbol{x} = [w^{[1]}x^{[1]}, \cdots, w^{[M]}x^{[M]}]^T \ (n = 1, \cdots N)$.
6: Construct a model from $\boldsymbol{z}_n$ by PLS.

In soft-sensor design, the correlation among multiple input variables needs to be considered as well as the correlation between an individual input variable and the output variable. Thus, the proposed NCVW does not evaluate the correlation between each input variable and the output variable, but the correlation of multiple input variables together, which may contribute to an improvement in the estimation performance of a soft-sensor. In addition, the proposed NCVW has only one parameter, which is the threshold of the NC method γ. This leads to a huge efficiency improvement of soft sensor development.

4. CASE STUDY

This case study evaluates the performance of the proposed NCVW through application to pharmaceutical data provided by Daiichi Sankyo Co., Ltd. (Kim et al., 2011).

4.1. Objective Data

The objective of this case study is to design a calibration model that estimates active pharmaceutical ingredient (API) content in a target drug. NIR spectra (2203 points in 800−2500 nm) and the API content were measured from the granules of the drug through experiments. Since the number of wavelengths in NIR spectra was large, appropriate input wavelengths of NIR spectra had to be selected for constructing a precise calibration model. The modeling data and validation data consisted of 576 and 20 samples, respectively.

4.2. Model Construction

Before modeling, a first-order differential Savitzky-Golay smoothing filter (Savitzky and Golay, 1964) was applied to the spectra. As a benchmark, a PLS model using all the wavelengths as the input was constructed, which was called PLS-All. The number of its adopted latent variables was determined by cross-validation. Input wavelengths were selected using PLS-Beta, VIP, stepwise, Lasso, NCSC-VS, and NCSC-GL. Parameters used in each method were selected by trial and error, which are shown in **Table 1**. We calculated the root-mean-square error (RMSE) for the modeling data in each parameter and determined the optimal wavelengths based on the calculated RMSE.

We designed PLS models with the wavelengths selected by each method in which cross-validation was used for determining the appropriate number of latent variables. Although Lasso derives regression coefficients, the PLS model was built from the wavelengths whose regression coefficient was not zero. This is for the reason that the number of retained wavelengths was still large and dimension reduction by PLS may have been needed. On the other hand, in the proposed NCVW, we calculated variable weights and constructed the PLS model from the weighted wavelengths. Finally, the API content was estimated by these constructed PLS models.

These procedures were repeated 100 times for calculating average CPU time per one modeling of each method. The computer configuration was as follows: OS: Windows10 (64bit),

TABLE 1 | Tested parameters.

	Parameters
PLS-All	–
PLS-Beta	$\nu = \{0.70, 0.75, 0.80, 0.85, 0.90, 0.95\}$
VIP	$\eta = \{0.6, 0.7, 0.8, 0.9, 1.0, 1.1\}$
Lasso	$\lambda = \{0.1, 0.2, 0.4, 0.5, 0.8, 1.0\}$
Stepwise	$\bar{\rho} = \{0.005, 0.05, 0.08, 0.1, 0.12, 0.15\}$
NCSC-VS	$\gamma = 0.99$
	$J = \{5, 6, 7, 8, 9, 10\}$
	$P = \{9, 10, 11\}$
	$D = \{2, 3\}$
NCSC-GL	$\gamma = 0.99$
	$J = \{5, 6, 7, 8, 9, 10\}$
	$\lambda = \{20, 25\}$
NCVW	$\gamma = 0.99$

CPU: Intel Core i7-8700 (3.2 GHz×6), RAM: 64G bytes, and MATLAB 2018a.

Table 2 summarizes the results of the case study. #Wavelength and #LV mean the numbers of selected wavelengths and adopted latent variables determined by cross-validation, R^2 is the determination coefficient, "CPU time" is the average CPU times [s], and "Parameters" denotes the optimal parameters in each method. In addition, **Figure 2** shows the detailed estimation results.

TABLE 2 | API content estimation results.

	#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	1.52
VIP	1133	19	$\eta = 0.8$	1.01	0.83	0.36
Lasso	1138	39	$\lambda = 0.2$	0.98	0.87	0.17
stepwise	561	24	$\tilde{p} = 0.15$	1.42	0.72	1.64
NCSC-VS	843	25	$\gamma = 0.99, J = 6, P = 10, D = 2$	0.77	0.92	202.39
NCSC-GL	1059	18	$\gamma = 0.99, J = 8, \lambda = 25$	0.71	0.93	204.04
NCVW	2203	15	$\gamma = 0.99$	0.74	0.92	202.27

While PLS-Beta, VIP, and Lasso improved the estimation performance compared to PLS-All, only stepwise was worse than PLS-All. Both NCSC-VS and NCSC-GL achieved higher performance than methods above; and, in particular, NCSC-GL had the best performance. The proposed NCVW achieved almost the same performance as NCSC-VS and NVSC-GL, even though NCVW has only one tuning parameter. RMSE of NCVW was improved by about 42% in comparison with PLS-All.

It is concluded that the proposed NCVW is a tuning-free soft-sensor design technique and that its performance is comparable to the NCSC-based methods.

4.3. Discussion

According to **Table 2**, the CPU time of NCSC-VS, NCSC-GL, and the proposed NCVW were much longer than those of other methods. NCSC occupied more than 99% of their CPU time since it uses iteration for similarity calculation, which means NCVW does not improve the computational load. In addition, the estimation performance of NCVW was not improved in

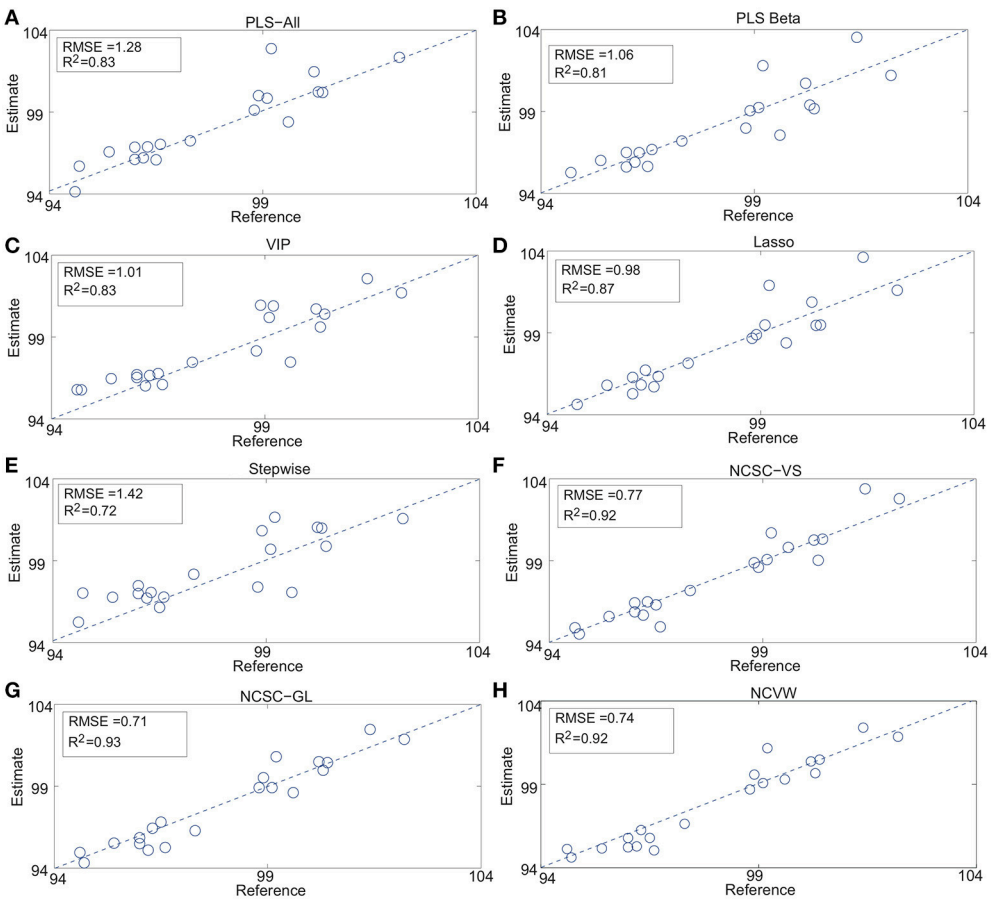

FIGURE 2 | API content estimation results: **(A)** PLS-All, **(B)** PLS Beta, **(C)** VIP, **(D)** Lasso, **(E)** Stepwise, **(F)** NCSC-VS, **(G)** NCSC-GL and, **(H)** NCVW (Fujiwara et al., 2010, 2012a,b).

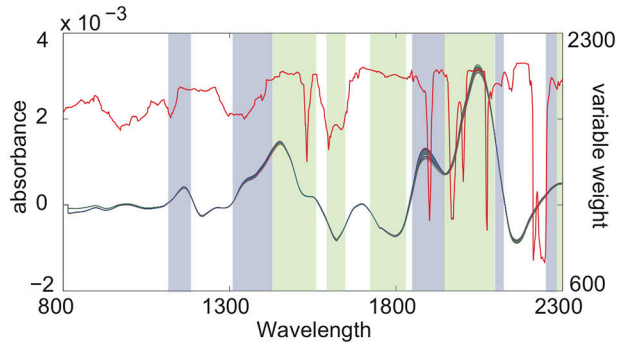

FIGURE 3 | Wavelength group selection by NCSC-VS and variable weights by NCVW.

comparison with NCSC-GL; however, construction of the actual soft-sensor therewith is much easier than NCSC-VS and NCSC-GL. The latter methods respectively have four and three tuning parameters. In this case study, 36 calculations in NCSC-VS and 12 calculations in NCSC-GL were repeated for searching the best parameter combination according to **Table 1**. It becomes difficult to find the optimal parameter combination when the number of tuning parameters increases. On the other hand, NCVW has just one parameter–the threshold of the NC method γ and its recommended value has been proposed to be $\gamma = 0.99$ (Fujiwara et al., 2010, 2011). In fact, the total computation times of NCSC-VS, NCSC-GL, and the proposed NCVW were about 121, 42, and 3 min, respectively, for parameter tuning in this case study. Thus, the proposed NCVW makes the soft-sensor design much more efficient than NCSC-VS and NCSC-GL.

Variable weighting based on another type of the weight, the correlation coefficient between each input variable and the output variable, was evaluated. This method is called correlation coefficient-based variable weighting (CCVW). The mth variable weight of CCVW is defined as follows:

$$c^{[m]} = \frac{\boldsymbol{y}^T \boldsymbol{x}^{[m]}}{||\boldsymbol{y}||||\boldsymbol{x}^{[m]}||} \tag{14}$$

where $\boldsymbol{x}^{[m]} \in \Re^N$ denotes the mth column in the input data matrix $\boldsymbol{X} \in \Re^{N \times M}$ and $\boldsymbol{y} \in \Re^N$ is the output data vector. A PLS model was constructed from the input variables weighted by $c^{[m]}$. RMSE and R^2 of NCVW were 1.34 and 0.84, respectively. This showed the effectiveness of the variable weight by NCVW which consider the correlation of multiple input variables and the output variable together.

Figure 3 shows the results of wavelength selection of NCSC-VS and the variable weights calculated by the proposed NCVW. The colored bands express the selected wavelengths, and the colors denote groups by NCSC-VS. The red line is the weights of NCVW. The wavelength groups selected by NCSC-VS contained almost only specific peaks. On the other hand, in NCVW, the weights of almost all wavelength regions that contain peaks, were large while some peaks had small weights. This is consistent with the physicochemical knowledge that information about compounds is contained in specific peaks. Some peaks might have important information about the API content, and other peaks might not contribute to API content estimation. Therefore, the weights by NCVW suggest that unnecessary peaks for API content estimation exist in NIR spectra. This indicates that NCVW can create meaningful weights for soft-sensor design.

5. CONCLUSION

In the present work, an input variable weighting method was proposed for efficient and highly-accurate soft-sensor design. The proposed NCVW derives the variable weights on the basis of the correlation between the input variables and output variable by utilizing the NC method and builds a PLS model from the weighted input variables. Since NCVW has just one tuning parameter, its soft-sensor design is efficient. The performance of NCVW was evaluated through the case study of calibration model development of the pharmaceutical process. The result showed that the estimation performance of NCVW was comparable to that of NCSC-VS and NCSC-GL, while the labor required for parameter tuning was greatly conserved. Although the objective data used in the case study was NIR spectra data, the application area of the proposed method is not limited to a specific type of data. The proposed NCVW is applicable to general soft-sensor design when the number of input variables is large. Therefore, NCVW will contribute to realizing the efficient soft-sensor design.

AUTHOR CONTRIBUTIONS

KF developed the proposed method, analyzed the data, and wrote the initial draft of the manuscript. MK contributed to data collection and analysis and assisted in the preparation of the manuscript. Both authors approved the final version of the manuscript, and agree to be accountable for all aspects of the work.

FUNDING

This work was partially supported by the JFE 21st Century Foundation.

ACKNOWLEDGMENTS

The authors thank Daiichi-Sankyo Co., Ltd. for providing real operation data used in case studies.

REFERENCES

Ahmad, I., Kano, M., Hasebe, S., Kitada, H., and Murata N. (2014). Gray-box modeling for prediction and control of molten steel temperature in tundish. *J. Process Control* 24, 375–382. doi: 10.1016/j.jprocont.2014.01.018

Andersen, C. M., and Bro, R. (2010). Variable selection in regression – a tutorial. *J. Chemometrics* 24, 728–737. doi: 10.1002/cem.1360

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

Ding, C. H. Q., He, X., Zha, H., Gu, M., and Simon, H. D. (2001). "A min-max cut algorithm for graph partitioning and data clustering," in *IEEE International Conference on Data Mining (ICDM)* (San Jose, CA) 107–114.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Ann. Stat.* 32, 407–499. doi: 10.1214/009053604000000067

Fujiwara, K., and Kano, M. (2015). Efficient input variable selection for soft-senor design based on nearest correlation spectral clustering and group lasso. *ISA Trans.* 58, 367–379. doi: 10.1016/j.isatra.2015.04.007

Fujiwara, K., Kano, M., and Hasebe, S. (2009). Soft-sensor development using correlation-based just-in-time modeling. *AIChE J.* 55, 1754–1765. doi: 10.1002/aic.11791

Fujiwara, K., Kano, M., and Hasebe, S. (2010). Development of correlation-based clustering method and its application to software sensing. *Chemom. Intell. Lab. Syst.* 101, 130–138. doi: 10.1016/j.chemolab.2010.02.006

Fujiwara, K., Kano, M., and Hasebe, S. (2012a). Development of correlation-based pattern recognition algorithm and adaptive soft-sensor design. *Control Eng. Pract.* 20, 371–378. doi: 10.1016/j.conengprac.2010.11.013

Fujiwara, K., Kano, M., and S.Hasebe (2011). Correlation-based spectral clustering for flexible process monitoring. *J. Process Control* 21, 1348–1448. doi: 10.1016/j.jprocont.2011.06.023

Fujiwara, K., Sawada, H., and Kano, M. (2012b). Input variable selection for pls modeling using nearest correlation spectral clustering. *Chemom. Intell. Lab. Syst.* 118, 109–119. doi: 10.1016/j.chemolab.2012.08.007

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

Kano, M., and Fujiwara, K. (2013). Virtual sensing technology in process industries: trends and challenges revealed by recent industrial applications. *J. Chem. Eng. Jpn.* 46, 1–17. doi: 10.1252/jcej.12we167

Kano, M., and Ogawa, M. (2010). The state of the art in chemical process control in japan: Good practice and questionnaire survey. *J. Process Control* 20, 969–982. doi: 10.1016/j.jprocont.2010.06.013

Kim, S., Kano, M., Nakagawa, H., and Hasebe, S. (2014). Input variable scaling for statistical modeling. *Comput. Chem. Eng.* 74, 59–65. doi: 10.1016/j.compchemeng.2014.12.016

Kim, S., Kano, M., Nakagawa, H., and Hasebe, S. (2011). Estimation of active pharmaceutical ingredients content using locally weighted partial least squares and statistical wavelength selection. *Int. J. Pharm.* 421, 269–274. doi: 10.1016/j.ijpharm.2011.10.007

Kubinyi, H. (1993). *3D QSAR in Drug Design; Theory, Methods, and Applications*. Leiden; Holland: ESCOM.

Mehmood, T., Liland, K. H., Snipen, L., and Sæbø, S. (2012). A review of variable selection methods in partial least squares regression. *Chemom. Intell. Lab. Syst.* 118, 62–69. doi: 10.1016/j.chemolab.2012.07.010

Miyano, T., Kano, M., Tanabe, H., Nakagawa, H., Watanabe, T., and Minami, H. (2014). 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, 504–513. doi: 10.1016/j.ijpharm.2014.09.007

Ng, A. N., Jordan, M. I., and Weiss, Y. (2002). On "spectral clustering: Analysis and an algorithm," in *NIPS* (Vancouver, BC), 849–856.

Roggo, Y., Chalus, P., Maurer, L., Lema-Martinez, C., Edmond, A., and Jent, N. (2007). A review of near infrared spectroscopy and chemometrics in pharmaceutical technologies. *J. Pharm. Biomed. Anal.* 44, 683–700. doi: 10.1016/j.jpba.2007.03.023

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

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

Wold, S., Sjostroma, M., and Erikssonb, L. (2001). Pls-regression: a basic tool of chemometrics. *Chemom. Intell. Lab. Syst.* 58, 109–130. doi: 10.1016/S0169-7439(01)00155-1

Yuan, M., and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 68, 49–67. doi: 10.1111/j.1467-9868.2005.00532.x

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.