Chemometrics and Intelligent Laboratory Systems 138 (2014) 72–83

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab

Partial least squares–slice transform hybrid model for nonlinear calibration

Peng Shan [a,*], Silong Peng [a], Yiming Bi [a], Liang Tang [a], Chixiao Yang [a], Qiong Xie [a], Changwen Li [b]

[a] Institute of Automation, Chinese Academy of Sciences, 100190 Beijing, China
[b] Food Research Institute, Tasly Group, 300410 Tianjin, China

ARTICLE INFO

Article history:
Received 23 May 2014
Received in revised form 7 July 2014
Accepted 20 July 2014
Available online 27 July 2014

Keywords:
Partial least squares
Slice transform
Piecewise linear mapping function
Cross-validation

ABSTRACT

A hybrid model (herein referred to as PLS-SLT) founded on partial least squares (PLS) and slice transform (SLT) is proposed to model nonlinear chemical systems with a wide range of response variable. In the modeling process, PLS predicted values of calibration set were taken as inputs for the subsequent SLT to further approximate to observed values by a least square criterion. The estimated optimal piecewise linear mapping function was then applied to test set to give the final prediction result. Theoretically, PLS-SLT can be proven to be equivalent to the PLS-based piecewise linear model in the **y**-space. PLS-SLT is compared with PLS and other calibration models on two spectral datasets. The Wilcoxon signed rank test is used to statistically compare predictive performance of two competing calibration models. Experimental results show that the performance of PLS-SLT is at least statistically not worse than PLS and other models.

1. Introduction

Multivariate calibration models are widely used to find the relationship between **X** (predictor variables/spectral data) and **y** (response variables/properties) in chemometrics. Among various models, the partial least squares (PLS) model has obtained consistent success due to its powerful ability to deal with multicollinearity with over-determined linear systems [1–4]. PLS is somewhat capable of handling nonlinearities by including additional latent variables, but at the risk of overfitting and at the cost of being an unnecessarily complex calibration model. Hence, when facing complex systems with significant nonlinear characteristics, the conventional PLS model is not appropriate for describing the underlying data structure [5,6].

To delineate this issue, an increasing number of PLS-based regression models have been proposed [7,8]. The first type is nonlinear PLS models that describe the relationship between latent variables in a nonlinear way. Original linear inner relation is substituted with some nonlinear functions such as quadratic polynomial, spline and artificial neural networks. As nonlinear variants of PLS, quadratic PLS (QPLS) [9–11], spline PLS (SPLS) [12] and neural network PLS (NNPLS) [13,14] have gradually been proposed. However, within these models there exist some inherent obstacles. For example, the nonlinearity of the QPLS model is limited because of the predefined form of the quadratic function. Conversely, SPLS and NNPLS are flexible enough to fit varying nonlinearity, but these two algorithms suffer from over-fitting or local minima. The second type is locally weighted regression PLS (LWR-PLS) [15–17], which uses PLS as a regression method to construct a local model by prioritizing samples in a dataset according to the similarity between them and a query sample. The defined drawbacks are the use of more than one model to cover all the test samples and the increased number of samples needed for the modeling.

The third type is kernel PLS (KPLS) [18–22]. Unlike the aforementioned nonlinear PLS methods, KPLS transforms original predictor variables into a high-dimensional feature spaces via nonlinear kernel functions and then establish a linear PLS model in the new feature space. Its advantage lies in the fact that nonlinear optimization problem is avoided by utilizing the kernel function corresponding to the inner product in the feature space. Nevertheless, without sufficient prior information and knowledge of the complex nonlinear relation in the data, selecting an optimal kernel function is a still trial and error process and depends largely on the experience or expertise of the practitioner. In general, the user-defined kernel functions are not sufficient to capture the variability of the nonlinear transformation and can lead to under-fitting of the model.

In addition to the above-mentioned nonlinear models, another PLS based model is to utilize some suitable transform on PLS output to handle the nonlinearity that remains in the output, such as polynomial transformation [23,24]. The corresponding model is abbreviated as PLS-Poly. Nevertheless, polynomial isn't flexible enough and another better transformation is desired. More recently, one of the authors (Y. Bi) presented a modified PLS method with slice transform-based weight updating strategy [25]. In his method, the substitution property of slice transform (SLT) was used to obtain the optimal piecewise linear

* Corresponding author. Tel.: +86 1062520293.
E-mail address: peng.shan@ia.ac.cn (P. Shan).

http://dx.doi.org/10.1016/j.chemolab.2014.07.015
0169-7439/

representation of the weight vectors. Hence, inspired by the excellent performance of SLT technique in piecewise linear approximation, we proposed a novel PLS based piecewise linear model termed as PLS-SLT. The fundamental idea is to use SLT to provide a piecewise linear representation of PLS predicted values and estimate the optimal mapping function (under least squares criterion) representing the nonlinear relationship between PLS predicted values and observed values. It is equivalent to complete two tasks. One is to automatically split the overall range of a noted response variable into some appropriate subranges; the other is to provide each subrange with more accurate linear predictor–response relationship. In a linear manner, PLS-SLT achieves nonlinear calibration. PLS, QPLS, LW-PLS, KPLS, SLT-PLS and PLS-Poly are selected to compare with PLS-SLT on prediction accuracy. In addition, the Wilcoxon signed rank test is used to determine whether PLS-SLT statistically significantly outperformed other models. Two publicly available near infrared (NIR) datasets are used as experimental objects. Results of these two datasets reveal that PLS-SLT has better predictive ability than other models and has a potential application in nonlinear calibration for chemical systems with a wide range of response variable.

2. Theory

2.1. Summary of PLS and SLT

In the following, PLS refers to PLS1 if there is no special annotation. Let us denote by $\mathbf{X} \in \mathbb{R}^{n \times N}$ the spectral data (predictor variables) matrix and $\mathbf{y} \in \mathbb{R}^{n \times 1}$ the concentration (response variable) vector. The superscripts n and N represent the number of samples and wavenumbers, respectively. Both $\mathbf{X}$ and $\mathbf{y}$ are assumed to be column mean-centered.

2.1.1. The PLS model

The core idea of linear PLS is to project two blocks of variables ($\mathbf{X}$ and $\mathbf{y}$) onto their corresponding subspace of orthogonal latent variables (scores) and then model the linear relationship between them. The nonlinear iterative partial least squares (NIPALS) [26] algorithm is commonly used to sequentially extract the weight vectors $\mathbf{w}$ and $\mathbf{c}$ by maximizing the covariance between the latent vectors ($\mathbf{t}$ and $\mathbf{u}$). In general, the PLS model is composed of two linear latent variable decompositions of the input and output variables and a linear inner relation between each pair of latent variables [27,28]. The corresponding formulas are demonstrated by the following:

$$\mathbf{X} = \sum_{i=1}^{A} \mathbf{t}_i \mathbf{p}_i^{\mathrm{T}} + \mathbf{E}_{\mathrm{X}} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E}_{\mathrm{X}} \tag{1}$$

$$\mathbf{y} = \sum_{i=1}^{A} \mathbf{u}_i \mathbf{q}_i^{\mathrm{T}} + \mathbf{E}_{\mathrm{y}} = \mathbf{U}\mathbf{Q}^{\mathrm{T}} + \mathbf{E}_{\mathrm{y}} \tag{2}$$

$$\mathbf{U} = \mathbf{T}\mathbf{B} + \mathbf{E}_{\mathrm{U}} = [b_1\mathbf{t}_1, \cdots, b_A\mathbf{t}_A] + \mathbf{E}_{\mathrm{U}} \tag{3}$$

where $\mathbf{T} \in \mathbb{R}^{n \times A}$ and $\mathbf{P} \in \mathbb{R}^{N \times A}$ are score and loading matrices for $\mathbf{X}$-block; $\mathbf{U} \in \mathbb{R}^{n \times A}$ and $\mathbf{Q} \in \mathbb{R}^{1 \times A}$ are the score and loading matrices for $\mathbf{y}$-block; $\mathbf{E}_{\mathrm{X}} \in \mathbb{R}^{n \times N}$, $\mathbf{E}_{\mathrm{y}} \in \mathbb{R}^{n \times 1}$ and $\mathbf{E}_{\mathrm{U}} \in \mathbb{R}^{n \times 1}$ are the corresponding residual error matrices. Note that the inner regression coefficients ($b_i = \mathbf{u}_i^{\mathrm{T}}\mathbf{t}_i / (\mathbf{t}_i^{\mathrm{T}}\mathbf{t}_i)$, $i = 1, 2, \ldots, A$) are represented by a diagonal matrix $\mathbf{B}$ with the off-diagonal elements set equal to zero.

By introducing the matrix $\mathbf{R} = \mathbf{W}(\mathbf{P}^{\mathrm{T}}\mathbf{W})^{-1}$, the input score matrix ($\mathbf{T}$) can be directly evaluated from the original predictor matrix, $\mathbf{X}$, as follows:

$$\mathbf{T} = \mathbf{X}\mathbf{R} = \mathbf{X}\mathbf{W}\left(\mathbf{P}^{\mathrm{T}}\mathbf{W}\right)^{-1} \tag{4}$$

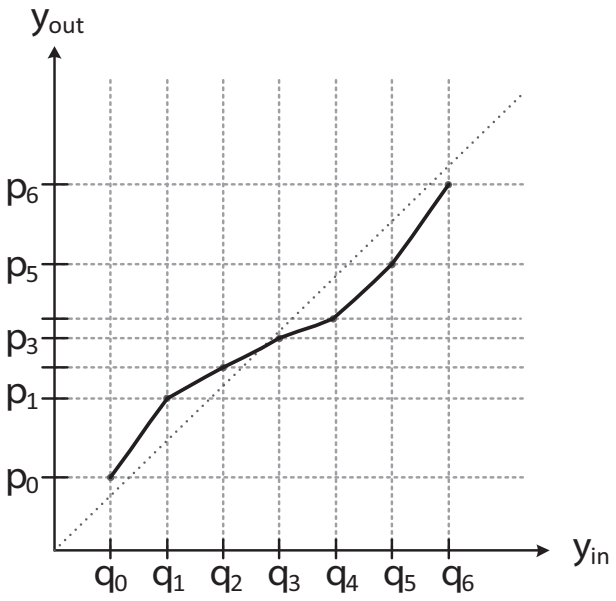

Fig. 1. Piecewise linear map with the substitution property of the slice transform.

where $\mathbf{W} \in \mathbb{R}^{N \times A}$ is the weight matrix (projection matrix) for $\mathbf{X}$-block. Thus, the final regression model given by the PLS method can be written as:

$$\hat{\mathbf{y}} = \mathbf{U}\mathbf{Q}^{\mathrm{T}} = \mathbf{T}\mathbf{B}\mathbf{Q}^{\mathrm{T}} = \mathbf{X}\mathbf{W}\left(\mathbf{P}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{B}\mathbf{Q}^{\mathrm{T}}. \tag{5}$$

2.1.2. Overview of SLT

The basic concept of SLT is to implement a linear representation of a signal with linear splines as basis functions [25,29]. For a vector $\mathbf{y} \in \mathbb{R}^{n \times 1}$ with each element satisfying $y_i \in [a, b)$, the SLT of $\mathbf{y}$ can be defined as follows:

$$\mathbf{y} = \mathbf{S}_{\mathbf{q}}(\mathbf{y})\mathbf{q} \tag{6}$$

where $\mathbf{S}_{\mathbf{q}}(\mathbf{y}) \in \mathbb{R}^{n \times (m+1)}$ and $\mathbf{q} \in \mathbb{R}^{m+1}$ are slice transform matrix and original boundary vector respectively. Original boundary vector $\mathbf{q} = [q_1, q_2, \ldots, q_{m+1}]^{\mathrm{T}}$ consists of $m + 1$ boundaries which is formed by dividing the interval $[a, b)$ into m bins arbitrarily and the elements satisfy the following property: $q_1 = a < q_2 < q_3 \ldots < q_{m+1} = b$.

Table 1
Summary of PLS-SLT.

Given calibration set ($\mathbf{X}_{calib}$, $\mathbf{y}_{calib}$) and test set ($\mathbf{X}_{test}$, $\mathbf{y}_{test}$)
1. Mean center $\mathbf{X}_{\mathrm{calib}}$, $\mathbf{y}_{\mathrm{calib}}$ and $\mathbf{X}_{\mathrm{test}}$
2. PLS modeling
   (a) Computer PLS regression coefficients $\mathbf{b}_{\mathrm{pls}}$ by Eq. (12)
   (b) Computer predicted values $\hat{\mathbf{y}}_{\mathrm{calib}}^{\mathrm{pls}}$ and $\hat{\mathbf{y}}_{\mathrm{test}}^{\mathrm{pls}}$ for calibration and test sets by Eq. (12)
3. SLT modeling
   (a) Form new calibration set ($\hat{\mathbf{y}}_{\mathrm{calib}}^{\mathrm{pls}}$, $\mathbf{y}_{\mathrm{calib}}$) and test set ($\hat{\mathbf{y}}_{\mathrm{test}}^{\mathrm{pls}}$, $\mathbf{y}_{\mathrm{test}}$)
   (b) Transform according to Eq. (13) ($\hat{\mathbf{y}}_{\mathrm{calib}}^{\mathrm{pls}} \rightarrow \tilde{\mathbf{y}}_{\mathrm{calib}}^{\mathrm{pls}}$, $\hat{\mathbf{y}}_{\mathrm{test}}^{\mathrm{pls}} \rightarrow \tilde{\mathbf{y}}_{\mathrm{test}}^{\mathrm{pls}}$ and $\mathbf{y}_{\mathrm{calib}} \rightarrow \tilde{\mathbf{y}}_{\mathrm{calib}}$)
   (c) Construct slice transform matrices $\mathbf{S}_{\mathbf{q}}\left(\tilde{\mathbf{y}}_{\mathrm{calib}}^{\mathrm{pls}}\right)$ and $\mathbf{S}_{\mathbf{q}}\left(\tilde{\mathbf{y}}_{\mathrm{test}}^{\mathrm{pls}}\right)$ by Eq. (9)
   (d) Calculate new boundary vector $\mathbf{p}$ with 5-fold cross-validation by Eqs. (17)–(18)
   (e) Calculate final predicted values $\hat{\mathbf{y}}_{\mathrm{calib}}$ and $\hat{\mathbf{y}}_{\mathrm{test}}$ by Eqs. (19)–(20)