Contents lists available at ScienceDirect

Chemical Engineering Research and Design

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

Multivariate data modeling using modified kernel partial least squares

Gao Yingbin, Kong Xiangyu, Hu Changhua, Zhang Zhengxin, Li Hongzeng, Hou Li'an*

The Xi'an Research Institute of High Technology, Xi'an, Shaanxi 710025, PR China

ABSTRACT

There are two problems, which should be paid attention to when using kernel partial least squares (KPLS), one is overfitting and another is how to eliminate the useless information mixed in the independent variables **X**. In this paper, the stochastic gradient boosting (SGB) method is adopted to solve the overfitting problems and a new method called kernel net analyte preprocessing (KNAP) is proposed to remove undesirable systematic variation in **X** that is unrelated to **Y**. Thus, by combining the two methods, a final modeling approach named modified KPLS (MKPLS) is proposed. Two simulation experiments are carried out to evaluate the performance of the MKPLS method. The simulation results show that MKPLS method can not only be resistant to overfitting but also improve the prediction accuracy.

Keywords: Kernel partial least squares; Stochastic gradient boosting; Kernel net analyte preprocessing; Overfitting; Multivariate data modeling; Data preprocessing

1. Introduction

Partial least squares (PLS) regression, which was proposed by Wold in 1983, is one of the most commonly used calibration methods in chemometrics (Wold et al., 1983). PLS regression searches for a set of components (called latent variables) that performs a simultaneous decomposition of independent variables (**X**) and dependent variables (**Y**) with the constraint that these components explain as much as possible of the covariance between **X** and **Y** (Abdi, 2010). PLS is a powerful technique for process modeling and calibration in systems where the predictor variables are collinear, measurement data contain noise, variables have high dimensionality, and where there are fewer observations than predictor variables (Zhang et al., 2010a). But PLS regression is a linear method and is inappropriate for describing the underlying data structure because such systems may exhibit significant nonlinear characteristics (Zhang and Zhang, 2009). To solve this issue, a nonlinear PLS method, called kernel partial least squares (KPLS), was proposed by Rosipal and Trejo (Rosipal and Trejo, 2002). The original datasets are nonlinearly transformed into a feature space of arbitrary dimensionality via nonlinear mapping, and then a linear model is created in the feature space (Zhang et al., 2012; Zhang and Hu, 2011). Because it's easy to understand and operate, KPLS has been widely used in many fields, such as pattern recognition (Qu et al., 2010), signal processing (Helander et al., 2012), fault diagnosis (Zhang et al., 2010b), and so on.

Data preprocessing methods can reduce the noise effect on the data, extract more useful information for model building, and improve the prediction ability and model robustness. Many data preprocessing methods have been proposed in recent years, such as multiplicative scatter correction (MSC) (Thennadil et al., 2006), standard normal variate (SNV) (Barnes et al., 1989), Savitzky–Golay smoothing and differentiation (Savitzky and Golay, 1964), and so on. Recent work has focused on one method called net analyte preprocessing (NAP), which was firstly proposed by Lorber (Lorber, 1986). Lorber proved that the useless information in **X**, which is not related to the dependent variables **Y** for model building, can be completely removed by NAP and the prediction accuracy can also

* *Corresponding author*. Tel.: +86 02984744954.
E-mail address: xiangyukong01@163.com (X. Kong).
Received 16 May 2014; Received in revised form 1 September 2014; Accepted 3 September 2014
Available online 15 September 2014
http://dx.doi.org/10.1016/j.cherd.2014.09.004

be improved. However, this method can be effectively performed only on a set of observations that vary linearly. When the variations are nonlinear data, the linear NAP is inappropriate for fitting the nonlinear data. This situation will affect the performance of NAP about removing systematic variation from an input set $\boldsymbol{X}$ not correlated to the response set $\boldsymbol{Y}$ (Zhang et al., 2010a). For this reason, inspired by the kernel function methods, we propose a new method called kernel net analyte preprocessing (KNAP) in this paper, which solves the issue of data nonlinearity compared to NAP.

Another problem, which has to face with when using PLS or KPLS, is how to avoid overfitting. Overfitting is a commonly observed situation where the learner performs well on training data, but has a large error on the test data (Hawkins, 2004). In (Zhang et al., 2004), Massart proposed a weighted averaged PLS (APLS) method, which has compared predication ability and is also relatively robust to overfitting. Recently, a new method called boosting has drawn much attention. In (Schapire, 1990), based on the so-called margin theory, Schapire proved that boosting was more robust to overfitting. Combining the boosting method with PLS, Massart proposed a new method called boosting partial least squares (BPLS) to solve the overfitting problem (Zhang et al., 2005). BPLS has been used in many fields, such as quantitative structure-activity/property relationship (QSAR/QSPR) study (Zhou et al., 2007), near infrared spectroscopy (Tan et al., 2010), mass spectrometry analysis (He et al., 2004), and so on. Although so many applications of boosting method in PLS regression modeling, there are little attention paid to the KPLS regression modeling. Since many applications have demonstrated the superiority of KPLS over PLS in solving nonlinear problems (Zhang and Hu, 2011; Qu et al., 2010; Helander et al., 2012; Zhang et al., 2010b), it is necessary to study the performance of boosting in the KPLS modeling. In (Friedman, 2002), a variant of boosting method called stochastic gradient boosting (SGB) was proposed, and this method has less computation time and higher prediction accuracy than the boosting method. Combing the SGB with KPLS, we proposed a new method called stochastic gradient boosting-kernel partial least squares (SGB-KPLS) in this paper, which aims to solve the overfitting problem when using KPLS method. In this paper, the KNAP method is introduced into SGB-KPLS modeling procedure and a final method called KNAP-SGB-KPLS (modified kernel partial least squares (MKPLS) for short) is proposed.

The rest of the paper is organized as follows. In Section 2, the basic theories and algorithms of KPLS and SGB are introduced, and then detail descriptions of the proposed methods (KNAP and MKPLS) are given. Computer simulations are carried out in Section 3. Finally, our conclusions are drawn in Section 4.

2. Modified kernel partial least squares

2.1. Notations

In order to conveniently understand the below mentioned symbols, some essential notations are illuminated in this section. Throughout the present work, matrices will be noted in capital bold (as in $\boldsymbol{X}$), column vectors in small bold (as in $\boldsymbol{x}$), and scalar variables in italicized characters (as in n). Some notational symbols are listed below:

$\boldsymbol{F}$ feature space
$\Phi(\boldsymbol{X})$ data matrix in feature space $\boldsymbol{F}$
$\boldsymbol{K}$ kernel matrix $\boldsymbol{K} = \Phi(\boldsymbol{X})\Phi^T(\boldsymbol{X})$
$\boldsymbol{K}^*$ net analyte kernel matrix
$\boldsymbol{X}$ independent variables matrix
$\boldsymbol{Y}$ dependent variables matrix
$\boldsymbol{I}$ identity matrix
$\boldsymbol{b}$ regression coefficient vector
h number of latent variables
m number of basis regression models when using SGB method
n number of samples in data set
n_c size of subsample used in the SGB method
υ shrinkage value

2.2. Kernel partial least squares

KPLS is an extension of PLS in the nonlinear feature space. According to Cover's theorem, the nonlinear structure in the feature space is more likely to be linear after a high-dimensional nonlinear mapping (Rosipal, 2003). This higher dimensional linear space is referred to as the feature space $\boldsymbol{F}$ (Zhang et al., 2010b). First, consider a nonlinear transformation of the input data $\boldsymbol{x}_i$, $i = 1, 2, \ldots, n$ into feature space $\boldsymbol{F}$.

$$\Phi : \boldsymbol{x}_i \in R^n \rightarrow \Phi(\boldsymbol{x}_i) \in \boldsymbol{F} \tag{1}$$

where it is assumed that $\sum_{i=1}^{n} \Phi(\boldsymbol{x}_i) = 0$, i.e. mean centering in the high-dimensional space should be performed before applying KPLS. $\Phi(\boldsymbol{x}_i)$ is a nonlinear mapping function that projects the input vectors from the original space to $\boldsymbol{F}$. Note that the dimensionality of the feature space $\boldsymbol{F}$ is arbitrarily large, and can even be infinite. Denote $\Phi(\boldsymbol{X})$ as the $(n \times s)$ matrix whose ith row is the vector $\Phi(\boldsymbol{x}_i)$ in the s-dimensional feature space $\boldsymbol{F}$. By means of the introduction of the kernel trick $\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i)\Phi^T(\boldsymbol{x}_j)$, one can avoid both performing explicit nonlinear mappings and computing dot products in the feature space (Cao et al., 2011). The commonly used kernel functions are the polynomial kernel $\boldsymbol{K}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \left\langle \boldsymbol{x}_i, \boldsymbol{x}_j \right\rangle^r$, radial basis kernel $\boldsymbol{K}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp\left(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2 / c\right)$, and sigmoidal kernel $\boldsymbol{K}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \tanh(\beta_0 \left\langle \boldsymbol{x}_i, \boldsymbol{x}_j \right\rangle + \beta_1)$, where c, r, β_0, β_1 are the parameters of the kernels and should be predefined by users. The steps of KPLS method are as follows:

For $i = 1, 2, \ldots, h$ (h is the number of latent variables), repeat the following steps:

Step 1: Initialize, set $\boldsymbol{K}_i = \boldsymbol{K}$, $\boldsymbol{Y}_i = \boldsymbol{Y}$, set $\boldsymbol{u}_i$ equal to any column of $\boldsymbol{Y}_i$.

Step 2: Compute the score vector of $\Phi(\boldsymbol{X}) : \boldsymbol{t}_i = \boldsymbol{K}\boldsymbol{u}_i / \sqrt{\boldsymbol{u}_i^T \boldsymbol{K}_i \boldsymbol{u}_i}$.

Step 3: Compute the loading vector of $\boldsymbol{Y}_i : \boldsymbol{q}_i = \boldsymbol{Y}_i \boldsymbol{t}_i / \left\|\boldsymbol{t}_i^T \boldsymbol{t}_i\right\|$.

Step 4: Compute the score vector of $\boldsymbol{Y}_i : \boldsymbol{u}_i = \boldsymbol{Y}_i \boldsymbol{q}_i / \boldsymbol{q}_i^T \boldsymbol{q}_i$.

Step 5: If $\boldsymbol{u}_i$ converges, then go to step 6; else return to step 2.

Step 6: Deflation $\begin{cases} \boldsymbol{K}_{i+1} = (\boldsymbol{I} - \boldsymbol{t}_i \boldsymbol{t}_i^T / \boldsymbol{t}_i^T \boldsymbol{t}_i)\boldsymbol{K}_i(\boldsymbol{I} - \boldsymbol{t}_i \boldsymbol{t}_i^T / \boldsymbol{t}_i^T \boldsymbol{t}_i) \\ \boldsymbol{Y}_{i+1} = (\boldsymbol{I} - \boldsymbol{t}_i \boldsymbol{t}_i^T / \boldsymbol{t}_i^T \boldsymbol{t}_i)\boldsymbol{Y}_i \end{cases}$, $i = i + 1$, and go to step 2.

After all the h latent variables are extracted, the regression coefficient $\boldsymbol{b}$ in KPLS can be obtained from

$$\boldsymbol{b} = \Phi^T \boldsymbol{U}(\boldsymbol{T}^T \boldsymbol{K} \boldsymbol{U})^{-1} \boldsymbol{T}^T \boldsymbol{Y} \tag{2}$$

where $\boldsymbol{T} = [\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_h]$ and $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h]$ are the score matrix. As a result, when the number of test data is n_t, the