QUANTITATIVE ANALYSIS OF SURFACE ENHANCED RAMAN SPECTRA

by

SHUO LI

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2013

To my mother and my father who set the example and who made me who I am.

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Jean Gao for constantly motivating and encouraging me, and also for her invaluable advice during the course of my doctoral studies. I wish to thank my academic advisors Dr. Chris Ding, Dr. Farhad A. Kamangar and Dr. Vassilis Athitsos for their interest in my research and for taking time to serve in my dissertation committee.

I would also like to extend my appreciation to our collaborators, Dr. Digant P. Dave and James O. Nyagilo, who provide us the experimental data and explanations of the background knowledge.

Finally, I would like to express my deep gratitude to my families who have encouraged and inspired me and sponsored my undergraduate and graduate studies.

July 19, 2013

ABSTRACT

QUANTITATIVE ANALYSIS OF SURFACE ENHANCED RAMAN SPECTRA

SHUO LI, Ph.D.

The University of Texas at Arlington, 2013

Supervising Professor: Jean Gao

Quantitative analysis of Raman spectra using surface-enhanced Raman scattering (SERS) nanoparticles has shown the potential and promising trend of development in vivo molecular imaging. One of the key job is from the intensities of Raman signals to predict the quantities of analytes. Direct classical least squares (DCLS) and multivariate calibration (MC) are commonly used models. DCLS relies on source Raman signals as the references. But the inherent Instability of Raman signals make the DCLS model biased. MC model relies on a batch of training mixture Raman signals together with the ground truth mixing concentrations to build the multivariate multiple linear regression models, so as to reduce the bias from the instability of source Raman signals. But it also brings in the more variables than observations problem. Latent variable regression (LVR) model avoids that problem by extracting low dimensional latent variables (LVs) (or extracted features) to do regression with concentrations. Among several LVR methods, partial least squares regression (PLSR) algorithms are more robust, since their LVs both represent original Raman signals and predict concentrations. In this thesis, quantitative analysis models and method-

s are compared to show why PLSR algorithms are more robust for the purpose of quantitative analysis of Raman spectra.

Only PLSR cannot handle the instable background of Raman signals. Baseline correction methods are commonly used as the preprocessing to find a slowly changed baseline under the signal as the estimated background. Raman peaks are extracted then by subtracting the baseline from the Raman signal. But baseline correction methods are usually time consuming iterative processes, and normally they cannot deal with the multi-scale property of Raman peaks. We designed a simple algorithm, called continuous wavelet transform (CWT) based partial least squares regression (CWT-PLSR) that uses the average CWT coefficients of mixture Raman signals to do PLSR with mixing concentrations. It extracts the multi-scale information of Raman peaks and so is more robust than traditional baseline correction methods.

PLSR balances two purposes, representing Raman signals and predicting concentrations, in the objective function. But the proportion of each purpose is fixed in the objective function of PLSR. To improve the flexibility of PLSR, we designed a new continuum regression (CR) method that use a tuning parameter to control the proportion of each purpose in the objective function and it gives more reasonable weights to Raman peaks. It beats other two CR methods by embracing PCR, RRR and PLS as three special cases, and is simply achieved by NIPALS algorithm.

Tuning parameters of PLSR and CR methods are normally decided by time-consuming cross-validation methods. And some parameters have infinite numbers of possible values in continuous ranges. There is no way to test every value by cross-validation methods. Nonparametric Bayesian models of these methods are needed to decide the parameters automatically from the training data. As a foundation work, we design a probabilistic PLS regression model to give a probabilistic view of the

PLSR methods. Future Bayesian models can be achieved by adding reasonable priors of the parameters.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction to Quantitative Analysis of Raman Spectra

1.1.1 Raman Spectra

Raman scattering or Raman effect is the physical phenomenon when the monochromatic laser light interacts with molecular vibrations or other excitations, resulting in the energy of the laser photons being shifted upwards or downwards. The shifts in energy are referred as Raman frequencies or Raman shifts. A characteristic range of Raman shifts, which give the unique spectral information of a particular molecule, are collectively referred to as the Raman spectrum [1,2]. Keren *et al.* [3] and Zavaleta *et al.* [4] reported three properties of Raman spectrum: (a) Source spectra do not change when the pure Nano-Tags are mixed; (b) The mixture spectrum equals to the summation of the source spectra; (c) Within certain range of concentrations, the intensities of source spectra are approximately linearly related to the concentrations of pure Nano-Tags. With these properties, Raman spectroscopy technique can be used to study vibrational, rotational, and other low-frequency modes in a system relying on Raman scattering.

But the inherent weak magnitude of Raman scattering limits the sensitivity and as a result, the biomedical applications of Raman spectroscopy. The development of the surface-enhanced Raman spectroscopy or scattering (SERS) offers an exciting opportunity to overcome this serious signal to noise problems inherent in Raman spectroscopy.

1.1.2 Surface-enhanced Raman Spectroscopy (SERS)

The SERS-nanoparticles, normally silver or gold colloids or substrate containing silver or gold, are designed to enhance the intensities of Raman spectra. When surface plasmons of silver or gold are excited by the laser, they result in an increase in the electric fields surrounding the metal. Given that Raman intensities are proportional to the electric field, there is a large increase in the measured signal [2].

With such large enhancement, SERS has been regarded as one of the most sensitive techniques that can provide the spectral fingerprint of every chemical compound and has been a routine method used as an analytical tool in food industry, pharmaceutical, chemical and biological community [5] to investigate the composition of materials. It has been applied by Cheung *et al.* [6] to quantify the banned food dye, by Lai *et al.* [7] to analyze sulfa drugs, by Strickland and Batt [8] to detect carbendazim, by Rainer *et al.* [9] to determine the amount of creatinine in human serum, and by authors in [10–12] to detect DNA sequence. It also has been studied in the field of biomedical diagnostics, especially in the research of cancer detection. Antibody conjugated nanoparticles, which can be attached to specific proteins in cancer cells, are injected into body. Cancer can be detected by imaging large amount of such nanoparticles gathered in certain place inside body by Raman imaging techniques. Kim *et al.* [13] used the antibody-conjugated SERS dots to target the surface receptor HER2 and CD10 of breast cancer cells (MCF-7) and floating leukemia cells (SP2/O) in living cells. Keren *et al.* [3] demonstrated the ability of the modified Raman microscope to detect single-walled carbon nanotubes (SWNTS) conjugated with arginine-glycine-aspartate (RGD) peptide fractions in an integrin positive U87MG tumor model in living mice. These RGD peptide fractions bind to $\alpha_v\beta_3$ integrin, which is overexpressed in angiogenic vessels and various tumor cells. Zavaleta *et al.* [4] demonstrated the picomolar sensitivity and multiplexing capabili-

ties of SERS nanoparticles and showed SERS to be a potential noninvasive preclinical imaging technique. Kennedy *et al.* [14] also developed nanoparticle probes for SERS imaging of cell surface receptor proteins.

1.1.3 Quantitative Analysis of Raman Spectra

In order to estimate the amount of the receptor proteins and so the amount of cancer cells, the so called quantitative analysis of surface-enhanced Raman spectrum, which is from the Raman spectrum of the mixed Nano-Tags to determine the mixing concentration of each pure Nano-Tag, is the key job. A simple way is based on the direct classical least squares (DCLS) model, which is used in literatures [3,4,15]. The preparation of training data is easy, but the model has an unavoidable biased problem. The more commonly used methods are multivariate approaches such as principal component regression (PCR) [6,7,16–18] and partial least squares regression (PLSR) algorithms [6–9, 16–22], which are essentially based on the multivariate calibration model that can reduce that bias.

1.1.4 Instable Background and Preprocessing

In reality a Raman signal obtained from the spectroscopy is composed of the Raman spectrum together with an instable background and some generated noises which makes the Raman signal irreproducible (shown in Fig. 1.1). This inherent instable background is mainly because of the emission of fluorescence [23]. Besides, some instrumental factors, like variations in laser power or wavelength, optical train variations or irreproducible sample placement, and the change of position and angel of Ag or Au sol attached on analyte molecules during time [21], will also give instable signals. In order to reduce the effects of backgrounds and noises on the quantitative analysis, baseline correction methods [24] are usually used as the preprocessing to

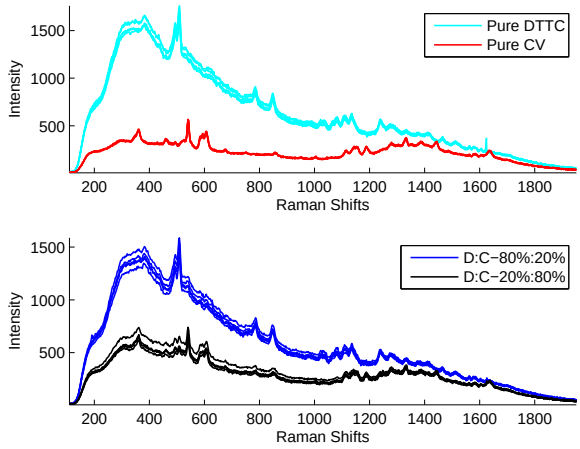


Figure 1.1: Instable Raman signals of two samples of mixed Nano-Tags. Signals with the same color are five duplicate Raman signals of one sample obtained at different time.

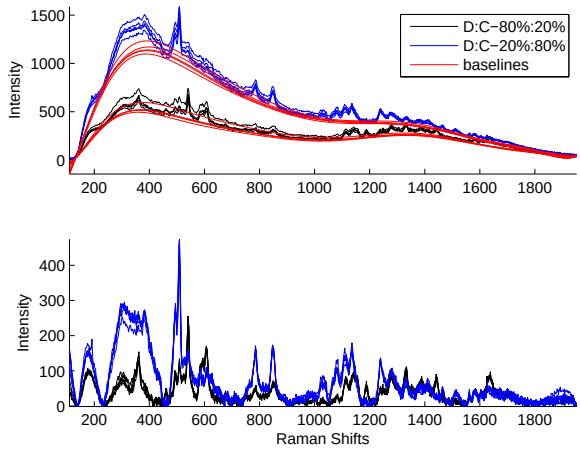


Figure 1.2: Baselines of Raman signals and results of baseline correction.

extract the Raman spectra from Raman signals. Fig. 1.2 shows the results of the baseline correction.

1.2 Motivations and Contributions

As introduced in section 1.1.3, several methods are currently used in literatures for quantitative analysis of SERS. In Chapter 2 we analyze the mathematical definitions and essential meanings of those models and methods, explain the suitable situation for each method and illustrate PLSR is more reasonable when doing the quantitative analysis of SERS. Also, since there are several variants and algorithms of PLSR methods, we analyze the differences between variants and details of algorithms to help readers easier to choose and implement them.

Traditional PLSR only considers the whole intensities of Raman signals without separating the Raman peaks (Raman spectrum) from the instable background. So PLSR itself can not solve the instable background problem mentioned in section 1.1.4. Inspired by the work of [25], in chapter 3, we design a new continuous wavelet transform (CWT) based PLSR method (CWT-PLSR) that only uses the Raman peaks to do the quantitative analysis. This method can effectively reduce random noise and avoid the influences of instable backgrounds and noisy peaks. It can omit the time consuming preprocessing, such as smoothing, de-noising and baseline correction, and so is more convenient.

As will be explained in chapter 2, LVR methods, including Principal Component Regression (PCR), Reduce Rank Regression (RRR) and Partial Least Square Regression (PLSR), actually combine feature extraction and multiple multivariate linear regression. PCR gives high weights to big Raman peaks that span big intensity variances, but may ignore weak peaks that are highly related to concentrations, so PCR is not effective enough; RRR extracts features in an opposite way, giving high weights to peaks whose intensities are highly correlated with concentrations, but may ignore the strong peaks that have big variances, so RRR is not robust enough; PLSR balances two objectives, so is more robust than RRR, and more effective than PCR.

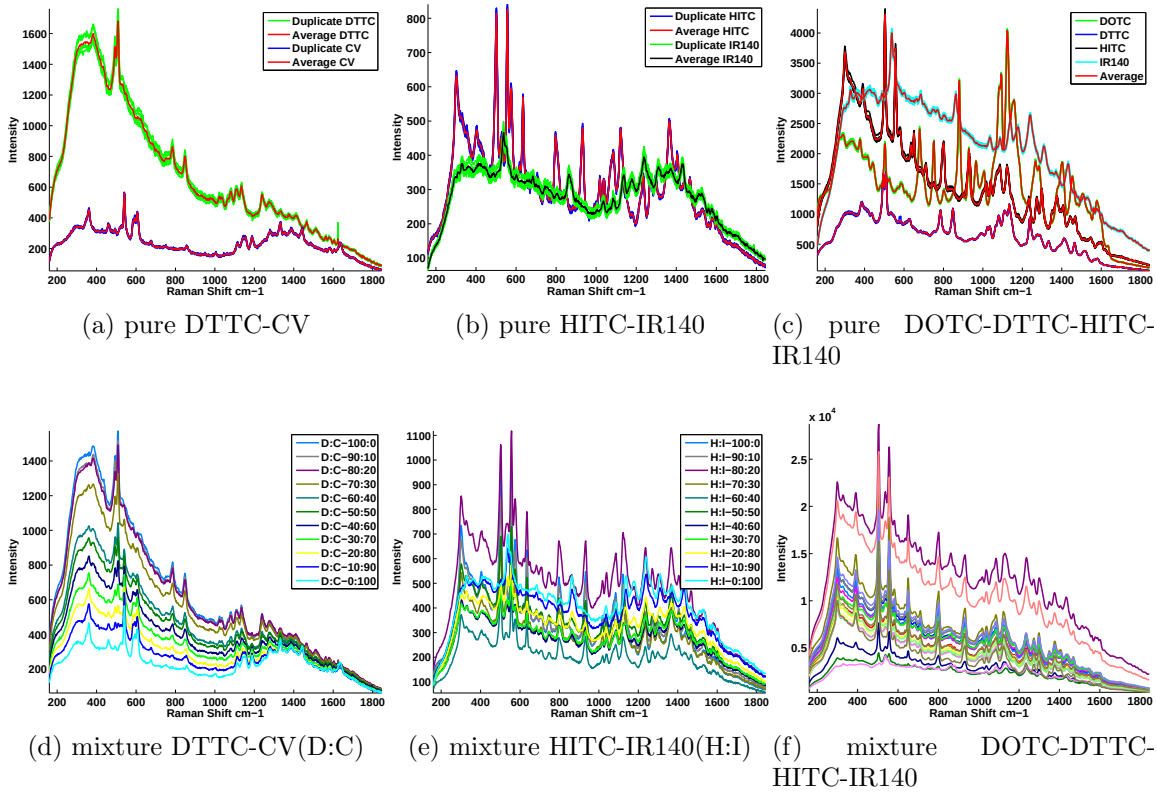
But the balance is not flexible enough. Continuum regression methods (CR) can adjust the proportions of two objectives in the objective function, so the weights can be assigned to the Raman peaks in an optimized way. In chapter 4, we give a new continuum regression method (NCR) that embraces PCR, RRR and PLS as three special cases.

The number of features (components) extracted by PLSR methods have to be determined by the time consuming cross-validation methods. Also the tuning parameter of CR has infinite possible values, there is no way to test every values by cross-validation methods. Bayesian nonparametrics models of these methods are needed to decide these parameters automatically from the training data. In chapter 5, we design a probabilistic PLS regression model that provides a probabilistic view of the traditional PLSR model. It also provide a foundation to develop further Bayesian PLS model.

1.3 Experiment Setup Description

1.3.1 Notations

In this paper, the Raman spectrum and the Raman signal of the pure Nano-Tags are called source spectrum and source signal, noted as the $(D_x \times 1)$ vector $\tilde{\mathbf{s}}$ and $\mathbf{s}$. The Raman spectrum and the Raman signal of the mixture Nano-Tags are called mixture spectrum and mixture signal, noted as the $(D_x \times 1)$ vector $\tilde{\mathbf{x}}$ and $\mathbf{x}$. Rows of the $(D_y \times D_x)$ matrixes $\tilde{\mathbf{S}} = [\tilde{\mathbf{s}}_1, ..., \tilde{\mathbf{s}}_{D_y}]^T$ and $\mathbf{S} = [\mathbf{s}_1, ..., \mathbf{s}_{D_y}]^T$ contains all D_y source spectra and preprocessed source signals, and each of them has D_x Raman shifts. Source signals are collected from the solutions of pure materials with the concentration α. Rows of the $(N \times D_x)$ matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^T$ are N preprocessed mixture signals obtained from samples of mixed Nano-Tags, and each sample is mixed



(a) pure DTTC-CV (b) pure HITC-IR140 (c) pure DOTC-DTTC-HITC-IR140

(d) mixture DTTC-CV(D:C) (e) mixture HITC-IR140(H:I) (f) mixture DOTC-DTTC-HITC-IR140

Figure 1.3: Pure and mixture Raman signals of different Nano-Tags. D:C-90:10, for example, means the ratio of volume of DTTC and CV is 90% : 10%

by those D_y pure Nano-Tags. Rows of the $(N \times D_y)$ matrix $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N]^T$ are the corresponding ground truth ratios of mixing volumes of those pure Nano-Tags.

1.3.2 Data Sets

The data sets we are working on are collected from the Raman spectroscopy system with $20\times$, 0.4_{NA} lens and $785nm$ laser wavelength. Raman shifts range from -79.65cm^{-1} to 2071.80cm^{-1} with 1044 Raman shifts values. All Nano-Tags are made from $54.67nm$ Au Nano-particles, coated with the dyes: DTTC and Cresyl violet (CV) (in data set 1); HITC and IR140 (in data set 2); DOTC, DTTC, HITC and IR140 (in data set 3). The pure Nano-Tag solutions are prepared with a concentration of $1.1e^{10}$

Nano-Tags/ml. Then with 11 ratios of volume $\{(0:100\%),(10\%:90\%),\ldots,(90\%:10\%),(100\%:0)\}$ we mix two pure Nano-Tags solutions in the first two groups, and with 21 ratios of volume $\{(25\%:25\%:25\%:25\%),(20\%:25\%:25\%:25\%),(15\%:25\%:25\%:25\%),\ldots,(0:25\%:25\%:25\%),(25\%:20\%:25\%:25\%),(25\%:15\%:25\%:25\%),\ldots,(25\%:25\%:25\%:5\%),(25\%:25\%:25\%:0)\}$, we mix four pure Nano-Tags solutions in the third group (the missing percentages are made up with water), and get three groups of mixture Nano-Tag solution samples. For each pure and mixture Nano-Tag solution sample, we collect 5 duplicated Raman signals, with $20s$ integration. To avoid the affect of the strong signal intensity from Rayleigh scattering and some noise frequencies on two sides, from 1044 Raman shifts, we extract the middle range (90th-900th) frequencies (from $160.13cm^{-1}$ to $1845.00cm^{-1}$). So for data set 1 and 2 (DTTC-CV and HITC-IR140), the ground truth ratio matrix is $\mathbf{Y} \in \Re^{55\times 2}$, and the mixture spectra matrix is $\mathbf{X} \in \Re^{55\times 811}$. And for data set 3 (DOTC-DTTC-HITC-IR140), the ground truth ratio matrix is $\mathbf{Y} \in \Re^{105\times 4}$, and the mixture signals matrix is $\mathbf{X} \in \Re^{105\times 811}$. Also we average each 5 duplicates for both pure and mixture signals, and get pure average signals $\bar{\mathbf{S}} \in \Re^{2\times 811}$ (data set 1,2) and $\bar{\mathbf{S}} \in \Re^{4\times 811}$ (data set 3); and mixture average signals $\bar{\mathbf{X}} \in \Re^{11\times 811}$ and $\bar{\mathbf{X}} \in \Re^{21\times 811}$, and the average ground truth ratio matrixes $\bar{\mathbf{Y}} \in \Re^{11\times 2}$ and $\bar{\mathbf{Y}} \in \Re^{21\times 4}$. The duplicate and average source signals are shown in Fig. 1.3a-Fig. 1.3c, and the average mixture signals are shown in Fig. 1.3d-Fig. 1.3f.

1.3.3 Evaluation Methods and Criteria

In order to take fully use of all three data sets, we use three cross-validation methods to evaluate the predicting ability of methods:

- Cross-Validation on Duplicate testing signals (CVD): all 5 duplicate mixture signals collected from the same sample are treated as the testing samples, and

all the other duplicate mixture signals are treated as the training samples. Iteratively, until every duplicate signal is treated as the testing sample once.

- Cross-Validation on Average testing signals (CVA): the average signal of the 5 duplicate mixture signals collected from the same sample is treated as the testing sample and all the other duplicate mixture signals are treated as training samples. Iteratively, until every average signal is treated as the testing sample once.
- Cross-Validation on Average testing Average training spectra (CVAA): one average signal is treated as the testing sample and all the other average signals are treated as training samples. Iteratively, until every average signal is treated as the testing sample once.

Square Root of Mean Squares Error (RMSE) is used as the criterion for evaluating the prediction accuracy. It is defined as:

$$RMSE = \sqrt{\frac{1}{ND_y}\sum_{i=1}^{N}\sum_{j=1}^{D_y}(\hat{y}_{i,j} - y_{i,j})^2},$$

in which, $\hat{y}_{i,j}$ and $y_{i,j}$ are elements of the matrix of estimated ratios $\hat{\mathbf{Y}}$ and ground truth ratios $\mathbf{Y}$ respectively of the ith sample and the jth Nano-Tag. D_y is number of pure Nano-Tags and N is number of testing signals.

CHAPTER 2

MODELS AND METHODS COMPARISON

In this chapter, we analyze the mathematical definitions and essential meanings of those models and methods, explain the suitable situation for each method and illustrate PLSR is more reasonable when doing the quantitative analysis of SERS. Also, since there are several variants and algorithms of PLSR methods, we analyze the differences between variants and details of algorithms to help readers easier to choose and implement them. It is organized as following, in section 2.1, we compare several models, including direct classical least squares model, full spectrum calibration model, selected (or weighting) calibration model and latent variable regression (LVR) model. Based on the properties of Raman spectra, we demonstrate LVR is better than the other models. So in section 2.2, Principal component regression (PCR), reduced-rank regression (RRR), orthonormalized PLS (OPLS), partial least squares regression (PLSR), canonical correlation regression (CCR), PLS Wold 2-block mode A (PLS-W2A) and robust canonical analysis (RCA) are compared to show only PLSR algorithms extract features for both representing and predicting purposes.

2.1 Models for Quantitative Analysis of SERS

2.1.1 Direct Classical Least Squares (DCLS)

Based on the properties of Raman spectra, theoretically the mixture spectrum can be modeled as a linear combination of the source spectra with the mixing ratios as the weights:

$$\tilde{\mathbf{x}} = \tilde{\mathbf{S}}^T \mathbf{y} + \tilde{\mathbf{e}}, \tag{2.1}$$

where elements of the $D_x \times 1$ vector $\tilde{\mathbf{e}}$ are the random noises at all Raman shifts. For example, two solutions of pure Nano-Tags, whose source spectra are $\tilde{\mathbf{s}}_1$ and $\tilde{\mathbf{s}}_2$, and both with the concentration of α, are mixed into one sample with the ratio of mixing volume as 30% : 70% ($\mathbf{y}$ is $[0.3, 0.7]^T$). Then the concentration of each Nano-Tag in the mixture sample is 0.3α and 0.7α. Based on the property (c), the two source spectra are approximately $0.3\tilde{\mathbf{s}}_1$ and $0.7\tilde{\mathbf{s}}_2$. And based on the property (a) and (b), the mixture spectrum should approximately be $0.3\tilde{\mathbf{s}}_1 + 0.7\tilde{\mathbf{s}}_2 = \tilde{\mathbf{S}}^T\mathbf{y}$. By minimizing the sum square errors $\tilde{\mathbf{e}}^T\tilde{\mathbf{e}}$, the mixing concentrations are estimated as

$$\alpha\hat{\mathbf{y}} = \alpha(\tilde{\mathbf{S}}\tilde{\mathbf{S}}^T)^{-1}\tilde{\mathbf{S}}\tilde{\mathbf{x}}. \tag{2.2}$$

In a Raman signal, usually the positions of Raman peaks only occupy a small part of all Raman shifts. And in the sum square errors, too many mixing errors are from the background noises, which will affect the estimated result. So naturally a modified way is to only use D selected Raman peaks from the source and mixture spectra, where $D_y < D < D_x$, and the model is changed into:

$$\tilde{\mathbf{x}}_s = \tilde{\mathbf{S}}_s^T\mathbf{y} + \tilde{\mathbf{e}}_s, \tag{2.3}$$

with the $D \times 1$ vector $\tilde{\mathbf{x}}_s$ and $D_y \times D$ matrix $\tilde{\mathbf{S}}_s$ are selected subsets of $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{S}}$ respectively.

DCLS model is simple since it only requires source spectra as the training data. But in reality it is difficult to get the perfect source spectra $\tilde{\mathbf{S}}$ and mixture spectra $\tilde{\mathbf{x}}$ in (2.1), instead, we can only get the preprocessed mixture signal $\mathbf{x}$ a nd source signals $\mathbf{S}$, which can be expressed as

$$\mathbf{x} = \tilde{\mathbf{x}} + \mathbf{e}_x \text{ and } \mathbf{S} = \tilde{\mathbf{S}} + \mathbf{E}_s, \tag{2.4}$$

where $\mathbf{e}_x$ and rows of $\mathbf{E}_s$ are the unremoved background and noises or preprocessing errors. So in model (2.1), all the estimations are based on certain unreliable source signals, which may cause the biased results.

One way to reduce the bias is to treat $\tilde{\mathbf{S}}$ as unknown parameters, and use a batch of mixture signals together with their ground truth mixing concentrations to directly find the relationship between mixture signals and concentrations. In the following subsections, the models we will introduce are based on this idea.

2.1.2 Multivariate Calibration

Combine model (2.4) and model (2.1), we can get $\mathbf{x} = \tilde{\mathbf{S}}^T\mathbf{y} + \tilde{\mathbf{e}} + \mathbf{e}_x$. If we observe N mixture signals, together with their ground truth mixing concentrations, we can get the multivariate calibration model [26] as

$$\mathbf{X} = \mathbf{Y}\tilde{\mathbf{S}} + \tilde{\mathbf{E}} + \mathbf{E}_x, \tag{2.5}$$

in which $\mathbf{X}$ and $\mathbf{Y}$ are given, unknown parameters $\tilde{\mathbf{S}}$ need to be estimated, rows of $\tilde{\mathbf{E}}$ and $\mathbf{E}_x$ are random noises and preprocessing errors. The purpose of the calibration is from spectra to estimate concentrations, so it is convenient to rewrite (2.5) into a multiple multivariate linear regression model

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E}_r + \tilde{\mathbf{E}}_r, \tag{2.6}$$

where the matrix of regression coefficients $\mathbf{B}$ is actually the general inverse matrix of $\tilde{\mathbf{S}}$, $\mathbf{E}_r = -\mathbf{E}_x\mathbf{B}$ can be treated as the bias items and $\tilde{\mathbf{E}}_r = -\tilde{\mathbf{E}}\mathbf{B}$ is the matrix of regression errors. Instead of solving (2.6), normally it is equivalent to solve

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \tilde{\mathbf{E}}_r, , \tag{2.7}$$

with $\mathbf{X}$ and $\mathbf{Y}$ are centered (zero-mean matrixes). This model uses all Raman shifts information, so it is called Full Spectrum Multivariate Calibration (FSMC). Also since

(2.6) is a least squares model, model (2.5) is also called inverse least squares (ILS) in [27]. If the rank of the covariance matrix $\mathbf{X}^T\mathbf{X}$ equals to D_x, by minimizing the trace of $\mathbf{E}_r\mathbf{E}_r^T$, $\mathbf{B}$ is solved as:

$$\hat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{2.8}$$

Then given a new preprocessed testing mixture signal $\mathbf{x}$, the mixing concentrations of each Nano-Tag can be predicted as $\hat{\mathbf{y}} = \hat{\mathbf{B}}^T(\mathbf{x} - \mu_x) + \mu_y$, where μ_x and μ_y are the mean vectors of rows of $\mathbf{X}$ and $\mathbf{Y}$.

Zavaleta *et al.* [4] claimed that source Raman spectra will not change when the pure Nana-Tags are injected into a living organism, nor did they change as a function of tissue depth. So in real applications, it is practical to do the calibration *in vitro*, and do the predictions *in vivo*.

Usually there are hundreds of or thousands of Raman shifts, and only small amount of samples are available, so one problem of FSMC is the covariance matrix $\mathbf{X}^T\mathbf{X}$ in (2.8) is not invertible. To solve this singularity problem, an SVD based method [28] can be used to calculate the generalized inverse of $\mathbf{X}^T\mathbf{X}$ (described in Appendix A.1). But this general inverse method can not deal with the overfitting problem [29] caused by the limited number of training data.

Normally the overfitting will cause very big absolute values of elements of $\hat{\mathbf{B}}$. So to solve the overfitting problem, ridge regression (RR) [30] adds a constraint $||\mathbf{B}||_F < \tau$, where $||.||_F$ is the Frobenius norm defined as $||\mathbf{B}||_F = tr(\mathbf{B}^T\mathbf{B})$, and $tr(.)$ is the trace of the matrix. The solution of RR is:

$$\hat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X} + \kappa\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}, \tag{2.9}$$

where $\mathbf{I}$ is the $(D_x \times D_x)$ identity matrix and κ is a small number. RR solves the overfitting problem in FSMC model, the limitation is that it treats all Raman shifts

equally in the regression and sometimes Raman peaks want to have more weights to estimate the regression coefficients.

A simple way is first to select K Raman shifts out of all by some feature selection methods, where $K < N$. Then the training samples becomes $(N \times K)$ matrix $\mathbf{X}_s$, and the estimated matrix of coefficients becomes $\hat{\mathbf{B}} = (\mathbf{X}_s^T \mathbf{X}_s)^{-1} \mathbf{X}_s^T \mathbf{Y}$. This is called Selected Multivariate Calibration (SMC) [26] Model. The problem of SMC is that some weak Raman peaks and background Raman shifts that contain quantitative information will easily be discarded, which affects the accuracy.

2.1.3 Latent Variable Regression (LVR)

To deal with the information lost problem in SMC model, instead of choosing only a few Raman shifts, latent variable regression (LVR) model [31, 32] linearly combines all Raman shifts (variables) with K groups of weights $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_K] \in R^{D_x \times K}$, and extracts features

$$\mathbf{T} = \mathbf{X}\mathbf{W}, \tag{2.10}$$

where $\mathbf{T} = [\mathbf{t}_1, ..., \mathbf{t}_K]$ is a $(N \times K)$ matrix containing K unrelated Latent Variables (LVs) of the zero-mean matrix $\mathbf{X}$. Then the multivariate regression is done between $\mathbf{T}$ and the zero-mean matrix $\mathbf{Y}$

$$\mathbf{Y} = \mathbf{T}\mathbf{C} + \tilde{\mathbf{E}}_r, \tag{2.11}$$

in which $\mathbf{C}$ is the matrix of regression coefficients, $\tilde{\mathbf{E}}_r$ is the same as the one in (2.7). If the Raman signals are thought as data points in a high dimensional space, the small numbers of data points actually stay in a low dimensional subspace. LVR model is actually to find a K-dimensional subspace to project those data points, then

the regression analysis is done between the projections $\mathbf{T}$ and the concentrations $\mathbf{Y}$. Similar to (2.8), $\mathbf{C}$ is solved as:

$$\mathbf{C} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{Y}. \tag{2.12}$$

In LVR model, the regression coefficients matrix $\mathbf{B}$ in (2.6) is actually decomposed as:

$$\mathbf{B} = \mathbf{W}\mathbf{C} = \mathbf{W}(\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W})^{-1}\mathbf{W}^T\mathbf{X}^T\mathbf{Y}. \tag{2.13}$$

Given a testing mixture signal $\mathbf{x}$, the concentrations $\hat{\mathbf{y}}$ can be predicted as:

$$\hat{\mathbf{y}} = \mathbf{B}^T(\mathbf{x} - \mu_x) + \mu_y. \tag{2.14}$$

With a reasonable choice of numbers of extracted features K, normally selected by a cross-validation method, LVR can reduce the overfitting problem, and so is more robust than FSMC. The linear combination or the projection in (2.10) is actually the feature extraction, with columns of $\mathbf{T}$ are the new extracted features. So comparing RR, LVR also provides a way for biological feature extraction. And since the extracted features contain the information of all Raman shifts (linear combinations of them), it avoids the information lost problem in SMC.

2.2 LVR Methods

The weights vectors $\mathbf{W}$ in (2.10) are usually found by solving a constraint optimization problem. For different purposes of feature extraction, different LVR methods have different objective functions and constraints. In this section, we compare those purposes and formulations of several LVR methods, including Principal Component Regression (PCR), Reduce Rank Regression (RRR), Orthonormalized Partial Least Squares Regression (OPLSR) and Partial Least Square Regression (PLSR), Canonical Correlation Regression (CCR), PLS Wold 2-block mode A (PLS-W2A) and robust

canonical analysis (RCA). Then explain why PLSR methods (including PLS2 and SIMPLS) are better than other methods for quantitative analysis of SERS. In the following subsections, both $\mathbf{X}$ and $\mathbf{Y}$ are zero-mean matrixes. All the indexes i and j in the following formulations are defined as $i = 1, 2, \ldots, K$ and $j = 1, 2, \ldots, (i-1)$.

2.2.1 Principal Component Regression (PCR)

PCR uses Principal Component Analysis (PCA) to find $\mathbf{W}$. PCA is a technique widely used for dimensionality reduction and feature extraction [29]. The goal of PCA is to make $\mathbf{T}$ retains the variation or information presented in $\mathbf{X}$ as much as possible. The successive formulation of PCA is to seek $\mathbf{w}_i$ one by one, and at each time get an unrelated component (LV) $\mathbf{t}_i$ that has the biggest variance:

$$\text{obj. } \max_{\mathbf{w}_i} J_{pcr1} = var(\mathbf{t}_i) \text{ s.t. } ||\mathbf{w}_i|| = 1;\ \mathbf{t}_i^T \mathbf{t}_j = 0, \tag{2.15}$$

with $var(\mathbf{t}_i) = \mathbf{t}_i^T \mathbf{t}_i$ is the sample variance of $\mathbf{t}_i$. $||\mathbf{w}_i|| = \mathbf{w}_i^T \mathbf{w}_i$ is the Euclidean norm of $\mathbf{w}_i$. The first constraint normalizes the lengthes of the weight vectors as 1. The second constraint makes sure the new LV is unrelated (independent) with the previous ones. The number of components (LVs) we can have is limited by the rank of $\mathbf{X}$. From (2.15), weight vectors $\{\mathbf{w}_i\}_{i=1}^K$ are solved as the first K eigenvectors of $\mathbf{X}^T\mathbf{X}$ (in this paper, the first K eigenvectors means the first K eigenvectors corresponding to the first K biggest eigenvalues). Since the covariance matrix $\mathbf{X}^T\mathbf{X}$ is a symmetric matrix and all eigenvectors are orthonormal, the K-dimensional subspace is spanned by columns of $\mathbf{W}$.

Another formulation called the simultaneous formulation of PCA is to find all $\mathbf{w}_i$ at once:

$$\text{obj. } \max_{\mathbf{W}} J_{pcr2} = \sum_{i}^{K} (\mathbf{t}_i^T \mathbf{t}_i) \text{ s.t. } ||\mathbf{w}_i|| = 1;\ \mathbf{t}_i^T \mathbf{t}_j = 0. \tag{2.16}$$

Jolliffe [33] proved the equivalence between the simultaneous and successive formulation of PCA, and gave the solution of $\mathbf{W}$ also as the first K eigenvectors of $\mathbf{X}^T\mathbf{X}$.

Beside maximizing variances of LVs, another objective function of PCA is to minimize the representing error [29] or to minimize the information in the residual matrix. The corresponding formulation is

$$\min_{\mathbf{W}} J_{pcr3} = ||\mathbf{X} - \mathbf{T}\mathbf{P}_x^T||_F, \tag{2.17}$$

with $||.||_F$ is the Frobenius norm, and columns of $\mathbf{P}_x$ are loading vectors. The equivalence between (2.17) and (2.16) is proved in Appendix A.2.

Both two objective functions of PCA, maximizing total variances (2.16) and minimizing representation errors (2.17), show the score vectors of PCA $\mathbf{T}$ are the best K dimensional representation of $\mathbf{X}$. Or say, $\mathbf{W}$ of PCR gives more weights to the Raman shifts that have bigger variances of the intensities, which may not be the locations of Raman peaks, but the random peaks or noisy peaks. And some weak Raman peaks that are highly correlate with $\mathbf{Y}$, may not get big weights. So for prediction purpose, LVs of PCR may not be efficient enough.

2.2.2 Reduced-Rank Regression (RRR) and Orthonormalized Partial Least Squares Regression (OPLSR)

The goal of RRR [34] is to make $\mathbf{T}$ the best rank K approximation to $\mathbf{Y}$. The objective function is to minimize the approximation (or regression) error. Plus the constraints of unrelated (independent) LVs, the formulation of RRR is

$$\text{obj. } \min_{\mathbf{W}} J_{rrr} = ||\mathbf{Y} - \mathbf{T}\mathbf{P}_y^T||_F \text{ s.t. } \mathbf{t}_i^T\mathbf{t}_j = 0. \tag{2.18}$$

Columns of $\mathbf{P}_y$ are the regression coefficients between $\mathbf{t}_i$ and $\mathbf{Y}$. The calculation of $\mathbf{P}_y$ is in Appendix A.3. The solution of $\mathbf{W}$ can be calculated [35] as the first K eigenvectors of $(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})$.

The purpose of OPLSR [36] is to extract features that are correlate with columns of $\mathbf{Y}$ most without considering the variances of LVs. The objective function is described as [37, 38]:

$$\text{obj. } \max_{\mathbf{W}} J_{opls} = tr(\mathbf{T}^T\mathbf{Y}\mathbf{Y}^T\mathbf{T}) \text{ s.t. } \mathbf{T}^T\mathbf{T} = I, \tag{2.19}$$

which is actually derived from the formulation of

$$\text{obj. } \max_{\mathbf{W}} tr\{(\mathbf{T}^T\mathbf{T})^{-1}(\mathbf{T}^T\mathbf{Y}\mathbf{Y}^T\mathbf{T})\} \text{ s.t. } \mathbf{t}_i^T\mathbf{t}_j = 0, \tag{2.20}$$

which is proved in Appendix A.3. (2.20) can be rewritten as

$$\text{obj. } \max_{\mathbf{W}} \sum_{i=1}^{K} (var(\mathbf{t}_i)^{-1}||cov(\mathbf{t}_i, \mathbf{Y})||^2) \text{ s.t. } \mathbf{t}_i^T\mathbf{t}_j = 0, \tag{2.21}$$

which is also proved in Appendix A.3. $var(.)$ is defined in (2.15) and $cov(\mathbf{t}_i, \mathbf{Y}) = \mathbf{t}_i^T\mathbf{Y}$ is the sample covariance vector between $\mathbf{t}_i$ and each column of $\mathbf{Y}$. $||cov(\mathbf{t}_i, \mathbf{Y})||^2 = \sum_{j=1}^{D_y} cov(\mathbf{t}_i, \mathbf{y}_j)^2 = \mathbf{t}_i^T\mathbf{Y}\mathbf{Y}^T\mathbf{t}_i$. (2.20) is equivalent to the formulation of RRR in (2.18), which is proved in Appendix A.3. That is to say OPLSR and RRR are essentially the same. Their $\mathbf{W}$ gives more weights to the Raman shifts, where the intensities of Raman spectra are more correlated with the concentrations $\mathbf{Y}$, without considering representing the Raman spectra. The problem is it may give more weights to the weak Raman peaks or even background that are more correlated with the concentrations, instead of the main Raman peaks with higher intensities. So they are sensitive to the small changes of intensities, which make the predictions not robust enough.

2.2.3 PLS Regression (PLSR)

Developed from the partial least squares (PLS) technique, which is originally designed to model the relationships between several data blocks or sets of variables [39],

and is achieved by the nonlinear iterative partial least squares (NIPALS) algorithm, PLSR algorithms, mainly including PLS2 and SIMPLS, are especially used for the purpose of regression and prediction [40]. PLS2 is based on the NIPALS algorithm [40–43]. For the special case of one dimensional response variable $\mathbf{Y}$, it is called PLS1 [44]. Hoskuldsson [42] analyzed several statistical properties of PLS2 and Wold *et al.* [40] gave a good picture of PLS2. Another PLSR algorithm, designed by de Jong [45], is called SIMPLS algorithm, which improves NIPALS by avoiding the deflation on original data matrixes.

The purpose of PLSR is to maximize the covariance between $\mathbf{T}$ and concentrations $\mathbf{Y}$ [35]:

$$\text{obj. } \max_{\mathbf{w}_i} ||cov(\mathbf{t}_i, \mathbf{Y})||^2 \text{ s.t. } ||\mathbf{w}_i|| = 1; \mathbf{t}_i^T \mathbf{t}_j = 0, \tag{2.22}$$

where $cov(\mathbf{t}_i, \mathbf{Y})$ is the vector of sample covariances between $\mathbf{t}_i$ and columns of $\mathbf{Y}$. Comparing the objective function in (2.22) with (2.15) and (2.21), we can see PLSR is the compromise between RRR and PCR. It makes $\mathbf{T}$ both represent $\mathbf{X}$ and approximate $\mathbf{Y}$ simultaneously. $\mathbf{W}$ of PLSR gives higher weights to the Raman shifts that have both big variances of intensities and high correlations with concentrations, which are more likely to be the positions of main Raman peaks.

The objective function in (2.22) equals to maximize $\mathbf{w}_i^T \mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X} \mathbf{w}_i$, with the normalization constraint of $\mathbf{w}_i$, for $i = 1$, $\mathbf{w}_1$ is the first eigenvector of $\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}$. But for $i = 2, \ldots, K$, because of the the independence constraint of the LVs, there is no closed form solution to $\mathbf{W}$ in (2.22). PLS2 [44] and SIMPLS [45], are designed to solve this problem. In the following two subsections we will compare the the details of two algorithms.

2.2.3.1 PLS2

PLS2 [40, 42] iteratively deflates on $\mathbf{X}$ to get residual matrix $\mathbf{X}_i$ and get the corresponding projection direction $\mathbf{r}_i$ by solving the following problem:

$$\text{obj. } \max_{\mathbf{r}_i} \mathbf{r}_i^T \mathbf{X}_i^T \mathbf{Y}\mathbf{Y}^T \mathbf{X}_i \mathbf{r}_i, \text{ s.t. } ||\mathbf{r}_i|| = 1, \tag{2.23}$$

with $\mathbf{X}_i$ is got from a deflation process in Algorithm 1, in which, $eig(\mathbf{A})$ means getting the first eigenvector of matrix $\mathbf{A}$ corresponding to the biggest eigenvalue. Hoskuldsson [42] proved that the independence constraint $\mathbf{t}_i^T \mathbf{t}_j = 0$ in (2.22) is satisfied after the deflation process. $\mathbf{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_K]$ in (2.23) are weight vectors of residual matrix $\mathbf{X}_i$, which are different with $\mathbf{W}$. The relation between $\mathbf{r}_i$ and $\mathbf{w}_i$ is $\mathbf{t}_i = \mathbf{X}_i \mathbf{r}_i = \mathbf{X}\mathbf{w}_i$ [46]. After deflation, $\mathbf{W}$ can be calculated as $\mathbf{W} = \mathbf{R}(\mathbf{P}^T\mathbf{R})^{-1}$ [40] or $\mathbf{W} = \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}$ [47].

Algorithm 1 PLS2 Deflation Process

1: **for** $i = 1$ to K **do**
2: $\mathbf{r}_i = eig(\mathbf{X}_i^T \mathbf{Y}\mathbf{Y}^T \mathbf{X}_i)$; % Get projection directions $\mathbf{r}_i$
3: $\mathbf{t}_i = \mathbf{X}_i \mathbf{r}_i$; % Get score vectors $\mathbf{t}_i$
4: $\mathbf{p}_i = \mathbf{X}_i^T \mathbf{t}_i / (\mathbf{t}_i^T \mathbf{t}_i)$; % Get loading vectors $\mathbf{p}_i$
5: $\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i \mathbf{p}_i^T$; % Get the residual matrices of $\mathbf{X}_i$
6: **end for**
7: Store $\mathbf{R} = [\mathbf{r}_1, ..., \mathbf{r}_K]$; $\mathbf{P} = [\mathbf{p}_1, ..., \mathbf{p}_K]$

$\mathbf{r}_i$ is actually the left singular vector of $\mathbf{X}_i^T \mathbf{Y}$. Since the size of matrix $\mathbf{X}_i^T \mathbf{Y}\mathbf{Y}^T \mathbf{X}_i$ is big, step 2 in Algorithm 1 is time consuming. The faster way is to first calculate the right singular vector $\mathbf{d}_i$ by $eig(\mathbf{Y}^T \mathbf{X}_i \mathbf{X}_i^T \mathbf{Y})$, then use the relationship between left and right singular vector, $\mathbf{r}_i = \mathbf{X}_i^T \mathbf{Y}\mathbf{d}_i / ||\mathbf{X}_i^T \mathbf{Y}\mathbf{d}_i||$, to get $\mathbf{r}_i$.

2.2.3.2 SIMPLS

Instead of based on residual matrixes $\mathbf{X}_i$, SIMPLS directly looks for directions of projections in the original $\mathbf{X}$ space by projecting the cross covariance matrix $\mathbf{X}^T\mathbf{Y}$ on to orthogonal subspace $\mathbf{P}_i^{\perp} = \mathbf{I} - \mathbf{P}_{i-1}\mathbf{P}_{i-1}^{+}$ iteratively to satisfy the unrelated constrain, with $\mathbf{P}_{i-1}^{+} = (\mathbf{P}_{i-1}^T\mathbf{P}_{i-1})^{-1}\mathbf{P}_{i-1}^T$ is the Moore-Penrose inverse of $\mathbf{P}_{i-1}$ and $\mathbf{P}_{i-1} = [\mathbf{p}_1, \ldots, \mathbf{p}_{i-1}]$ is the loading matrix. The objective function of SIMPLS in each iteration is:

$$\text{obj. } \max_{\mathbf{w}_i} \mathbf{w}_i^T\mathbf{P}_i^{\perp}\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{P}_i^{\perp}\mathbf{w}_i, \text{ s.t. } ||\mathbf{w}_i|| = 1. \tag{2.24}$$

$\mathbf{w}_i$ can be solved as the first eigenvector of $\mathbf{P}_i^{\perp}\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{P}_i^{\perp}$ corresponding to the biggest eigenvalue. de Jong [45] proved that these $\mathbf{w}_i$ satisfy the constrain $\mathbf{t}_i^T\mathbf{t}_j = 0$ in (2.22). The SIMPLS algorithm is summarized in [45].

2.2.4 Symmetric and Asymmetric Relation

PCR, RRR or OPLSR, PLSR have the asymmetrical relationship between $\mathbf{X}$ and $\mathbf{Y}$ [44], because these methods are from $\mathbf{X}$ to predict $\mathbf{Y}$, and only get LVs of $\mathbf{X}$. These methods are only suitable for the low dimensional and independent response variables $\mathbf{Y}$. If $D_y > N$ or there is collinearity between columns of $\mathbf{Y}$, LVR methods with the symmetrical relationship between $\mathbf{X}$ and $\mathbf{Y}$, which get the LVs for both $\mathbf{X}$ and $\mathbf{Y}$, have to be used. It is called symmetrical relationship because Both $\mathbf{X}$ and $\mathbf{Y}$ can be predictor or response matrix. In the following subsections, we will introduce three such methods, including CCA, PLS-W2A and RCA.

For symmetric relation LVR methods, the components number K is limited by the rank of both $\mathbf{X}$ and $\mathbf{Y}$: $K \leq min(rank(\mathbf{X}), rank(\mathbf{Y}))$. But for asymmetric relation LVR methods, K is limited by the rank of $\mathbf{X}$: $K \leq rank(\mathbf{X})$. In our application of quantitative analysis of Raman spectra, normally we only want to predict

concentrations $\mathbf{Y}$ from Raman spectra $\mathbf{X}$, and the number of pure Nano-Tags (rank of $\mathbf{Y}$) is usually low, which will limit the number of LVs used in the model and so the effectiveness of the model if we use the symmetric relation LVR methods.

2.2.5 Canonical Correlation Regression (CCR)

CCR is based on the technique of canonical correlation analysis (CCA) [48]. The purpose of CCA is to find K pairs of LVs of $\mathbf{X}$ and $\mathbf{Y}$, $\{(\mathbf{t}_i = \mathbf{X}\mathbf{w}_i, \mathbf{u}_i = \mathbf{Y}\mathbf{v}_i)\}_{i=1}^{K}$, such that each LV within each set is only correlated with a single LV in the other set. The objective function is to maximize the correlation coefficients of two LVs. The first way to describe CCA is the successive formulation:

$$\begin{aligned} \text{obj.} \quad & \max_{(\mathbf{w}_i, \mathbf{v}_i)} J_{ccr1} = corr(\mathbf{t}_i, \mathbf{u}_i) \\ \text{s.t.} \quad & \mathbf{t}_i^T \mathbf{t}_j = 0;\ \mathbf{u}_i^T \mathbf{u}_j = 0;\ \mathbf{t}_i^T \mathbf{u}_j = 0;\ \mathbf{t}_j^T \mathbf{u}_i = 0 \end{aligned} \tag{2.25}$$

where $corr(\mathbf{t}_i, \mathbf{u}_i) = \frac{\mathbf{t}_i^T \mathbf{u}_i}{\sqrt{\mathbf{t}_i^T \mathbf{t}_i}\sqrt{\mathbf{u}_i^T \mathbf{u}_i}}$ is the sample correlation coefficient between two LVs. The constraints make sure the components are unrelated. Because the scales of $\mathbf{w}_i$ and $\mathbf{v}_i$ do not affect the correlation coefficient value, it is always proper to fix the projection variance as a constant, usually as 1. So (2.25) is usually rewritten as (e.g., [48, 49]):

$$\begin{aligned} \text{obj.} \quad & \max_{(\mathbf{w}_i, \mathbf{v}_i)} J_{ccr2} = \mathbf{t}_i^T \mathbf{u}_i \\ \text{s.t.} \quad & \mathbf{t}_i^T \mathbf{t}_i = 1;\ \mathbf{u}_i^T \mathbf{u}_i = 1;\ \mathbf{t}_i^T \mathbf{t}_j = 0;\ \mathbf{u}_i^T \mathbf{u}_j = 0;\ \mathbf{t}_i^T \mathbf{u}_j = 0;\ \mathbf{t}_j^T \mathbf{u}_i = 0. \end{aligned} \tag{2.26}$$

Appendix A.4 shows $\mathbf{W}$ and $\mathbf{V}$ are calculated as the first K eigenvectors of matrix $(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})(\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{Y}^T\mathbf{X})$ and $(\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{Y}^T\mathbf{X})(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})$. Besides successive formulation, CCA has another equivalent simultaneous formulation [49, 50]:

$$\begin{aligned} \text{obj.} \quad & \max_{(\mathbf{w}_i, \mathbf{v}_i)} J_{ccr3} = \sum_{i=1}^{K} (\mathbf{t}_i^T \mathbf{u}_i) \\ \text{s.t.} \quad & \mathbf{t}_i^T \mathbf{t}_i = 1;\ \mathbf{u}_i^T \mathbf{u}_i = 1;\ \mathbf{t}_i^T \mathbf{t}_j = 0;\ \mathbf{u}_i^T \mathbf{u}_j = 0;\ \mathbf{t}_i^T \mathbf{u}_j = 0;\ \mathbf{t}_j^T \mathbf{u}_i = 0. \end{aligned} \tag{2.27}$$

As discussed in section 2.1.2, the singularity problem of the covariance matrix $\mathbf{X}^T\mathbf{X}$ in the solution of CCR and RRR or OPLSR can be solved by the generalized inverse of $\mathbf{X}^T\mathbf{X}$ based on SVD (described in Appendix A.1). Another commonly used way (e.g., [49]), similar to RR, named regularized CCA (rCCA), is to add a regularized term κ to the covariance matrix:

$$(\mathbf{X}^T\mathbf{X} + \kappa\mathbf{I})^{-1}(\mathbf{X}^T\mathbf{Y})(\mathbf{Y}^T\mathbf{Y} + \kappa\mathbf{I})^{-1}(\mathbf{Y}^T\mathbf{X})\mathbf{W} = \mathbf{W}\Lambda. \tag{2.28}$$

Similar to RRR and OPLS, $\mathbf{W}$ of CCR also gives higher weights to the Raman shifts that more correlate with LVs of $\mathbf{Y}$. The combinational weights $\mathbf{c}_i$ of columns of $\mathbf{Y}$ make CCR can also deal with the collinearity in matrix $\mathbf{Y}$. When the numbers of pure Nano-Tags in the mixtures are more than sample numbers and the existence of certain pure Nano-Tags are highly correlated in the mixtures, CCR is a substitution of RRR and OPLS.

2.2.6 PLS Wold 2-Block mode A (PLS-W2A)

Another variant of PLS, named PLS-W2A by Wegelin (Wold's Two-block mode A PLS [43]), which is especially used for modeling and predicting between two data blocks [41]. Nowadays it is also commonly used for feature extraction applications [51–53]. PLSR methods only find LVs of dependent variables to relate with each independent variable, and they assume the independent variables are unrelated. Similar to CCR, PLS-W2A can handle the collinearity between columns of $\mathbf{Y}$ and even the ill-conditional problem in matrix $\mathbf{Y}$ (numbers of variables of $\mathbf{Y}$ is bigger than the numbers of samples). It gets K pairs of unrelated LVs $\{(\mathbf{t}_i = \mathbf{X}\mathbf{w}_i, \mathbf{u}_i = \mathbf{Y}\mathbf{v}_i)\}_{i=1}^{K}$ for both data blocks $\mathbf{X}$ and $\mathbf{Y}$ that have the maximum sample covariances:

$$\begin{aligned} &\text{obj.} \quad \max_{\mathbf{w}_i, \mathbf{v}_i} cov(\mathbf{t}_i, \mathbf{u}_i) \\ &\text{s.t.} \quad \mathbf{w}_i^T\mathbf{w}_i = a_i;\ \mathbf{t}_i^T\mathbf{t}_j = 0;\ \mathbf{t}_i^T\mathbf{u}_j = 0;\ \mathbf{v}_i^T\mathbf{v}_i = a_i;\ \mathbf{u}_i^T\mathbf{u}_j = 0;\ \mathbf{t}_j^T\mathbf{u}_i = 0. \end{aligned} \tag{2.29}$$

where a_i is a constant to fix the length of $\mathbf{w}_i$ and $\mathbf{v}_i$. $cov(\mathbf{t}_i, \mathbf{u}_i)$ is the sample covariance between $\mathbf{t}_i$ and $\mathbf{u}_i$.

Similar to PLSR, there is no closed form solution to $\mathbf{W}$ and $\mathbf{V}$ in (2.29). Based on NIPALS algorithm, PLS-W2A uses the deflation process on both $\mathbf{X}$ and $\mathbf{Y}$, which is summarized in Algorithm 2, to satisfy the independence constraints. Here $\mathbf{r}_i$ and $\mathbf{d}_i$ are the projection directions corresponding to the deflated data $\mathbf{X}_i$ and $\mathbf{Y}_i$ instead of the original data $\mathbf{X}$ and $\mathbf{Y}$. For $i = 1$, $\mathbf{X}_1 = \mathbf{X}$ and $\mathbf{Y}_1 = \mathbf{Y}$, so $\mathbf{r}_1$ is the same as $\mathbf{w}_1$ and $\mathbf{d}_1$ is the same as $\mathbf{v}_1$. But for $i = 2, \ldots, K$, they are different. The relation is $\mathbf{t}_i = \mathbf{X}_i\mathbf{r}_i = \mathbf{X}\mathbf{w}_i$ and $\mathbf{u}_i = \mathbf{Y}_i\mathbf{d}_i = \mathbf{Y}\mathbf{v}_i$. After getting all the loading vectors $\mathbf{P} = [\mathbf{p}_1, ..., \mathbf{p}_K]$ and $\mathbf{Q} = [\mathbf{q}_1, ..., \mathbf{q}_K]$, all the combinational weights vectors $\mathbf{W}$ and $\mathbf{V}$ can be calculated as $\mathbf{W} = \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}$ and $\mathbf{V} = \mathbf{Q}(\mathbf{Q}^T\mathbf{Q})^{-1}$.

Algorithm 2 PLS-W2A Deflation Process

1: **for** $i = 1$ to K **do**

2: $\mathbf{r}_i = eig(\mathbf{X}_i^T\mathbf{Y}_i\mathbf{Y}_i^T\mathbf{X}_i); \mathbf{d}_i = eig(\mathbf{Y}_i^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y}_i);$

3: $\mathbf{t}_i = \mathbf{X}_i\mathbf{r}_i; \mathbf{u}_i = \mathbf{Y}_i\mathbf{d}_i;$

4: $\mathbf{p}_i = \mathbf{X}_i^T\mathbf{t}_i/(\mathbf{t}_i^T\mathbf{t}_i); \mathbf{q}_i = \mathbf{Y}_i^T\mathbf{u}_i/(\mathbf{u}_i^T\mathbf{u}_i);$

5: $b_i = \mathbf{u}_i^T\mathbf{t}_i/(\mathbf{t}_i^T\mathbf{t}_i)$;% Get regression coefficient

6: $\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i\mathbf{p}_i^T; \mathbf{Y}_{i+1} = \mathbf{Y}_i - \mathbf{u}_i\mathbf{q}_i^T;$

7: **end for**

2.2.7 Robust Canonical Analysis (RCA)

Tishler *et al.* [54] presented an "Intercorrelations Analysis" method or "Canonical Covariance", which is similar to PLS-W2A. In that paper, RCA is used as modeling method, later Tishler and Lipovetsky [55] presented a RCA regression model for pre-

diction, which solves all eigenvectors of $\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$ at once. Rosipal and Kramer [44] named it as PLS-SB, and Wegelin [43] called it PLS-SVD, since it is actually PLS-W2A without deflation process. The projection directions $\mathbf{W}$ of RCA are the first K eigenvectors of the generalized eigenfunctions:

$$\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{W} = \mathbf{W}\Lambda, \tag{2.30}$$

with the diagonal elements of the diagonal matrix Λ are descending ordered eigenvalues. RCA is a kind of compromise between CCA's non-deflation and PLSW2A's covariance as objective. The math description can be illustrated as:

$$\text{obj. } \max_{\mathbf{w}_i,\mathbf{v}_i} \mathbf{t}_i^T\mathbf{u}_i \text{ s.t. } ||\mathbf{w}_i|| = ||\mathbf{v}_i|| = 1;\ \mathbf{w}_i^T\mathbf{w}_j = \mathbf{v}_i^T\mathbf{v}_j = 0 \tag{2.31}$$

Comparing with Equation (2.29), RCA releases the unrelated components constraints. So LVs in RCA are not necessary mutually unrelated, and there is overlapped information between the LVs and so RCA is not efficient enough.

2.3 Experiment

2.3.1 Methods Specification

For DCLS, in order to reduce the influence of the instability of pure Raman spectra, we use average source signals $\bar{\mathbf{S}}$ as the standard sources to estimate: $\tilde{\mathbf{Y}} = \mathbf{X}\bar{\mathbf{S}}^T(\bar{\mathbf{S}}\bar{\mathbf{S}}^T)^{-1}$. For RR and rCCR, the parameter κ in Equation (2.9) and Equation (2.28) are set as 0.1. To maximize the performance of all the latent variable regression methods, the component number K needs to be optimized for each data set and each cross-validation method. Every possible component number is tested, the one giving the lowest RMSE returns as the optimized one K^*. The results are in Table 2.1. Also we used the iteratively curve-fitting baseline correction method [24] to remove the backgrounds, and the polynomial curve-fitting order *pOrder* needs to

Table 2.1: Optimized components number K^* using two criteria. D is short for evaluation method CVD; A is for CVA; AA is for CVAA

Methods	Data Set 1			Data Set 2			Data Set 3		
	D	A	AA	D	A	AA	D	A	AA
PCR	8	8	8	3	3	3	30	30	20
OPLS	27	25	8	8	8	7	30	30	18
PLS2	4	3	3	3	3	3	20	28	17
SIMPLS	4	3	11	3	3	3	29	29	14

be decided. Different orders (from 3 to 10) are tested, the one giving the lowest RMSE returns as the optimized $pOrder^*$. The results are in Table 2.2.

Table 2.2: Optimized curve fitting order $pOrder^*$ for baseline correction. D is short for evaluation method CVD; A is CVA; AA is CVAA

Methods	Data Set 1			Data Set 2			Data Set 3		
	D	A	AA	D	A	AA	D	A	AA
DCLS	-	8	4	-	3	3	-	6	6
RR	10	6	6	3	3	3	8	8	4
PCR	6	6	6	3	3	3	6	6	4
OPLS	10	6	6	4	4	3	8	8	4
PLS2	6	6	8	3	3	3	6	8	4
SIMPLS	6	6	6	3	3	3	8	8	4
rCCR	10	6	6	3	3	3	8	8	4

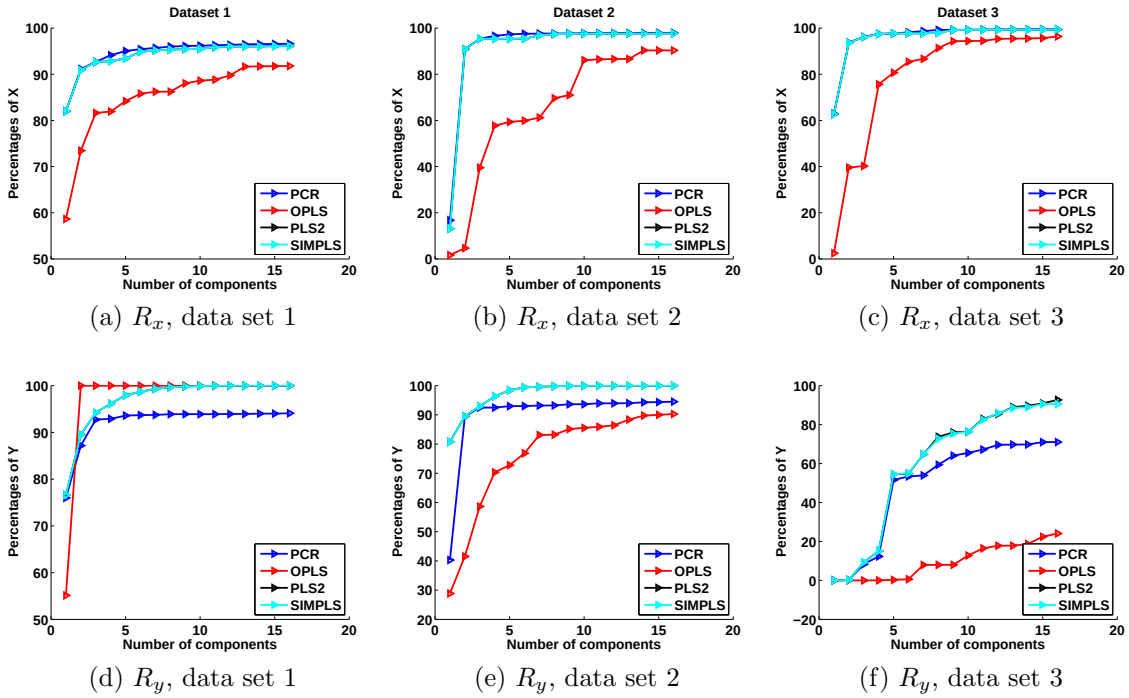
2.3.2 Results

In Table 2.3, we show the estimated errors of different regression methods, using three data sets and three cross-validation methods. The results of CVA and CVAA tend to be better than CVD, since the average testing signals will reduce the influence of instable backgrounds. And CVA is better than CVAA, since CVA has more training samples than CVAA.

Table 2.3: Estimation errors for each methods on three data sets. Each data set use three evaluation methods. The bold face represents the best result.

Methods	Data set 1			Data set 2			Data set 3		
	CVD	CVA	CVAA	CVD	CVA	CVAA	CVD	CVA	CVAA
DCLS	–	6.11	2.87	–	16.44	16.37	–	64.01	64.01
RR	2.75	1.62	1.77	4.37	4.22	4.28	2.85	2.68	2.94
PCR	2.50	1.39	1.73	3.24	3.13	3.14	3.01	2.92	**2.89**
OPLS	2.53	1.39	1.71	4.09	3.96	3.87	3.07	2.92	3.04
PLS2	**2.48**	**1.37**	1.68	**3.18**	**3.07**	**3.06**	**2.83**	2.68	2.93
SIMPLS	2.58	**1.37**	**1.41**	3.30	3.19	3.18	2.84	**2.67**	2.92
rCCR	2.75	1.63	1.77	4.37	4.22	4.28	2.85	2.68	2.94
PLS-W2A	8.08	4.84	4.18	9.40	9.34	9.32	7.33	7.33	7.33
RCA	7.79	4.16	4.10	4.83	4.71	5.61	5.26	5.22	5.23

For DCLS method, CVA tests each duplicate mixture signal and CVAA tests each average mixture signal. We leave CVD as empty. The results of DCLS is worse than other calibration methods, because the source signals are not reliable enough, especially in data set 3. (We find that the heights of the Raman peaks in the mixture are higher than those in source signals. Theoretically they should be lower, since the concentrations of each pure nano-tag is lower in the mixture solution.) Most of the results of LVR methods (PLS2, SIMPLS, PCR, OPLS) are better than those of RR, which means the latent space calibration model is better than the full spectrum calibration model, because there may be noise in certain latent dimensions, and by choosing the optimized component number K^*, most noise are removed. CCR, PLS-W2A and RCA are bad for data set 1 and 2, and relatively better in data set 3. That is because of the limited number of components problem. In data set 1 and 2, rank of $\mathbf{Y}$ is 1, so only one component can be used, and in data set 3, only four components can be used, so the accuracy is relatively increased. In the end, the similar results of PLSR methods proves that PLS2 and SIMPLS are almost the same, and they are



(a) R_x, data set 1 (b) R_x, data set 2 (c) R_x, data set 3

(d) R_y, data set 1 (e) R_y, data set 2 (f) R_y, data set 3

Figure 2.1: Percentages of $\mathbf{X}$ and $\mathbf{Y}$ represented by $\mathbf{T}$ of LVR methods

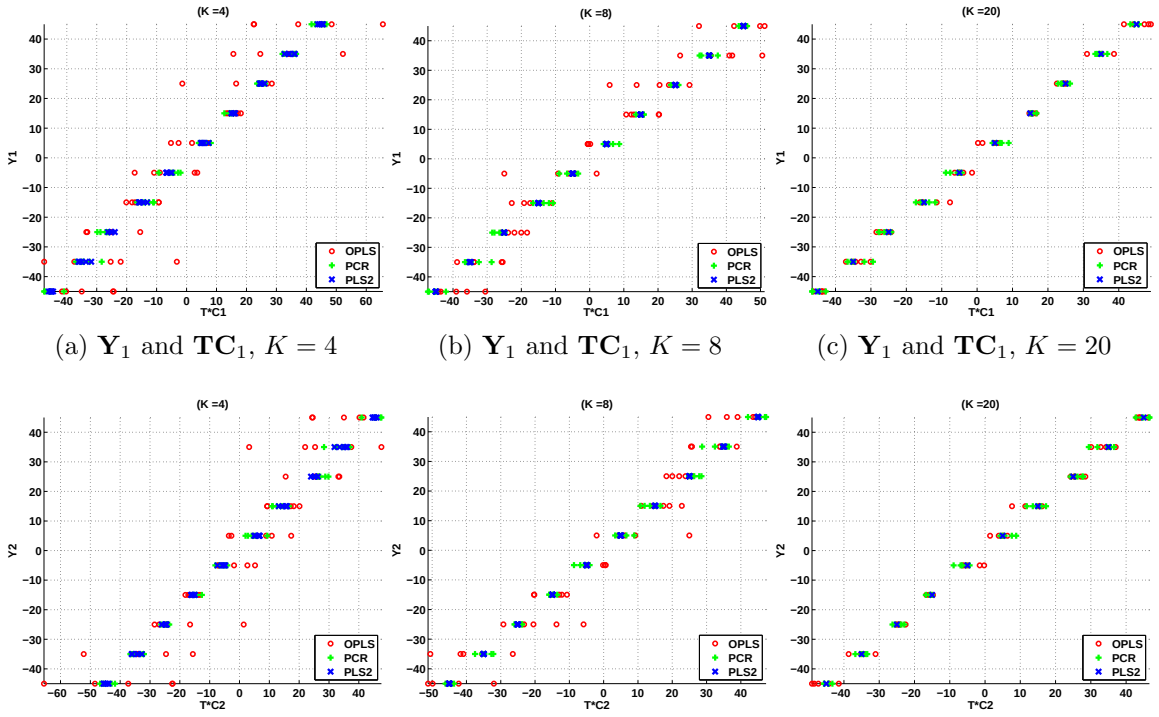
better than the other LVR methods, which shows the robustness of PLSR methods (the reason is explained in section 2.2.3).

2.3.3 Discussion

2.3.3.1 Representation and Prediction Effectiveness

In order to show the effectiveness of latent variables $\mathbf{T}$ to represent the Raman signals $\mathbf{X}$ and to predict the concentration $\mathbf{Y}$, Fig. 2.1 shows how much information of $\mathbf{X}$ and $\mathbf{Y}$ are presented in $\mathbf{T}$, which can be defined as $R_x = (1 - (||\mathbf{X}|| - ||\mathbf{T}\mathbf{P}^T||)/||\mathbf{X}||) \times 100\%$ and $R_y = (1 - (||\mathbf{Y}|| - ||\mathbf{T}\mathbf{C}||)/||\mathbf{Y}||) \times 100\%$.

The curves of PLS2 and SIMPLS are overlapped from Fig. 2.1a to Fig. 2.1e, which means their effectiveness are almost the same. The effectiveness of PLS2, SIMPLS and PCR for representing $\mathbf{X}$ are similar (Fig. 2.1a - Fig.2.1c), but for predicting



(a) $\mathbf{Y}_1$ and $\mathbf{TC}_1$, $K = 4$ (b) $\mathbf{Y}_1$ and $\mathbf{TC}_1$, $K = 8$ (c) $\mathbf{Y}_1$ and $\mathbf{TC}_1$, $K = 20$

(d) $\mathbf{Y}_2$ and $\mathbf{TC}_2$, $K = 4$ (e) $\mathbf{Y}_2$ and $\mathbf{TC}_2$, $K = 8$ (f) $\mathbf{Y}_2$ and $\mathbf{TC}_2$, $K = 20$

Figure 2.2: Relation between latent variables and concentrations, for data set 1. Elements $\mathbf{Y}_1$ are the mixing ratios of CV, and elements of $\mathbf{Y}_2$ are the mixing ratios of DTTC. K is the number of components used in LVR methods

$\mathbf{Y}$, PLS2 and PCR is better than PCR (Fig. 2.1d - Fig.2.1f). The effectiveness of OPLS for both purposes are the worst, and until the number of components is big enough (around 20), it get close to the other three methods. Fig. 2.1 explains why the prediction accuracy of PLSR is better than PCR (Table 2.3), and the components used are less (Table 2.1).

2.3.3.2 Calibration Effectiveness

In order to see the calibration effectiveness of LVR methods, we run OPLS, PLS2 and PCR on data set 1, using three component numbers (K=4, K=8, K=20), and with 50 mixture signals and corresponding mixing ratios as the training samples,

and get the score $\mathbf{T}$ and coefficients $\mathbf{C}$ in (2.11). Fig. 2.2 shows the relation between the ratios of the ith nano-tag $\mathbf{Y}_i$ and $\mathbf{TC}_i$, where $\mathbf{C}_i$ is the ith column of $\mathbf{C}$. It illustrates that the low dimensional representation (or weighted mean) of spectra intensities ($\mathbf{T}$) and the relative concentrations of pure components ($\mathbf{Y}$) are linearly related, which explains why calibration methods work well on Raman spectra data. It also demonstrates the calibration effectiveness of these methods is increasing with more components used, and PLS2 is more effective than PCR and OPLS, because its convergence is the fastest.

2.4 Conclusion

For quantitative analysis of Raman spectra, we group the current methods into four models. And base on the properties of Raman spectra, and the fact that usually data points stay in a low dimensional subspace, we analyze why the latent variable regression model outperforms the other three. Among the LVR methods, we divide them into two groups based on the symmetrical relation between two input matrixes. For our application, asymmetric relation methods are better than symmetric relation methods because the later has the limited component number problem. By comparing the objective function and constraints form of LVR methods, we illustrate PLS2 and SIMPLS are almost the same, both belonging to PLSR methods, and both are better than PCR and OPLS (or RRR) because they are combination of best representation of predictor $\mathbf{X}$ and best prediction of response $\mathbf{Y}$.

CHAPTER 3

CONTINUOUS WAVELET TRANSFORM BASED PLSR

Traditional PLSR only considers the intensities information of Raman signals without separating the Raman peaks from the instable background, which affects the quantity prediction accuracy. Continuous wavelet transform (CWT) is an effective way to extract the peak information and automatically remove the background [25]. To use both peak shape information of Raman spectrum and the correlation between peak heights and concentrations, we design a CWT based PLS (CWT-PLSR) that uses the average CWT coefficients of Raman spectra and the mixing concentrations to do PLS regression. It is robust to random noises and instable baseline and the performance is better than traditional PLSR and baseline correction based PLSR.

3.1 CWT-PLSR

In this section, we introduce a continuous wavelet transform (CWT) based PLSR algorithm which can automatically remove the background and extract the Raman spectrum (peaks). CWT [56] can be described as:

$$\mathbf{C}(a,b) = \int_R x(\tau)\psi_{a,b}(\tau)d\tau, \tag{3.1}$$

with $x(\tau)$ is one Raman signal, τ is the time variable, here means different Raman shifts, $\psi_{a,b}(\tau) = \frac{1}{\sqrt{a}}\psi(\frac{\tau-b}{a})$ is any scaled and translated wavelet function, $a = 1, 2, ..., s$ is the scale, $b = 1, 2, ..., D_x$ is the translation and $\mathbf{C}(a,b)$ is the 2D matrix of wavelet coefficients.

3.1.1 CWT-PLSR Algorithm

The CWT-PLS algorithm includes two parts: training (modeling) part and testing (predicting) part. Given training data: mixture Raman signals $\mathbf{X}$ and mixing concentrations $\mathbf{Y}$, maximum wavelet scale s and PLS components number K, the training part is:

1. For every Raman signal (each row of $\mathbf{X}$), get its CWT coefficients $\mathbf{C}$ in (3.1) with Mexican hat mother wavelet [25];

2. Calculate the average coefficients of $\mathbf{C}$ along the scale dimension as $Mean(\mathbf{C}) = \frac{1}{s}\sum_{a=1}^{s}\mathbf{C}(a,b)$, and store them in one row of matrix $\mathbf{D}$;

3. Instead of using $\mathbf{X}$, using $\mathbf{D}$ and $\mathbf{Y}$ to do PLSR, and return the PLSR coefficients $\mathbf{\Theta}$;

Then given a testing Raman signal $\mathbf{x}$, the testing part is:

1. Get the CWT coefficients $\mathbf{C}$ of $\mathbf{x}$, and calculate its average coefficients $\mathbf{d}$;

2. Estimate the mixing concentrations $\mathbf{y}$.

The whole algorithm is summarized in Algorithm 3. The definition of Mexican hat function [56] is

$$\psi(\tau) = (\frac{2}{\sqrt{(3)}}\pi^{-1/4})(1-\tau^2)e^{-\tau^2/2}. \tag{3.2}$$

Curves of Mexican hat wavelet function with 4 different scales are shown in Fig. 3.1.

3.1.2 Principles of CWT-PLSR

3.1.2.1 Remove the Baseline and Random Noise

The baseline of the Raman signal is assumed to change slowly and monotonically in any small region, so it can be locally approximated as a constant G plus an odd

Algorithm 3 CWT-PLSR Algorithm

Input: $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{x}$, K, s

Output: $\mathbf{y}$

1: **for** $i = 1$ to N **do**

2: $\quad \mathbf{C} = CWT(\mathbf{X}(i,:), s)$;

3: $\quad \mathbf{D}(i,:) = Mean(\mathbf{C})$;

4: **end for**

5: $\mathbf{\Theta} = PLSR(\mathbf{D}, \mathbf{Y}, K)$;

6: $\mathbf{C} = CWT(\mathbf{x}, s)$;

7: $\mathbf{d} = Mean(\mathbf{C})$;

8: $\mathbf{y} = (\mathbf{d} - Mean(\mathbf{D}))\mathbf{\Theta} + Mean(\mathbf{Y})$;

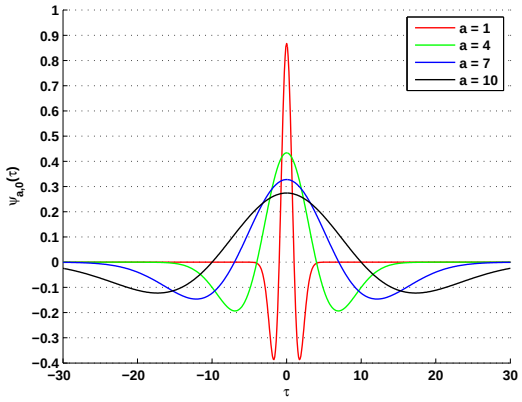


Figure 3.1: Mexican hat wavelet functions with different scales.

function $B(\tau)$. Similar to [25], the intensities of the Raman signal $x(\tau)$ at any small region $[\tau_1, \tau_2]$ can be represented as following:

$$x(\tau) = P(\tau) + B(\tau) + G + E(\tau);\ \tau \in [\tau_1, \tau_2], \tag{3.3}$$

If $[\tau_1, \tau_2]$ is the region of the Raman peak, $P(\tau)$ is the real Raman peak; otherwise if it is the region of the background, $P(\tau) = 0$. $B(\tau)$ is the background function with

zero mean, G is a constant, $E(\tau)$ is the random noise. The coefficients in (3.1) can be rewritten as:

$$\begin{aligned}\mathbf{C}(a,b) = &\int_R P(\tau)\psi_{a,b}(\tau)d\tau + \int_R B(\tau)\psi_{a,b}(\tau)d\tau \\ &+ \int_R G\psi_{a,b}(\tau)d\tau + \int_R E(\tau)\psi_{a,b}(\tau)d\tau.\end{aligned} \tag{3.4}$$

Because the wavelet function $\psi_{a,b}(\tau)$ is a zero-mean function, the third term in (3.4) is zero. And for symmetric wavelet, like Mexican Hat wavelet, $B(\tau)$ is an even function, the second term is zero. Also the zero-mean random noise function $E(\tau)$ tends to be canceled out by the convolution with the symmetric wavelet function, so the fourth term tends to be zero. Thus, only the term with real peak $P(\tau)$ is left in (3.4). That is to say, as long as the background is slowly changing and locally monotonic in the Raman peak region with random noise, it will be automatically removed in calculating the CWT coefficients.

3.1.2.2 Peaks Extraction

If the mother wavelet is treated as a mask function, the integration in (3.1) is essentially a pattern matching, and the coefficients $\mathbf{C}$ are scores that measure how much the shapes of the signal matching to the mask function with different scales, at each Raman shift. For peaks extraction purpose, Mexican hat function is chosen as the mother wavelet because it has the shape of a peak. So the positions of Raman peaks tend to have high scores and backgrounds tend to have low scores. And at smaller scales, the scores measure the shape in narrow ranges; at bigger scales, the scores measure the peak shape in wider ranges. Fig. 3.2 shows the CWT coefficients of one Raman signal. The brightness of the figure represents the intensities of coefficients. Dimension b is the same with the dimension of Raman shifts. We

can see at peak positions, the corresponding CWT coefficients are high, and the coefficients are increasing as the increasing of scales.

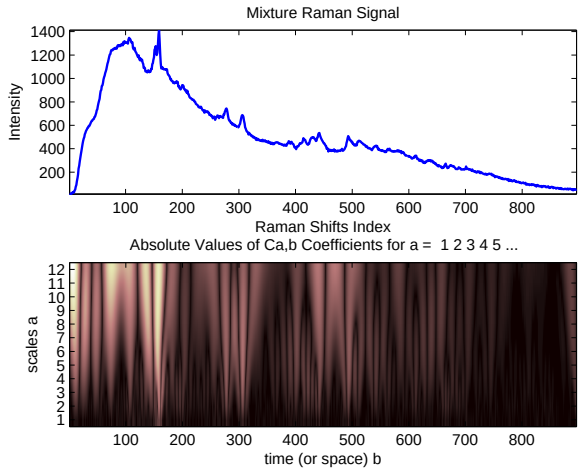


Figure 3.2: CWT coefficients of one Raman signal.

3.1.2.3 Why $Mean(\mathbf{C})$ Works

We want to illustrate the mean values of the CWT scores along different scales are approximately proportional to the heights of Raman peaks. Then based on the property of Raman spectrum mentioned in section 1.1.1: the height of one Raman peak in the mixture Raman signals is proportional to the concentration of the pure material, we can say $Mean(\mathbf{C})$ is proportional to concentration, which explains why $Mean(\mathbf{C})$ works in CWT-PLSR.

If we assume in certain Raman peak range $\tau \in R$, the height function of a Raman peak of certain nano-tag with standard concentration (100%) is $P(\tau)$, then for 90% concentration, the height function is $90\%P(\tau)$, and for 50% concentration, the height function is $50\%P(\tau)$. According to the definition of CWT in (3.1)

and $Mean(\mathbf{C})$ in section 3.1.1, the ratio of $Mean(\mathbf{C})$ with 90% concentration and $Mean(\mathbf{C})$ with 50% concentration at the Raman peak is

$$\frac{\frac{1}{s}\sum_{a=1}^{s}\int_R 90\%P(\tau)\psi_{a,b}(\tau)d\tau}{\frac{1}{s}\sum_{a=1}^{s}\int_R 50\%P(\tau)\psi_{a,b}(\tau)d\tau} = \frac{90\%}{50\%}, \tag{3.5}$$

which is the ratio of their peak heights as well as the ratio of their mixing concentrations.

Fig. 3.3 shows $Mean(\mathbf{C})$ of different Raman signals. We can see all blue signals or black signals are overlapped (stable) in the lower figure; background parts become zero, peak part become high (background removed); and signals are more smooth (random noise removed). Also because the high intensity backgrounds are removed, the weak peaks can be used more efficiently. In the CWT coefficients (or $Mean(\mathbf{C})$),

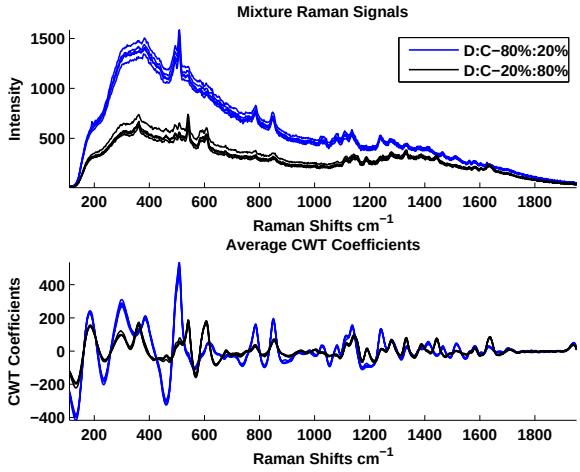


Figure 3.3: Average CWT coefficients along different scales. Signals with the same color are five duplicate Raman signals of one sample collected at different time.

there are negative values (valley points in the lower figure of Fig. 3.3) that locate near the Raman peaks. They are the convolution results between the Raman peaks and the negative part of the Mexican hat mask (Fig. 3.1). And the depthes of the valley

points are proportional to the heights of the Raman peaks, as well as the mixing concentrations. So they do not affect the PLSR results.

3.1.2.4 Robustness to Noisy Peaks

CWT is based on the shape information of one signal, weak Raman peaks and noisy peaks are not differentiable. But when consider many signals together with the mixing concentrations information, the heights of weak Raman peaks are highly correlated with mixing concentrations, but noisy peaks are randomly happened, there is no correlation. Then PLS will give low weights on noisy peaks. Also when there is noisy peaks in testing sample, since their weights are low, they will not contribute to latent variables and the prediction. So combining CWT with PLSR will reduce the influence of noisy peaks automatically.

3.2 Experiment

3.2.1 Methods Specification

To evaluate the effectiveness of CWT-PLSR for quantitative analysis of Raman spectrum, in this section, we compare it with the following methods: Ridge Regression (RR) [30]; Principle Component Regression (PCR) [33]; Orthonormalized PLS (O-PLS) [**?**]; PLS2 [42]; SIMPLS(SIM) [45]; linear programming baseline correction [57] based PLSR (P-PLS2 and P-SIM) and iteratively curve-fitting baseline correction [24] based PLSR (I-PLS2 and I-SIM). RR needs to add a parameter κ to $(\mathbf{X}^T\mathbf{X})^{-1}$ in the least square solution to solve the singularity problem: $\hat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X} + \kappa\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$, κ needs to be a small number and we can simply set it as 0.1. Similar to RR, OPLS also needs a parameter to remove the singularity problem, and it is also set as 0.1.

Table 3.1: Optimized components number K^*

Parameter	Data Set 1			Data Set 2			Data Set 3		
	D	A	AA	D	A	AA	D	A	AA
PCR	6	9	7	5	5	6	13	13	10
OPLS	1	2	2	2	2	1	4	4	4
PLS2	8	9	4	5	5	4	10	10	10
SIM	4	9	4	4	4	4	10	10	8
P-PLS2	5	5	5	6	4	4	21	21	19
P-SIM	5	5	4	10	4	4	21	21	11
I-PLS2	3	3	4	4	4	4	24	24	19
I-SIM	3	3	4	6	6	4	24	24	19
CWTPLS2	3	3	3	3	3	3	8	8	8
CWTSIM	3	3	3	3	3	3	8	8	8

Table 3.2: Optimized parameters

Parameter	Data Set 1			Data Set 2			Data Set 3		
	D	A	AA	D	A	AA	D	A	AA
P-PLS2-p	5	5	5	6	11	7	7	7	7
P-SIM-p	10	10	10	11	11	11	10	10	10
I-PLS2-p	8	9	8	7	7	7	10	9	9
I-SIM-p	8	8	12	9	9	7	10	10	9
PLS2-s	15	15	14	30	30	28	10	10	9
SIM-s	15	15	14	30	30	29	8	8	8

To maximize the performance of all the latent variable regression methods, the component number K needs to be optimized for each data set and each cross-validation method. Every possible component number is tested, the one giving the lowest RMSE returns as the optimized one K^*. The results are in Table 3.1.

Beside optimized component number, other parameters also need to be optimized. For baseline correction based PLSR methods (P-PLS2, P-SIM, I-PLS2 and I-SIM), the polynomial curve-fitting order p needs to be decided. For CWT-PLS (PLS2 and SIM), the optimized wavelet scale numbers s needs to be found (PLS2-s and SIM-s). All of the optimized parameters are found by the same way as K^*, and summarized in Table 3.2.

3.2.2 Results and Discussion

Table 3.3: RMSE for each method, on three data sets and three cross-validation methods (CVD, CVA and CVAA). The bold face represents the best result.

Methods	Data Set One			Data Set Two			Data Set Three		
	CVD	CVA	CVAA	CVD	CVA	CVAA	CVD	CVA	CVAA
RR	2.94	1.61	2.34	4.36	4.23	4.41	4.46	4.84	5.85
PCR	2.81	1.63	2.25	4.14	4.04	4.12	4.11	4.50	5.08
OPLS	2.94	1.54	2.08	4.20	4.09	4.41	4.46	4.84	5.85
PLS2	2.93	1.59	2.20	4.22	4.11	3.86	3.90	4.33	4.86
SIM	2.72	1.61	2.27	4.13	4.03	3.90	4.15	4.53	5.34
P-PLS2	2.78	1.47	1.56	4.27	3.99	3.74	2.89	2.75	3.05
P-SIM	2.84	1.61	2.21	4.52	4.29	4.30	2.99	2.84	3.24
I-PLS2	2.70	1.50	1.92	3.99	3.75	3.86	2.86	2.72	3.09
I-SIM	2.66	1.51	1.88	4.56	4.42	4.53	2.86	2.73	3.10
CWTPLS2	**2.56**	**1.43**	**1.45**	**3.28**	**3.16**	**3.13**	2.72	2.63	2.66
CWTSIM	2.65	1.47	1.46	3.39	3.27	3.24	**2.67**	**2.58**	**2.59**

In Table 3.3, we show the results of different regression methods, using three data sets and three cross-validation methods. Most of the results of latent variable regression methods (PLS2, SIM, PCR, OPLS) are better than those of RR, which means the latent space calibration model is better than the full spectrum calibration model. Baseline correction based PLSR (P-PLS2, P-SIM, I-PLS2 and I-SIM) are usually better than PLSR (PLS2 and SIM), since they reduce the influence of the instable backgrounds. But they are not always better because locally the baselines may not be perfect backgrounds, and the hard cut of these baselines will lose Raman peak information. The results of CWT-PLS methods (CWTPLS2 and CWTSIM) are always better than other methods, and the optimized component numbers of them are lower and more stable than other methods. These because CWT-PLSR methods more effectively reduce the instable background and random noises, and extract more useful peak information.

The results of CVA and CVAA tend to be better than CVD, since the average testing signals will reduce the influence of instable backgrounds. And CVA is better than CVAA, since CVA has more training samples than CVAA. The results of data set 1 is the best among three data sets. Because first, the mixing concentrations of two nano-tags are related (summation equals to 100%), if one concentration can be estimated, the other is easy to get; second, data set 1 has one nano-tag (DTTC) dominating the Florence background and this background tends to linearly related to its concentration, so the instable backgrounds in data set 1 do not affect the estimation of the concentration of DTTC; third, the Raman peaks of two nano-tags in data set 1 have less overlaps than the other two data sets, which also decreases the difficulty of prediction. In data set 2 the mixing concentrations of two nano-tags are also related, but there is no dominating Florence background, so the instable mixing background will affect the prediction more than data set 1. Plus, there are more overlaps between Raman peaks of two nano-tags, so the results of data set 2 are worse than data set 1. In data set 3, one nano-tag (HITC) also has a dominated background, but its results are the worst. This because it contains four nano-tags, are there are more overlaps between the Raman peaks in data set 3, so it increases the difficulty of prediction. Also the mixing concentrations of four nano-tags are not related, then the dominated background will affect the prediction of the other three. So baseline correction based methods and CWT-PLS improve most in data set 3.

If we think the projections in (2.10) as linear combinations of the intensities of Raman spectra at different Raman shifts, elements of each column of $\mathbf{W}$ are actually the weights showing how important each Raman shift in the combination is for the representing of Raman spectra $\mathbf{X}$ and relating to concentrations $\mathbf{Y}$. In order to show how efficient the algorithm is using the peak information, in Fig. 3.4, we compare the projection direction $\mathbf{w}_i$ of the first four components (latent variables) of PLS2 and

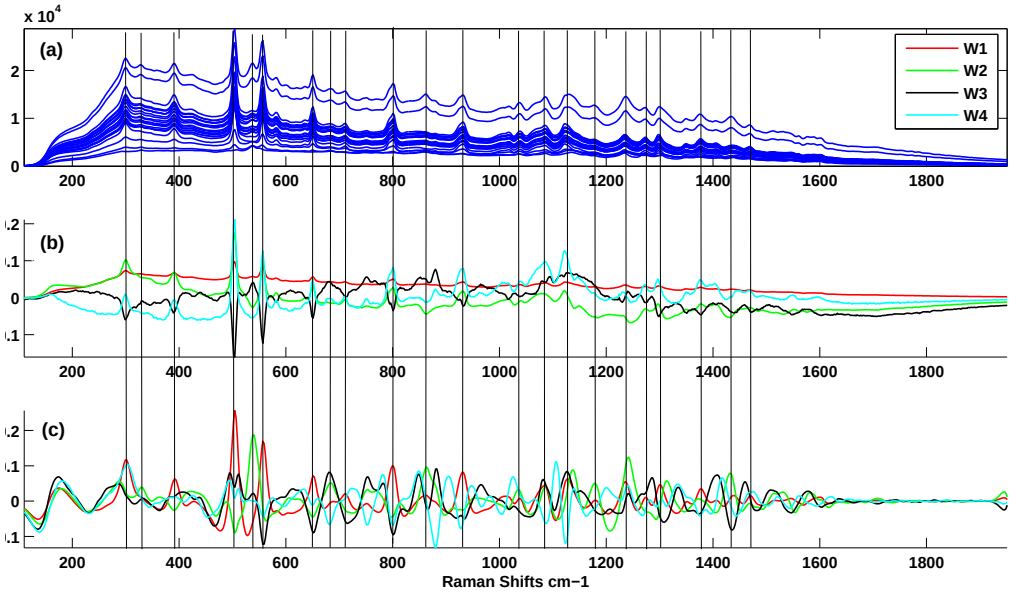


Figure 3.4: Raman peaks usage demonstration: (a) average Raman signals in data set 3; (b) first four projection directions got from PLS2; (c) first four projection directions got from CWT-PLS2. Vertical lines show the positions of selected Raman peaks

CWT-PLS2 on data set 3. We can see CWT-PLS2 gives bigger weights on peaks, which means it use more Raman peak information to do regression than traditional PLS2. And almost every big weights in the first (red) and second (green) component are corresponding to Raman peaks. This illustrates that CWT-PLS2 is mainly based on Raman peaks to predict, which makes it more reliable and robust than PLS2.

3.3 Conclusion and Future Work

We give a new CWT-PLSR algorithm for quantitative analysis of Raman spectrum. It treats the average CWT coefficients along different scales as the estimation of Raman spectrum (peaks) and combines mixing concentrations to do PLS. This method can effectively reduce random noise and avoid the influences of instable backgrounds and noisy peaks. So it can omit the preprocessing works, such as smoothing,

de-noising and baseline correction, and is more convenient. The experimental results illustrate its prediction accuracy outperforms the direct PLSR and some of the baseline correction methods based PLSR, and is a robust and efficient method for quantitative analysis of Raman spectrum.

In CWT-PLSR, we use $Mean(C)$ as the estimation of Raman peaks without explaining the reason why it is a robust and effective way. In the future work, we will deeply analysis the reason why it is robust and accurate, and show the comparing results of other estimating methods. Also we will compare it with more baseline correction methods to show the robust performance of the multi-scale essence of CWT-PLSR; and with DWT decomposition based multi-scale regression methods to show the effective performance of the peak extraction essence of CWT-PLSR.

CHAPTER 4

CONTINUUM REGRESSION

PCR maximizes variance of $\mathbf{t}_i$ (noted as $var(\mathbf{t}_i)$); RRR maximizes the correlation between $\mathbf{t}_i$ and $\mathbf{Y}$ (noted as $corr(\mathbf{t}_i, \mathbf{Y})$); PLS maximizes covariance between $\mathbf{t}_i$ and $\mathbf{Y}$ (noted as $cov(\mathbf{t}_i, \mathbf{Y})$). So the latent variables $\mathbf{T}$ of PCR and RRR best represent $\mathbf{X}$ and best predict $\mathbf{Y}$. PLS balances the portion of the two tasks (representation and prediction) with equal weight.

Continuum regression methods balance the representation and prediction in a flexible way, and the objective functions are combinations of $var(\mathbf{t}_i)$ and $corr(\mathbf{t}_i, \mathbf{Y})$, balanced by weight parameters. When the parameters are continuously adjusted, the portions of two tasks in the objective functions are also continuously adjusted. In this paper, we propose a new tactics to combine two tasks and embrace PCR, RRR and PLS as three special cases. It beats simple continuum regression (SCR) [58] who only contains PLS and RRR, and PCovR [59] who only contains PCR and RRR. And since the algorithm is based on NIPALS algorithm, it is easy to implemented.

4.1 Continuum Regression Methods

4.1.1 PCovR

PCovR [59] combines two criterions (J_{PCR1} and J_{RRR}) with a weight parameter α:

$$\text{obj. } \min_{\mathbf{T}} \alpha||\mathbf{X} - \mathbf{T}\mathbf{P}_x||^2 + (1-\alpha)||\mathbf{Y} - \mathbf{T}\mathbf{P}_y||^2. \tag{4.1}$$

The lengths of the projecting vectors $\{\mathbf{w}_i\}_{i=1}^K$ do not affect the objective function (proved in Appendix A.2 and Appendix A.3) so the constraint $\mathbf{T}^T\mathbf{T} = \mathbf{I}_K$ does not

lose generality. And the solution of $\mathbf{T}$ can be calculated as the first K eigenvectors of $\alpha\mathbf{X}\mathbf{X}^T + (1-\alpha)\mathbf{Y}\mathbf{Y}^T$, and $\mathbf{W}$ and $\mathbf{P}_y$ can be computed as: $\mathbf{W} = \mathbf{X}^{-}\mathbf{T}$ and $\mathbf{P}_y = \mathbf{W}^T\mathbf{X}^T\mathbf{Y}$. Then the PCovR regression coefficients are $\mathbf{B} = \mathbf{W}\mathbf{P}_y$. When $\alpha = 0$, PCovR equals to RRR or OPLS; when $\alpha = 1$, it equals to PCR; when $0 < \alpha < 1$, it is a compromise between two tasks. But de Jong [59] demonstrated PCovR doesn't embrace PLS.

4.1.2 Simple Continuum Regression (SCR)

SCR [58, 60] (or called canonical ridge analysis in [61]) is between RRR (or OPLS) and PLS. And the objective function is:

$$\text{obj.} \quad \max_{\mathbf{w}_i} \frac{||cov(\mathbf{X}\mathbf{w}_i, \mathbf{Y})||^2}{(1-\gamma)||\mathbf{X}\mathbf{w}_i||^2 + \gamma||\mathbf{w}_i||^2}, \tag{4.2}$$

which can be written as:

$$\begin{aligned} \text{obj.} \quad & \max_{\mathbf{w}_i} ||cov(\mathbf{X}\mathbf{w}_i, \mathbf{Y})||^2 \\ \text{s.t.} \quad & (1-\gamma)||\mathbf{X}\mathbf{w}_i||^2 + \gamma||\mathbf{w}_i||^2 = 1, \end{aligned} \tag{4.3}$$

$||cov(\mathbf{X}\mathbf{w}_i, \mathbf{Y})||^2 = \mathbf{w}_i^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_i$. When $\gamma = 0$, it equals to RRR, when $\gamma = 1$, it equals to PLS. $\mathbf{w}_i$ is solved as the first eigenvector of $[(1-\gamma)\mathbf{X}_i^T\mathbf{X}_i + \gamma\mathbf{I}]^{-1}(\mathbf{X}_i^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}_i)$, and $\mathbf{X}_i$ is the ith residual matrix of $\mathbf{X}$ calculated by a deflation process similar to Algorithm 1.

4.2 New Continuum Regression Method

de Jong [59] presented the limitation of PCovR is from the summation of two criterions in the objective function. In the summation formulation Equation (4.1), only one big criterion can make the total criterion big and compensate the other small one. So the solution may focus on one criterion. In order to force both criterions to

be big, multiplication of two may be superior to the summation. And the limitation of SCR is it can not achieve PCR.

4.2.1 Formulation

Considering the two limitations listed above, we design the objective function as:

$$\begin{aligned} \text{obj. } & \max_{\mathbf{w}_i}[\mathbf{w}_i^T(\mathbf{X}^T\mathbf{X})^{1-\alpha}\mathbf{w}_i]^{-1}(\mathbf{w}_i^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_i) \\ \text{s.t. } & \mathbf{t}_i^T\mathbf{t}_j = 0, i = 1, ..., K, j = 1, ..., i-1, \end{aligned} \tag{4.4}$$

When $\alpha = 0$, Equation (4.4) becomes Equation (2.20), and it is RRR; when $\alpha = 1$, Equation (4.4) equals to Equation (2.22), and it is PLS; when $\alpha = \infty$, the portion of $\mathbf{w}_i^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_i$ can be ignored, and it becomes PCR. Since the length of $\mathbf{w}_i$ does not affect the value of the objective function, we can always change the length of $\mathbf{w}_i$ to make $\mathbf{w}_i^T(\mathbf{X}^T\mathbf{X})^{1-\alpha}\mathbf{w}_i = 1$, so the objective function (4.4) can be written as:

$$\begin{aligned} \text{obj. } & \max_{\mathbf{w}_i} \mathbf{w}_i^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_i \\ \text{s.t. } & \mathbf{w}_i^T(\mathbf{X}^T\mathbf{X})^{1-\alpha}\mathbf{w}_i = 1 \text{ and } \mathbf{t}_i^T\mathbf{t}_j = 0. \end{aligned} \tag{4.5}$$

4.2.2 Algorithm

When $i = 1$, $\mathbf{w}_1$ is solved as the first eigenvector of $(\mathbf{X}^T\mathbf{X})^{(\alpha-1)}(\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})$. The calculation of $(\mathbf{X}^T\mathbf{X})^{(\alpha-1)}$ can be done by the eigen-decomposition of $\mathbf{X}^T\mathbf{X}$ as $\mathbf{X}^T\mathbf{X} = \mathbf{U}\Sigma\mathbf{U}^T$ first, Σ is diagonal matrix with diagonal elements are eigenvalues. Then $(\mathbf{X}^T\mathbf{X})^{(\alpha-1)} = \mathbf{U}\Sigma^{(\alpha-1)}\mathbf{U}^T$. When $i = 2, ..., K$, $\mathbf{w}_i$ is calculated as the first eigenvector of $(\mathbf{X}_i^T\mathbf{X}_i)^{(\alpha-1)}(\mathbf{X}_i^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}_i)$, with $\mathbf{X}_i$ is the ith residual matrix of $\mathbf{X}$ calculated in Algorithm 4, in which $eig(\mathbf{X}_i^T\mathbf{X}_i)$ is the eigen-decomposition of $\mathbf{X}_i^T\mathbf{X}_i$, and return all eigenvectors $\mathbf{U}$ and eigenvalues Σ. Function $powereig(\mathbf{H}(\mathbf{X}_i^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}_i))$ is used and introduced in Algorithm 1, which is to calculate the first eigenvector of the

Algorithm 4 New Continuum Regression Algorithm

Input: $\mathbf{X}$, $\mathbf{Y}$, K

Output: $\mathbf{B}$

1: **for** $i = 1$ to K **do**

2: $[\mathbf{U}, \Sigma] = eig(\mathbf{X}_i^T \mathbf{X}_i)$; % eigen-decomposition.

3: $\mathbf{H} = \mathbf{U}\Sigma^{(\alpha-1)}\mathbf{U}^T$; % calculate $(\mathbf{X}_i^T \mathbf{X}_i)^{(\alpha-1)}$.

4: $\mathbf{w}_i = powereig(\mathbf{H}(\mathbf{X}_i^T \mathbf{Y}\mathbf{Y}^T \mathbf{X}_i))$; % power method

5: $\mathbf{t}_i = \mathbf{X}_i \mathbf{w}_i$;% score vector of $\mathbf{X}_i$.

6: $\mathbf{p}_i = \mathbf{X}_i^T \mathbf{t}_i / (\mathbf{t}_i^T \mathbf{t}_i)$; % loading vector of $\mathbf{X}_i$.

7: $\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i \mathbf{p}_i^T$; % residual matrix $\mathbf{X}_{i+1}$.

8: **end for**

9: Store $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_K]; \mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_K]$;

10: $\mathbf{C} = \mathbf{Y}^T \mathbf{T}(\mathbf{T}^T \mathbf{T})^{-1}$;

11: $\mathbf{B} = \mathbf{P}(\mathbf{P}^T \mathbf{P})^{-1} \mathbf{C}$;

matrix $(\mathbf{X}_i^T \mathbf{X}_i)^{(\alpha-1)}(\mathbf{X}_i^T \mathbf{Y}\mathbf{Y}^T \mathbf{X}_i)$ by using power method [62]. Steps 5-7 are deflation process on matrix $\mathbf{X}$ to satisfy the constraint $\mathbf{t}_i^T \mathbf{t}_j = 0$.

4.3 Experiment

4.3.1 Methods Specification

Before evaluate each method on three data sets with three cross-validation methods, all Raman signals are preprocessed by baseline correction method (e.g. iteratively curve-fitting baseline correction [24]), to remove the instable background intensity and extract the Raman spectrum.

To evaluate the effectiveness of the new continuum regression method (NCR) for quantitative analysis of Raman spectrum, in this section, we compare it with Ridge

Table 4.1: Optimized $pOrder^*$

CR	Data Set 1			Data Set 2			Data Set 3		
	D	A	AA	D	A	AA	D	A	AA
RR	10	6	6	3	3	3	6	8	4
PCR	6	6	6	3	3	3	6	6	4
RRR	6	8	6	3	3	3	7	7	4
PLS2	6	6	8	3	3	3	6	8	4
PCovR	6	6	5	3	4	3	6	8	4
S-CR	6	6	8	3	3	3	6	8	4
N-CR	10	10	6	3	3	3	8	8	4

Table 4.2: Optimized α^*

CR	Data Set 1			Data Set 2			Data Set 3		
	D	A	AA	D	A	AA	D	A	AA
PCovR	1	1	0.05	1	0.05	1	0.05	0.05	1
S-CR	1	1	1	0	0	0	1	0.4	0.65
N-CR	0.1	0.1	0.25	1	1	1	0.25	0.25	8

Table 4.3: Optimized K^*

CR	Data Set 1			Data Set 2			Data Set 3		
	D	A	AA	D	A	AA	D	A	AA
PCR	8	8	8	3	3	3	30	30	20
RRR	21	5	9	9	9	2	30	30	20
PLS2	4	3	3	3	3	3	20	28	17
PCovR	8	8	4	3	6	3	15	23	20
S-CR	4	3	3	6	6	3	20	13	15
N-CR	5	5	3	3	3	3	23	27	20

Regression (RR) [30], three latent variable regression methods (PCR [33], RRR [63], PLS2 [42]), and other two continuum regression methods (PCovR [59] and SCR [58]), testing on three Raman spectrum data sets and using three cross validation methods.

To maximize the performance of all methods, several parameters needs to be optimized. First is the polynomial curve-fitting order *pOrder* of the baseline correction method for each data set and each cross-validation method. Different orders

(from 3 to 10) are tested, the one giving the lowest RMSE returns as the optimized $pOrder^*$. The results are in Table 4.1.

Second, all the continuum regression methods (PCovR, SCR, NCR) have a weight parameter α to balance the portions of prediction and representation in the model. In our experiment, we test $\{0, 0.05, 0.1, 0.15, ..., 0.95, 1\}$ for PCovR and SCR; $\{0, 0.05, 0.1, 0.15, ..., 0.95, 1, 2, 4, 6, 8, 10\}$ for NCR, and the one giving the lowest RMSE returns as the optimized α^*. The results are in Table 4.2.

Third, all the continuum regression methods (PCovR, SCR, NCR) and latent variable regression methods (PCR, RRR, PLS) have to optimize the component number K. We test different numbers (from 1 to 30) for each data set and each cross-validation method, the one giving the lowest RMSE returns as the optimized K^*. The results are in Table 4.3.

4.3.2 Results and Discussion

Table 4.4: RMSE, $pOrder^*$ and K^* for each method, on three data sets and three cross-validation methods (CVD, CVA and CVAA). The results are shown as: RMSE ($pOrder^*$, K^*).

Methods	Data Set One			Data Set Two			Data Set Three		
	CVD	CVA	CVAA	CVD	CVA	CVAA	CVD	CVA	CVAA
RR	2.75	1.63	1.77	4.37	4.22	4.28	5.75	5.35	5.87
PCR	2.50	1.39	1.73	3.24	3.13	3.14	6.01	5.84	5.79
RRR	2.65	1.50	1.77	3.51	3.38	4.02	6.36	6.14	5.86
PLS2	2.48	1.37	1.68	3.18	3.07	**3.06**	5.65	5.35	5.85
PCovR	2.50	1.39	**1.50**	3.24	3.00	3.14	5.83	5.60	5.79
SCR	2.48	1.37	1.68	**3.07**	**2.92**	**3.06**	5.65	5.33	5.84
NCR	**2.38**	**1.25**	1.64	3.18	3.07	**3.06**	**5.58**	**5.23**	**5.75**

In Table 4.4, we show RMSE of different regression methods, using three data sets and three cross-validation methods, and the bold face represents the best result.

The results of CVA and CVAA tend to be better than CVD, since the average testing signals will reduce the influence of instable backgrounds. And CVA is better than CVAA, since CVA has more training samples than CVAA.

The results of data set 1 is the best among three data sets. Because first, the mixing concentrations of two nano-tags are related (summation equals to 100%), if one concentration can be estimated, the other is easy to get; second, the Raman peaks of two nano-tags in data set 1 have less overlaps than the other two data sets, which also decreases the difficulty of prediction. In data set 2 the mixing concentrations of two nano-tags are also related, but there are more overlaps between Raman peaks of two nano-tags, so the results of data set 2 are worse than data set 1. The results of data set 3 are the worst, because it contains four nano-tags, and there are more overlaps between the Raman peaks in data set 3, so it increases the difficulty of prediction.

Most of the results of latent variable methods (PLS2, PCR, RRR) are better than those of RR, because RR is more easily to get over-fitting to the training sets and latent variable methods do regression in a subspace, which avoid some random noise. Within latent variable methods, RRR is worse than PCR in all situation, because RRR doesn't consider any representation of $\mathbf{X}$, which makes it not rubost; and PLS is better than PCR in most case, since PLS is the middle point between PCR and RRR, and its $\mathbf{T}$ balances both prediction of $\mathbf{Y}$ and representation of $\mathbf{X}$. PCovR is always not worse than PCR, since its two extreme models are RRR ($\alpha = 0$) and PCR ($\alpha = 1$), but it can be worse than PLS since it doesn't include PLS. SCR is not worse than PLS, since its two extreme models are RRR ($\alpha = 0$) and PLS ($\alpha = 1$). Our algorithm (NCR) is also not worse than PLS, and it has more best results than the other two CR methods (6 out of 9), because it has a bigger range of model than SCR (from RRR to PCR), and contains PLS, which beats PCovR.

RMSE represents the average prediction error for all the pure components (dyes), in Table 4.5, we also show the prediction error of each pure component (dye) in data set 3 (In data set 1 and 2, the prediction error of each dye are the same as their RMSE). NCR also has the most best result (5 out of 12). So it is a robust quantitative analysis method for Raman spectrum.

Table 4.5: Comparison of RMSE of each dye, for each method, by three cross-validation methods (CVD, CVA and CVAA) on data sets 3.

CV	Dyes	Methods						
		RR	PCR	RRR	PLS	PCovR	SCR	**NCR**
CVD	DOTC	3.57	3.80	**3.22**	3.47	3.86	3.47	3.57
	DTTC	6.37	6.13	5.62	6.18	6.16	6.17	**5.39**
	HITC	**4.99**	5.74	6.59	5.12	5.54	5.13	5.66
	IR140	7.36	7.73	9.79	7.16	7.25	7.16	**7.13**
CVA	DOTC	3.37	3.74	**3.11**	3.38	3.47	3.39	3.37
	DTTC	5.34	6.01	5.44	5.32	**5.02**	5.37	5.24
	HITC	5.42	5.52	6.40	5.47	6.29	5.39	**5.31**
	IR140	6.74	7.47	9.52	6.71	6.98	6.75	**6.53**
CVAA	DOTC	**4.09**	4.48	4.19	4.37	4.48	4.56	4.34
	DTTC	5.18	**5.08**	5.58	5.17	**5.08**	5.18	5.10
	HITC	4.99	4.86	5.22	4.94	4.84	5.32	**4.80**
	IR140	8.33	8.03	**7.94**	8.17	8.03	8.07	8.02

4.4 Conclusion and Future Work

In this chapter, we present a new continuum regression method to do the quantitative analysis of Raman spectrum. It uses a continuous weight parameter to adjust the portions of representing X and predicting Y in the objective function, and is achieved by NIPALS algorithm. Since it contains RRR, PCR and PLS as its special cases, its performance beats the other two CR methods (PCovR and SCR).

In the future works, we will explain why when $\alpha = \infty$, NCR equals to PCR and why the NIPALS based algorithm can effectively solve the objective function in (4.5). Also we will compare our NCR with more CR methods to show its performance. Besides, to determine the tuning parameter α, a less time consuming and more effective Bayesian nonparametrics method need to be designed.

CHAPTER 5

PROBABILISTIC PARTIAL LEAST SQUARES REGRESSION

5.1 Probabilistic Models

In this section, we will introduce latent variable methods PCA and CCA, and compare their probabilistic models, which are the foundations of our presented models. Latent variable methods usually work on the centralized data set, so $\mathbf{X}$ and $\mathbf{Y}$ denote zero mean matrixes. $\mathbf{W} = \{\mathbf{w}_i\}_{i=1}^{K}$ denotes K projecting vectors of $\mathbf{X}$ and $(\mathbf{W}_x, \mathbf{W}_y) = \{(\mathbf{w}_{xi}, \mathbf{w}_{yi})\}_{i=1}^{K}$ denotes K pairs of projecting vectors of or $\mathbf{X}$ and $\mathbf{Y}$. Index $i, j \in [1, K]$, and $\forall i \neq j$.

5.1.1 PCA and PPCA

The goal of PCA is to reduce the dimensionality of a data set that contains a large number of interrelated variables, and remain as much as possible of the variation [**?**]. It is achieved by transforming $\mathbf{X}$ to a new sets of uncorrelated variables, which are ordered so that the first few remain most variation of $\mathbf{X}$. The derivation of sample PCA is given sample sets $\mathbf{X}$, to find K projecting directions (PCA loading vectors) $\{\mathbf{w}_i\}_{i=1}^{K}$ to project $\mathbf{X}$, in which $K \ll D_x$, and get K sets of uncorrelated projections (principle components, or scores) that span the biggest variances of $\mathbf{X}$:

$$\text{obj. } \max_{\mathbf{w}_i} var(\mathbf{X}\mathbf{w}_i); \text{ s.t. } \mathbf{w}_i^T\mathbf{w}_i = 1; \mathbf{w}_i^T\mathbf{X}^T\mathbf{X}\mathbf{w}_j = 0, \tag{5.1}$$

$var(\mathbf{X}\mathbf{w}_i) = \mathbf{w}_i^T\mathbf{X}^T\mathbf{X}\mathbf{w}_i$ is the sample variance of the ith PCA component. In order to achieve the maximum, lengths of the projecting vectors $\{\mathbf{w}_i\}_{i=1}^{K}$ are fixed to 1. The solution of all the projecting directions $\mathbf{W}$ are the first K eigenvectors of covariance

matrix $\mathbf{X}^T\mathbf{X}$ corresponding to the K biggest eigenvalues. $\mathbf{w}_i^T\mathbf{X}^T\mathbf{X}\mathbf{w}_i$ are called the ith principle variance.

Tipping and Bishop [64] present a Probabilistic PCA (PPCA) model to illustrate PCA from a probabilistic point of view, in which an observation $\mathbf{x}$ is treated as random variables governed by low dimensional latent variables:

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon},$$

with columns of matrix $\mathbf{W}$ are scaled loadings of the principle components, whose lengthes are different from the loadings $\mathbf{W}$ in (5.1). $\mathbf{z}$ is a K dimensional vector of random variables representing the normalized principle components, defined as an isotropic Gaussian with unit variance: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\boldsymbol{\mu}$ represents the center of data. Residual part $\boldsymbol{\epsilon}$ describes the random noise outside the principle variances, which has the same effect on each variable, so it is assumed to be isotropic Gaussian: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. The conditional distribution of $\mathbf{x}$ given $\mathbf{z}$ and the marginal distribution of $\mathbf{x}$ are also Gaussian:

$$\mathbf{x}|\mathbf{z} \sim \mathcal{N}(\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I}) \tag{5.2}$$

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}), \tag{5.3}$$

with (5.3) is calculated from the properties of Gaussian distribution described in Appendix B.1.

5.1.2 CCA and PCCA

The purpose of CCA [49] is to find the latent relationship between two or more groups of variables from data sets. When consider two groups, CCA is achieved by finding K pairs of projecting directions (CCA loading vectors) $\{(\mathbf{w}_{xi}, \mathbf{w}_{yi})\}_{i=1}^K$ for $\mathbf{X}$ and $\mathbf{Y}$ respectively, and get K pairs of projections (canonical components) that

correlate most and each component within each set is only correlated with a single component in the other set:

$$\begin{aligned} \text{obj.} \quad & \max_{\mathbf{w}_{xi},\mathbf{w}_{yi}} corr(\mathbf{X}\mathbf{w}_{xi}, \mathbf{Y}\mathbf{w}_{yi}) \\ \text{s.t.} \quad & \mathbf{w}_{xi}^T\mathbf{X}^T\mathbf{X}\mathbf{w}_{xj} = 0;\ \mathbf{w}_{yi}^T\mathbf{Y}^T\mathbf{Y}\mathbf{w}_{yj} = 0 \\ & \mathbf{w}_{xi}^T\mathbf{X}^T\mathbf{Y}\mathbf{w}_{yj} = 0;\ \mathbf{w}_{xj}^T\mathbf{X}^T\mathbf{Y}\mathbf{w}_{yi} = 0, \end{aligned} \tag{5.4}$$

in which $corr(\mathbf{X}\mathbf{w}_{xi}, \mathbf{Y}\mathbf{w}_{yi}) = \frac{\mathbf{w}_{xi}^T\mathbf{X}^T\mathbf{Y}\mathbf{w}_{yi}}{\sqrt{var(\mathbf{X}\mathbf{w}_{xi})var(\mathbf{Y}\mathbf{w}_{yi})}}$ represents ith canonical correlation between two components, and for $i = 1 \dots K$, they are decreasing ordered. Without affecting the result, $var(\mathbf{X}\mathbf{w}_{xi})$ and $var(\mathbf{Y}\mathbf{w}_{yi})$ in (5.4) are set to 1, getting the solution of $\mathbf{W}_x$ and $\mathbf{W}_y$ as the first K eigenvectors of matrix $(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})(\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{Y}^T\mathbf{X})$ and $(\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{Y}^T\mathbf{X})(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})$ corresponding to the K biggest eigenvalues [49].

Bach and Jordan [65] give a Probabilistic CCA (PCCA) model that relates two sets of variables with a common set of latent variables under two Gaussian distributions:

$$\mathbf{x}|\mathbf{z} \sim \mathcal{N}(\mathbf{W}_x\mathbf{z} + \boldsymbol{\mu}_x, \boldsymbol{\Psi}_x) \tag{5.5}$$

$$\mathbf{y}|\mathbf{z} \sim \mathcal{N}(\mathbf{W}_y\mathbf{z} + \boldsymbol{\mu}_y, \boldsymbol{\Psi}_y), \tag{5.6}$$

with $\mathbf{z}$ follows the zero mean unit covariance Gaussian and elements of $\mathbf{z}$ are normalized canonical components describing the normalized common variances. $\mathbf{W}_x$ and $\mathbf{W}_y$ are scaled canonical correlation directions (in $\mathbf{X}$ and $\mathbf{Y}$ space), projected on which, two data sets are related most. $\mathbf{W}_x\mathbf{z}$ and $\mathbf{W}_y\mathbf{z}$ span the canonical correlation subspaces in $\mathbf{X}$ and $\mathbf{Y}$ space. $\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y$ model the unique variances on each dimension of $\mathbf{X}$ and $\mathbf{Y}$, outside their canonical correlation subspaces.

5.2 Probabilistic PLS (PPLS)

5.2.1 PPLS Model

Based on the comparison of three PLS methods (2.22) and (2.29), we can see all methods are essentially the same: all represent two group of variables $\mathbf{x}$ and $\mathbf{y}$ with a few shared latent variables $\mathbf{z}$. So similar to PPCA and PCCA, the PPLS can be modeled as:

$$\mathbf{x} = \mathbf{W}_x\mathbf{z} + \boldsymbol{\mu}_x + \boldsymbol{\epsilon}_x \text{ and } \mathbf{y} = \mathbf{W}_y\mathbf{z} + \boldsymbol{\mu}_y + \boldsymbol{\epsilon}_y$$

For PLS-W2A, $\mathbf{W}_y$ are loadings of $\mathbf{y}$; and for two PLS regression models, they are just regression coefficients between normalized PLS components $\mathbf{z}$ and $\mathbf{y}$. Since PLS components are unrelated to each other, we have:

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{5.7}$$

All three PLS methods require $\mathbf{W}_x\mathbf{z}$ and $\mathbf{W}_y\mathbf{z}$ span the systematic (principle) variance of $\mathbf{x}$ and $\mathbf{y}$, the residual part $\boldsymbol{\epsilon}_x$ and $\boldsymbol{\epsilon}_y$ can be modeled as random noises, which have the same effect on each dimension:

$$\boldsymbol{\epsilon}_x \sim \mathcal{N}(\mathbf{0}, \sigma_x^2\mathbf{I}) \text{ and } \boldsymbol{\epsilon}_y \sim \mathcal{N}(\mathbf{0}, \sigma_y^2\mathbf{I}).$$

Then the Probabilistic PLS model can be summarized as:

$$\mathbf{x}|\mathbf{z} \sim \mathcal{N}(\mathbf{W}_x\mathbf{z} + \boldsymbol{\mu}_x, \sigma_x^2\mathbf{I}) \tag{5.8}$$

$$\mathbf{y}|\mathbf{z} \sim \mathcal{N}(\mathbf{W}_y\mathbf{z} + \boldsymbol{\mu}_y, \sigma_y^2\mathbf{I}). \tag{5.9}$$

From (5.8) and (5.9), the joint distribution of $\mathbf{x}$ and $\mathbf{y}$ given $\mathbf{z}$ is formed as:

$$\begin{aligned} &\mathbf{x}, \mathbf{y}|\mathbf{z} \sim \mathcal{N}(\mathbf{m}_{xy|z}, \mathbf{S}_{xy|z}); \text{ with} \\ &\mathbf{S}_{xy|z} = \begin{pmatrix} \sigma_x^2\mathbf{I} & 0 \\ 0 & \sigma_y^2\mathbf{I} \end{pmatrix}; \mathbf{m}_{xy|z} = \begin{pmatrix} \mathbf{W}_x\mathbf{z} + \boldsymbol{\mu}_x \\ \mathbf{W}_y\mathbf{z} + \boldsymbol{\mu}_y \end{pmatrix}. \end{aligned} \tag{5.10}$$

From (5.7) and (5.10), and using the properties in Appendix B.1, the conditional distribution of $\mathbf{z}$ is calculated as:

$$\begin{aligned} p(\mathbf{z}|\mathbf{x},\mathbf{y}) &= \mathcal{N}(\mathbf{m}_{z|xy}, \mathbf{S}_{z|xy}); \\ \text{with } \mathbf{S}_{z|xy} &= (\mathbf{I} + \mathbf{A}_u^T \mathbf{S}_{xy|z}^{-1} \mathbf{A}_u)^{-1} \\ \mathbf{m}_{z|xy} &= \mathbf{S}_{z|xy} \mathbf{A}_u^T \mathbf{S}_{xy|z}^{-1} (\mathbf{u} - \boldsymbol{\mu}_u) \\ \mathbf{u} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}, \boldsymbol{\mu}_u &= \begin{pmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{pmatrix}, \mathbf{A}_u = \begin{pmatrix} \mathbf{W}_x \\ \mathbf{W}_y \end{pmatrix}. \end{aligned} \tag{5.11}$$

and the marginal distribution of $\mathbf{x}$ and $\mathbf{y}$ is calculated as:

$$p(\mathbf{x},\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_u, \mathbf{S}_{xy|z} + \mathbf{A}_u \mathbf{A}_u^T)$$

5.2.2 PPLS Regression (PPLSR)

The finial goal of PLS regression is given a testing predictor $\mathbf{x}$ to predict its response $\hat{\mathbf{y}}$. Here we give an estimation method using PPLS model. From (5.7) and (5.8), we can get:

$$\begin{aligned} \mathbf{z}|\mathbf{x} &\sim \mathcal{N}(\mathbf{m}_{z|x}, \mathbf{S}_{z|x}) \\ \text{with } \mathbf{S}_{z|x} &= (\mathbf{I} + \sigma_x^{-2} \mathbf{W}_x^T \mathbf{W}_x)^{-1}; \\ \mathbf{m}_{z|x} &= \sigma_x^{-2} \mathbf{S}_{z|x} \mathbf{W}_x^T (\mathbf{x} - \boldsymbol{\mu}_x) \end{aligned} \tag{5.12}$$

And from (5.9) and (5.12), we get:

$$\begin{aligned} \mathbf{y}|\mathbf{x} &\sim \mathcal{N}(\mathbf{m}_{y|x}, \mathbf{S}_{y|x}) \\ \text{with } \mathbf{m}_{y|x} &= \mathbf{W}_y \mathbf{m}_{z|x} + \boldsymbol{\mu}_y; \\ \mathbf{S}_{y|x} &= \sigma_y^2 \mathbf{I} + \mathbf{W}_y \mathbf{S}_{z|x} \mathbf{W}_y^T \end{aligned} \tag{5.13}$$

The calculation of (5.12) and (5.13) can be referred to Appendix B.1. The mean value $\mathbf{m}_{y|x}$ of $\mathbf{y}$ given $\mathbf{x}$ can be used as the prediction of $\hat{\mathbf{y}}$. The regression coefficients are:

$$\mathbf{B} = \mathbf{W}_y (\mathbf{W}_x^T \mathbf{W}_x + \sigma_x^2 \mathbf{I})^{-1} \mathbf{W}_x^T.$$

5.2.3 EM Algorithm for PPLSR

It is complicated to directly estimate all parameters by maximizing the log likelihood function $\sum_{n=1}^{N} \ln p(\mathbf{x}_n, \mathbf{y}_n)$. Here we give an EM algorithm. In expectation (E) step, it builds the distribution of latent variables $p(\mathbf{z}_n|\mathbf{x}_n, \mathbf{y}_n; \hat{\Theta})$ with training data and previously estimated parameters according to (5.11), $\hat{\Theta}$ denotes all the estimated parameters. We note the conditional distribution (5.11) as $Q(\mathbf{z})$, which gives the distribution of $\mathbf{z}$. In maximization (M) step, we estimate the parameters by maximizing the log likelihood:

$$\max_{\Theta} \sum_{n=1}^{N} E_{\mathbf{z}_n|Q} \left[\ln p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{z}_n; \Theta)\right] \tag{5.14}$$

The subscripts $\mathbf{z}_n|Q$ indicates that the expectations are with respect to $\mathbf{z}_n$ drawn from distribution Q. Θ denotes the unknown parameters. The derivation of (5.14) is proved in Appendix B.2.1. Set derivative of the (5.14) with respect to all parameters to zero (the calculation is briefly summarized in Appendix B.2.2), for means of $\mathbf{x}$ and $\mathbf{y}$ we get:

$$\boldsymbol{\mu}_x = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n \text{ and } \boldsymbol{\mu}_y = \frac{1}{N} \sum_{n=1}^{N} \mathbf{y}_n \tag{5.15}$$

These two parameters only relate to original data, so we can get them without EM algorithm. The loading matrixes are updated as:

$$\begin{aligned} \hat{\mathbf{W}}_x &= \left[\sum_{n=1}^{N} (\mathbf{x}_n - \boldsymbol{\mu}_x) E[\mathbf{z}_n]^T\right] \left[\sum_{n=1}^{N} E[\mathbf{z}_n \mathbf{z}_n^T]\right]^{-1} \\ \hat{\mathbf{W}}_y &= \left[\sum_{n=1}^{N} (\mathbf{y}_n - \boldsymbol{\mu}_y) E[\mathbf{z}_n]^T\right] \left[\sum_{n=1}^{N} E[\mathbf{z}_n \mathbf{z}_n^T]\right]^{-1}, \end{aligned} \tag{5.16}$$

with $E[\mathbf{z}_n] = \mathbf{m}_{z|xy}(\mathbf{z}_n)$ and $E[\mathbf{z}_n\mathbf{z}_n^T] = \mathbf{S}_{z|xy} + E[\mathbf{z}_n]E[\mathbf{z}_n]^T$ are calculated in the E step. Finally the noise levels are updated as:

$$\begin{aligned}
\hat{\sigma}_x^2 = \frac{1}{D_x N}\sum_{n=1}^{N}\{&Tr(E[\mathbf{z}_n\mathbf{z}_n^T]\hat{\mathbf{W}}_x^T\hat{\mathbf{W}}_x) \\
&- 2E[\mathbf{z}_n]^T\hat{\mathbf{W}}_x^T(\mathbf{x}_n - \boldsymbol{\mu}_x) + \|\mathbf{x}_n - \boldsymbol{\mu}_x\|^2\} \\
\hat{\sigma}_y^2 = \frac{1}{D_y N}\sum_{n=1}^{N}\{&Tr(E[\mathbf{z}_n\mathbf{z}_n^T]\hat{\mathbf{W}}_y^T\hat{\mathbf{W}}_y) \\
&- 2E[\mathbf{z}_n]^T\hat{\mathbf{W}}_y^T(\mathbf{y}_n - \boldsymbol{\mu}_y) + \|\mathbf{y}_n - \boldsymbol{\mu}_y\|^2\}
\end{aligned} \tag{5.17}$$

Iteratively until estimated parameters are converge. The pseudo-code of the EM algorithm is summarized in Algorithm 5.

Theoretically, the conditional log likelihood would be the criterion to control the convergence, since it is monotone increasing and converge to a upper bound. But for high dimensional Gaussian distribution, the probability of single data point as well as the likelihood are zero. So in real case it can not be used to control the convergence. Here we use $\mathbf{m}_{z|xy}$, since it is a function of all parameters. Its converge means all parameters are converge.

5.3 Experiment

5.3.1 Results and Analysis

Using three data sets and three cross-validation methods, we test the prediction ability of different regression methods show the accuracies in Table 5.1, bold face are the best results. And the optimized K^* for each method are shown in Table 5.2. RR is worse than most latent variable regression methods because of the sparsity of the high dimensional spectra data. OPLS uses very small number of components since it is limited by $rank(\mathbf{Y})$, but the accuracy is acceptable. This efficiency is because of

Algorithm 5 EM Algorithm for PPLSR

Input: $\mathbf{X}$, $\mathbf{Y}$, K

Output: $\mathbf{B}$

1: $\boldsymbol{\mu}_x \leftarrow \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$; $\boldsymbol{\mu}_y \leftarrow \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n$;

2: $\mathbf{X} \leftarrow \mathbf{X} - \boldsymbol{\mu}_x$; $\mathbf{Y} \leftarrow \mathbf{Y} - \boldsymbol{\mu}_y$;

3: Initialize parameters: $\hat{\Theta} = [\hat{\sigma}_x, \hat{\sigma}_y, \hat{\mathbf{W}}_x, \hat{\mathbf{W}}_y]$

4: **while** $\mathbf{m}_{z|xy}(\mathbf{Z})$ is not converge **do**

5: Calculate $\mathbf{m}_{z|xy}(\mathbf{Z}), \mathbf{S}_{z|xy}(\mathbf{Z})$ for N samples as in (5.11):

$\mathbf{S}_{z|xy}(\mathbf{Z}) = (\hat{\sigma}_x^2\hat{\mathbf{W}}_x^T\hat{\mathbf{W}}_x + \hat{\sigma}_y^2\hat{\mathbf{W}}_y^T\hat{\mathbf{W}}_y + \mathbf{I})^{-1}$

$\mathbf{m}_{z|xy}(\mathbf{Z}) = \mathbf{S}_{z|xy}(\hat{\sigma}_x^{-2}\hat{\mathbf{W}}_x^T\hat{\mathbf{X}}^T + \hat{\sigma}_y^{-2}\hat{\mathbf{W}}_y^T\hat{\mathbf{Y}}^T)$

with $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$.

6: Calculate $\Sigma_{\mathbf{Z}} = \sum_{n=1}^{N} E[\mathbf{z}_n\mathbf{z}_n^T]$ as in (5.16):

$\Sigma_{\mathbf{Z}} = N\mathbf{S}_{z|xy}(\mathbf{Z}) + \mathbf{m}_{z|xy}(\mathbf{Z})\mathbf{m}_{z|xy}(\mathbf{Z})^T$

7: Estimate new $\hat{\mathbf{W}}_x$ and $\hat{\mathbf{W}}_y$ as in (5.16):

$\hat{\mathbf{W}}_x = \mathbf{X}^T\mathbf{m}_{z|xy}(\mathbf{Z})^T\Sigma_{\mathbf{Z}}^{-1}$; $\hat{\mathbf{W}}_y = \mathbf{Y}^T\mathbf{m}_{z|xy}(\mathbf{Z})^T\Sigma_{\mathbf{Z}}^{-1}$

8: Estimate new parameters $\hat{\sigma}_x$ and $\hat{\sigma}_y$ as in (5.17):

$\hat{\sigma}_x^2 = Tr\{\Sigma_{\mathbf{Z}}\hat{\mathbf{W}}_x^T\hat{\mathbf{W}}_x + \mathbf{X}\mathbf{X}^T - 2\mathbf{X}\hat{\mathbf{W}}_x\mathbf{m}_{z|xy}(\mathbf{Z})\}/D_xN$

$\hat{\sigma}_y^2 = Tr\{\Sigma_{\mathbf{Z}}\hat{\mathbf{W}}_y^T\hat{\mathbf{W}}_y + \mathbf{Y}\mathbf{Y}^T - 2\mathbf{Y}\hat{\mathbf{W}}_y\mathbf{m}_{z|xy}(\mathbf{Z})\}/D_yN$

9: **end while**

10: $\mathbf{B} = \mathbf{W}_y(\mathbf{W}_x^T\mathbf{W}_x + \sigma_x^2\mathbf{I})^{-1}\mathbf{W}_x^T$

the normalization on the variances of predictor components of OPLS, it removes the variance of $\mathbf{X}$ that is unrelated with prediction.

The results of PPLSR are not the best but similar to those of PLS2 and SIMPLS. This is because PPLSR model is an probabilistic view of PLSR methods, and essentially they are the same. The merit of PPLSR over PLSR is it models the obser-

Table 5.1: RMSE of different regression methods using different cross-validation methods.

Methods	Data Set One			Data Set Two			Data Set Three		
	CVD	CVA	CVAA	CVD	CVA	CVAA	CVD	CVA	CVAA
RR	2.94	1.61	2.34	4.36	4.23	4.41	8.91	9.67	11.69
PCR	2.81	1.63	2.25	4.14	4.04	4.12	8.22	8.99	10.15
OPLS	2.94	1.54	**2.08**	4.20	4.09	4.41	8.91	9.67	11.69
PLS2	2.93	1.59	2.20	4.22	4.11	**3.86**	**7.90**	**8.65**	**9.75**
SIM	**2.72**	1.61	2.27	4.13	4.03	3.90	8.30	9.06	10.68
PPLSR	2.75	**1.53**	2.24	**4.07**	**3.97**	4.17	8.21	8.89	10.00

Table 5.2: Optimized K^*

CR	Data Set 1			Data Set 2			Data Set 3		
	D	A	AA	D	A	AA	D	A	AA
PCR	6	9	7	5	5	6	13	13	10
OPLS	1	2	2	2	2	1	4	4	4
PLS2	8	9	4	5	5	4	10	10	10
SIMPLS	4	9	4	4	4	4	10	10	8
PPLSR	8	37	7	5	5	4	13	8	6

vations into systematic part and unrelated noise part with parameters, which provides a foundation for Bayesian models to avoid the over-fitting problem that PLSR can not avoid.

The unrelated noise of $\mathbf{X}$ and $\mathbf{Y}$ are governed by parameters σ_x and σ_y. They are related with the components number K. From Figure 5.1 and 5.2, we can see, the uncertainty of the model is decreasing corresponding with the increasing number of components it uses. The systematic part of $\mathbf{X}$ and $\mathbf{Y}$ are governed by parameters $\mathbf{W}_x$ and $\mathbf{W}_y$. Figure 5.3 shows the ability of the latent variables to represent $\mathbf{X}$. It plots $100\% \times norm(\mathbf{X} - \mathbf{m}_{z|xy}(\mathbf{Z})\mathbf{W}_x^T)/norm(X)$ under different component numbers. Figure 5.4 shows the ability of the latent variables to represent $\mathbf{Y}$. It plots by and $100\% \times norm(\mathbf{Y} - \mathbf{m}_{z|xy}(\mathbf{Z})\mathbf{W}_y^T)/norm(X)$ under different component numbers. $\mathbf{m}_{z|xy}(\mathbf{Z}) \in R^{N\times K}$ are the latent variables of all samples calculated in step

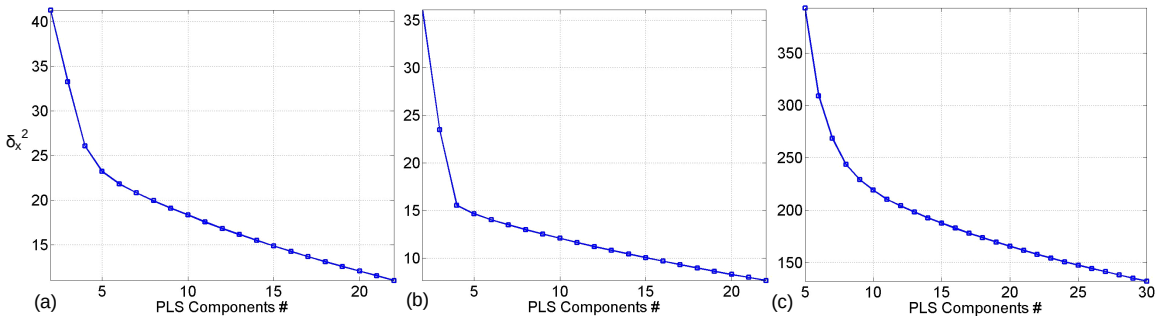


Figure 5.1: Relation between σ_x and components number K: (a) Data set 1; (b) Data set 2; (c) Data set 3.

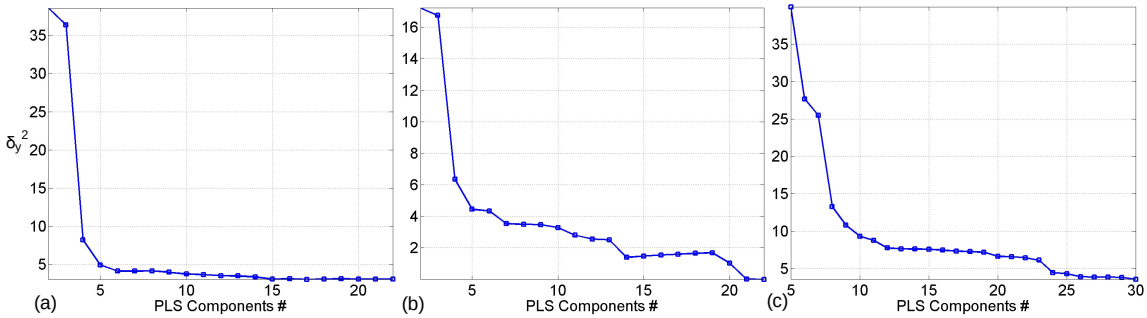


Figure 5.2: Relation between σ_y and components number K: (a) Data set 1; (b) Data set 2; (c) Data set 3.

5 of Algorithm 5. We can see the representation ability of the latent variables are increasing with the increasing numbers of components used and tent to be steady in the end. Figure 5.3 and 5.4 also illustrate the representation and prediction ability of the latent variables of PLSR methods. We can see the representation ability are similar (Figure 5.3), but the prediction ability of PPLSR is lower than PLSR methods (Figure 5.4). Because PLSR methods tend to be over-fitting to training data sets with more components used, which will affects the general prediction ability. PPLSR uses parameter σ_x and σ_y to control the uncertainty of the data set, so the PLS components part will not cover all information of original data sets.

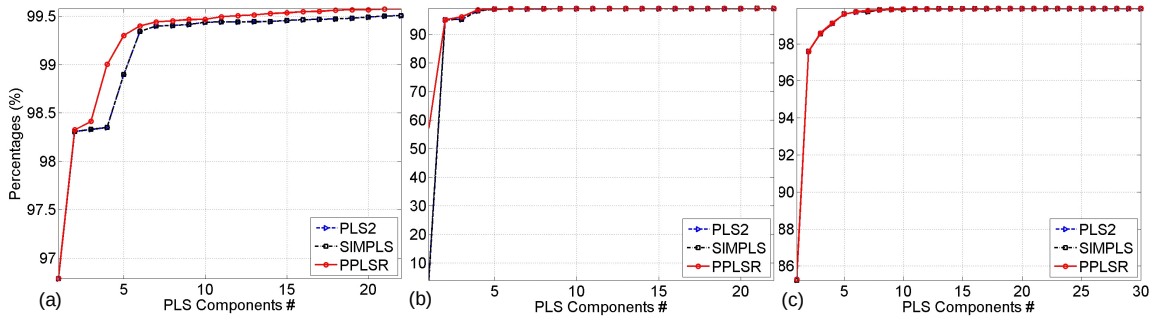


Figure 5.3: Representation of $\mathbf{X}$ by $\mathbf{ZW}_x^T$: (a) Data set 1; (b) Data set 2; (c) Data set 3.

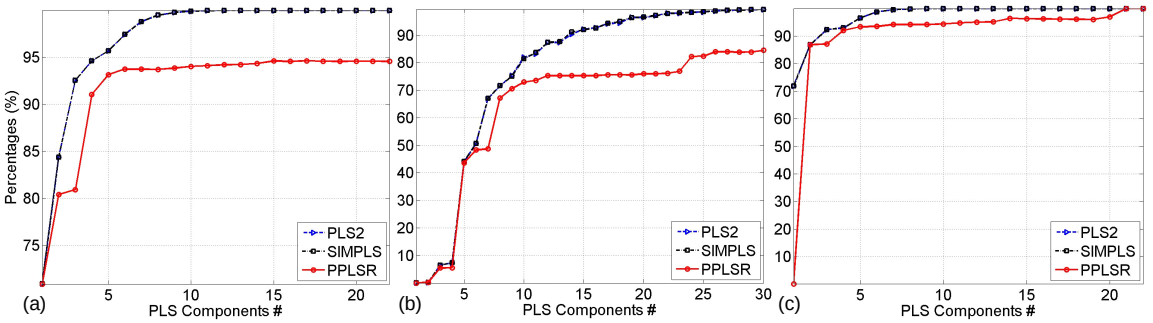


Figure 5.4: Prediction of $\mathbf{Y}$ by $\mathbf{ZW}_y^T$: (a) Data set 1; (b) Data set 2; (c) Data set 3.

5.4 Conclusion and Future Work

Based on the ideas of PPCA and PCCA and the connection between PLS-W2A, PLS2 and SIMPLS, this paper presents a unified probabilistic PLS model, PPLS, to illustrate the traditional PLS regression from a probabilistic point of view: systematic or principal components part and unrelated noise part. Though the experimental results show it is performance is similar to PLSR methods, it provides a solid foundation for future probabilistic and Bayesian models for continuum regression which is more flexible than PLS.

So our future works are to design the probabilistic and Bayesian models for CR methods to easily decide the tuning parameter that control the portions of two objectives.

APPENDIX A

Proofs in Chapter 2

In this appendix, we present some proofs of section 2

A.1 Calculation of generalized inverse of matrix $\mathbf{A}$

The generalized inverse (or pseudoinverse) can be calculated using SVD decomposition [28]. The $(m \times n)$ matrix $\mathbf{A}$ as $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$, where columns of the $(m \times m)$ matrix $\mathbf{U}$ are the left-singular vectors, columns of the $(n \times n)$ matrix $\mathbf{V}$ are the right-singular vectors and diagonal entries of the $m \times n$ diagonal matrix Σ are the decreasing singular values σ_i, with $i = 1, \ldots, \min(m, n)$. If $\mathbf{A}$ is the rank k matrix, where $k < \min(m, n)$, then for $i = k+1, \ldots, \min(m, n)$, $\sigma_i = 0$. Then the generalized inverse of $\mathbf{A}$ is calculated as $\mathbf{A}^- = \mathbf{U}_k\Sigma_k^{-1}\mathbf{V}_k^T$, with columns of $(m \times k)$ matrix $\mathbf{U}_k$ and $(n \times k)$ matrix $\mathbf{V}_k$ are the first k columns of $\mathbf{U}$ and $\mathbf{V}$, and diagonal entries of the $k \times k$ diagonal matrix are the first k σ_i.

A.2 Proof for PCR

(2.17) can be written as:

$$\begin{aligned} \min_{\mathbf{W}} J_{PCR3} &= tr\{(\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{P}_x^T)^T(\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{P}_x^T)\} \\ &= tr(\mathbf{X}^T\mathbf{X} - 2\mathbf{X}^T\mathbf{X}\mathbf{W}\mathbf{P}_x^T + \mathbf{P}_x\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}\mathbf{P}_x^T). \end{aligned} \tag{A.1}$$

Take the derivative of J_{PCR3} to $\mathbf{P}_x$ and set to 0, $\mathbf{P}_x^T$ can be calculated as the generalized inverse of $\mathbf{W}$ (noted as $\mathbf{W}^-$). So the objective function in (2.17) becomes $J = ||\mathbf{X} - \mathbf{X}\mathbf{W}(\mathbf{W})^-||^2$, and so the lengths of projection directions do not affect the

objective function. Without loss of generality, we can have the constraint $\mathbf{W}^T\mathbf{W} = \mathbf{I}$. And since $\mathbf{P}_x^T\mathbf{W} = \mathbf{I}$, $\mathbf{P}_x = \mathbf{W}$. So (A.1) can be rewritten as:

$$\begin{aligned}\min_{\mathbf{W}} J_{PCR3} &= tr(-2\mathbf{P}_x^T\mathbf{X}^T\mathbf{X}\mathbf{W} + \mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}\mathbf{P}_x^T\mathbf{P}_x) \\ &= tr(-2\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W} + \mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}) \\ &= -tr(\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}) = -\sum_{i}^{K}(\mathbf{w}_i^T\mathbf{X}^T\mathbf{X}\mathbf{w}_i) \\ &= \max_{\mathbf{W}} J_{PCR2}.\end{aligned} \tag{A.2}$$

A.3 Proof for RRR

(2.18) can be written as:

$$\min_{\mathbf{W}} J = tr(\mathbf{Y}^T\mathbf{Y} - 2\mathbf{Y}^T\mathbf{X}\mathbf{W}\mathbf{P}_y^T + \mathbf{P}_y\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}\mathbf{P}_y^T). \tag{A.3}$$

Take the derivative of J to $\mathbf{P}_y$ and set to 0, we can get

$$\mathbf{P}_y = \mathbf{Y}^T\mathbf{X}\mathbf{W}(\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W})^{-1}. \tag{A.4}$$

Substitute (A.4) back into (A.3), the objective function of RRR becomes

$$\min_{\mathbf{W}} J = tr(\mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{X}\mathbf{W}(\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W})^{-1}\mathbf{W}^T\mathbf{X}^T\mathbf{Y}).$$

Since the $\mathbf{Y}^T\mathbf{Y}$ is constant, the objective function becomes:

$$\max_{\mathbf{W}} tr[(\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W})^{-1}(\mathbf{W}^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{W})]. \tag{A.5}$$

which equals to (2.20). Because of the constraints $\mathbf{w}_i^T\mathbf{X}^T\mathbf{X}\mathbf{w}_j = 0$, $\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}$ is a diagonal matrix, so (A.5) can be rewritten as:

$$\max_{\mathbf{W}} \sum_{i=1}^{K}(\mathbf{w}_i^T\mathbf{X}^T\mathbf{X}\mathbf{w}_i)^{-1}(\mathbf{w}_i^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_i). \tag{A.6}$$

In (A.6), the scales of all $\{\mathbf{w}_i\}_{i=1}^K$ do not affect the objective function, so they can always be adjusted to make $\mathbf{w}_i^T\mathbf{X}^T\mathbf{X}\mathbf{w}_i = 1$. So (A.6) can be rewritten as:

$$\max_{\mathbf{W}} \sum_{i=1}^{K} (\mathbf{w}_i^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_i) = tr(\mathbf{W}^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{W}), \tag{A.7}$$

with the constraint $\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W} = \mathbf{I}$, which is equal to (2.19).

A.4 Calculation of (2.25) for CCR

When $i = 1$, $j = 0$, from (2.25) we get the Lagrange function as:

$$L = \mathbf{w}_1^T\mathbf{X}^T\mathbf{Y}\mathbf{v}_1 - \lambda_1(\mathbf{w}_1^T\mathbf{X}^T\mathbf{X}\mathbf{w}_1 - 1) - \sigma_1(\mathbf{v}_1^T\mathbf{Y}^T\mathbf{Y}\mathbf{v}_1 - 1)$$

Take the derivative of L to $\mathbf{w}_1$ and $\mathbf{v}_1$ and set to 0, we get:

$$\mathbf{X}^T\mathbf{X}\mathbf{w}_1 = \lambda_1\mathbf{X}^T\mathbf{Y}\mathbf{v}_1 \tag{A.8}$$

$$\mathbf{Y}^T\mathbf{Y}\mathbf{v}_1 = \sigma_1\mathbf{Y}^T\mathbf{X}\mathbf{w}_1 \tag{A.9}$$

By multiplying $\mathbf{w}_1^T$ and $\mathbf{v}_1^T$ on both sides of (A.8) and (A.9) respectively, we get $\lambda_1 = \sigma_1$. From (A.9) we get $\mathbf{v}_1 = \sigma_1(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{X}\mathbf{w}_1$, and substitute into (A.8), we get $\mathbf{w}_1$ as the first eigenvector of matrix $(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})(\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{Y}^T\mathbf{X})$. Similarly we get $\mathbf{v}_1$ as the first eigenvector of matrix $(\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{Y}^T\mathbf{X})(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})$. The corresponding eigenvalues are both λ_1^2.

When $i = 2$, $j = 1$, by multiplying $\mathbf{w}_2^T$ and $\mathbf{v}_2^T$ on both sides of (A.8) and (A.9), constraints $\mathbf{w}_2^T\mathbf{X}^T\mathbf{Y}\mathbf{v}_1 = 0$ and $\mathbf{w}_1^T\mathbf{X}^T\mathbf{Y}\mathbf{v}_2 = 0$ can be deduced from constraints $\mathbf{w}_2^T\mathbf{X}^T\mathbf{X}\mathbf{w}_1 = 0$ and $\mathbf{v}_2^T\mathbf{Y}^T\mathbf{Y}\mathbf{v}_1 = 0$. So the Lagrange multipliers is:

$$\begin{aligned} L = {} & \mathbf{w}_2^T\mathbf{X}^T\mathbf{Y}\mathbf{v}_2 - \lambda_2(\mathbf{w}_2^T\mathbf{X}^T\mathbf{X}\mathbf{w}_2 - 1) - \sigma_2(\mathbf{v}_2^T\mathbf{Y}^T\mathbf{Y}\mathbf{v}_2 - 1) \\ & - \gamma_{21}(\mathbf{w}_2^T\mathbf{X}^T\mathbf{X}\mathbf{w}_1) - \delta_{21}(\mathbf{v}_2^T\mathbf{Y}^T\mathbf{Y}\mathbf{v}_1) \end{aligned} \tag{A.10}$$

Take derivative of L to $\mathbf{w}_2$ and $\mathbf{v}_2$, and set to 0, we get

$$\mathbf{X}^T\mathbf{Y}\mathbf{v}_2 - \lambda_2\mathbf{X}^T\mathbf{X}\mathbf{w}_2 - \gamma_{21}\mathbf{X}^T\mathbf{X}\mathbf{w}_1 = 0 \tag{A.11}$$

$$\mathbf{Y}^T\mathbf{X}\mathbf{w}_2 = \sigma_2\mathbf{Y}^T\mathbf{Y}\mathbf{v}_1 - \delta_{21}\mathbf{Y}^T\mathbf{Y}\mathbf{v}_1 = 0 \tag{A.12}$$

By multiplying $\mathbf{w}_1^T$ and $\mathbf{v}_1^T$ on both sides of (A.11) and (A.12) respectively, we get $\gamma_{21} = \delta_{21} = 0$. So constraints $\mathbf{w}_2^T\mathbf{X}^T\mathbf{X}\mathbf{w}_1 = 0$ and $\mathbf{v}_2^T\mathbf{Y}^T\mathbf{Y}\mathbf{v}_1 = 0$ are also removed. And we can get $\mathbf{w}_2$ and $\mathbf{v}_2$ as the second eigenvector of matrix $(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})(\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{Y}^T\mathbf{X})$ and $(\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{Y}^T\mathbf{X})(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})$. Iteratively, until $i = K$, all 0 constraints are reduced, and all $\mathbf{W}$ and $\mathbf{V}$ are calculated as the the K eigenvectors of matrix $(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})(\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{Y}^T\mathbf{X})$ and $(\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{Y}^T\mathbf{X})(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})$.

A.5 Fast PLS2 Algorithm

$\bar{\mathbf{X}}$ or $\bar{\mathbf{Y}}$ have the same size as $\mathbf{X}$ or $\mathbf{Y}$, with each row is the average vector of matrix $\mathbf{X}$ or $\mathbf{Y}$. Step 4-9 use power method to find the first eigenvector of $\mathbf{Y}^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y}$ corresponding to the biggest eigenvalue. Step 4-11 are the fast way to find the first eigenvector of $\mathbf{X}_i^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}_i$. Step 12 and 13 find the ith PLS component and loading vector $\mathbf{t}_i$ and $\mathbf{p}_i$; step 14 does deflation on $\mathbf{X}$. Step 17 calculates the matrix of regression coefficients and step 18 calculates the matrix of PLS coefficients, with $\mathbf{W} = \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}$ is the PLS projection matrix.

A.6 SIMPLS

Algorithm 6 Fast PLS2 Algorithm

Input: $\mathbf{X}$, $\mathbf{Y}$, K

Output: $\mathbf{B}$

$\mathbf{X}_1 = \mathbf{X} - \bar{\mathbf{X}}$;

$\mathbf{Y} = \mathbf{Y} - \bar{\mathbf{Y}}$;

for $i = 1$ to K **do**

  $\mathbf{A} = \mathbf{Y}^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y}$;

  $\mathbf{d}_i = \mathbf{Y}(1, :)$;

  while $\mathbf{d}_i$ is not converge **do**

    $\mathbf{d}_i = \mathbf{A}\mathbf{d}_i$;

    $\mathbf{d}_i = \mathbf{d}_i / ||\mathbf{d}_i||$;

  end while

  $\mathbf{r}_i = \mathbf{X}_i^T\mathbf{Y}\mathbf{d}_i$;

  $\mathbf{r}_i = \mathbf{r}_i / ||\mathbf{r}_i||$;

  $\mathbf{t}_i = \mathbf{X}_i\mathbf{r}_i$;

  $\mathbf{p}_i = \mathbf{X}_i^T\mathbf{t}_i / (\mathbf{t}_i^T\mathbf{t}_i)$;

  $\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i\mathbf{p}_i^T$;

end for

Store $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_K]; \mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_K]$;

$\mathbf{C} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{Y}$;

$\mathbf{B} = \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{C}$;

Algorithm 7 SIMPLS Iterative Process

$\mathbf{S} = \mathbf{X}^T\mathbf{Y}$; % Get cross-covariance

$\mathbf{P}_0 = []$; % $\mathbf{P}_0$ is empty

for $i = 1$ to K **do**

  $\mathbf{P}_i^{\perp} = \mathbf{I} - \mathbf{P}_{i-1}\mathbf{P}_{i-1}^{+}$; % Get orthogonal subspace

  $\mathbf{S}_i = \mathbf{P}_i^{\perp}\mathbf{S}$; % Get ith cross-covariance

  Solve $\mathbf{S}_i\mathbf{S}_i^T\mathbf{w}_i = \lambda_i\mathbf{w}_i$; % Get projection direction

  $\mathbf{t}_i = \mathbf{X}\mathbf{w}_i$; % Get score vectors $\mathbf{t}_i$; % Get score vector

  $\mathbf{p}_i = \mathbf{X}^T\mathbf{t}_i/(\mathbf{t}_i^T\mathbf{t}_i)$; % Get loading vector

  $\mathbf{P}_i = [\mathbf{P}_{i-1}; \mathbf{p}_i]$;

end for

Store $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_K]$;

$\mathbf{C} = \mathbf{Y}^T\mathbf{T}(\mathbf{T}^T\mathbf{T})^{-1}$;% Get relation coefficients matrix

APPENDIX B

Proofs in Chapter 5

B.1 Marginal and Conditional Gaussian

Given a marginal Gaussian distribution for $\mathbf{x}$ and a conditional Gaussian distribution for $\mathbf{y}$ given $\mathbf{x}$ in the form:

$$p(\mathbf{x}) = N(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \text{ and } p(\mathbf{y}|\mathbf{x}) = N(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1})$$

the marginal distribution of $\mathbf{y}$ and the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ are also Gaussian distribution, and the distribution functions are given by:

$$\begin{aligned} p(\mathbf{y}) &= N(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^T) \\ p(\mathbf{x}|\mathbf{y}) &= N(\mathbf{x}|\boldsymbol{\Sigma}\{\mathbf{A}^T\mathbf{L}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma}) \\ \text{where} \boldsymbol{\Sigma} &= (\boldsymbol{\Lambda} + \mathbf{A}^T\mathbf{L}\mathbf{A})^{-1} \end{aligned}$$

The proof can be found in [29].

B.2 EM Algorithm Details

B.2.1 Expectation

The log likelihood function can be written as:

$$\begin{aligned} &\sum_{n=1}^{N} \ln \int p(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n; \Theta) d\mathbf{z}_n \\ =&\sum_{n=1}^{N} \ln \int Q(\mathbf{z}_n) \frac{p(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n; \Theta)}{Q(\mathbf{z}_n)} d\mathbf{z}_n \\ \geq&\sum_{n=1}^{N} \int Q(\mathbf{z}_n) \ln \frac{p(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n; \Theta)}{Q(\mathbf{z}_n)} d\mathbf{z}_n \\ =&\sum_{n=1}^{N} E_{\mathbf{z}_n|Q} \left[\ln \frac{p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{z}_n; \Theta) p(\mathbf{z}_n)}{Q(\mathbf{z}_n)} \right] \end{aligned} \tag{B.1}$$

The last two steps are based on Jensen's inequality rule. Since natural logarithm function $f(X) = \ln(X)$ is a concave function, we have $E[f(X)] \leq f(E[X])$. Remove items that unrelated with Θ, to maximize (B.1) is equals to (5.14).

B.2.2 Maximization

Since for PPLSR we have:

$$
\begin{aligned}
p(\mathbf{x},\mathbf{y}|\mathbf{z}) = & \frac{1}{(2\pi)^{\frac{D_x+D_y}{2}}} \frac{1}{|\sigma_x^2\mathbf{I}|^{1/2}} \frac{1}{|\sigma_y^2\mathbf{I}|^{1/2}} \\
& \exp\{-\frac{1}{2}[\sigma_x^{-2}(\mathbf{x}-\mathbf{W}_x\mathbf{z}-\boldsymbol{\mu}_x)^T(\mathbf{x}-\mathbf{W}_x\mathbf{z}-\boldsymbol{\mu}_x)] \\
& -\frac{1}{2}[\sigma_y^{-2}(\mathbf{y}-\mathbf{W}_y\mathbf{z}-\boldsymbol{\mu}_y)^T(\mathbf{y}-\mathbf{W}_y\mathbf{z}-\boldsymbol{\mu}_y)]\}
\end{aligned}
$$

So (5.14) can be written as:

$$
\begin{aligned}
& \sum_{n=1}^{N} E_{\mathbf{z}_n|Q}(\ln p(\mathbf{x}_n,\mathbf{y}_n|\mathbf{z}_n;\Theta)) \\
= & -\sum_{n=1}^{N} E_{\mathbf{z}_n|Q}[\frac{D_x}{2}\ln(2\pi\sigma_x^2) + \frac{D_y}{2}\ln(2\pi\sigma_y^2) \\
& + \frac{1}{2\sigma_x^2}\|\mathbf{x}_n-\boldsymbol{\mu}_x\|^2 + \frac{1}{2\sigma_y^2}\|\mathbf{y}_n-\boldsymbol{\mu}_y\|^2 \\
& - \frac{1}{\sigma_x^2}\mathbf{z}_n^T\mathbf{W}_x^T(\mathbf{x}_n-\boldsymbol{\mu}_x) - \frac{1}{\sigma_y^2}\mathbf{z}_n^T\mathbf{W}_y^T(\mathbf{y}_n-\boldsymbol{\mu}_y) \\
& + \frac{1}{2\sigma_x^2}Tr(\mathbf{z}_n\mathbf{z}_n^T\mathbf{W}_x^T\mathbf{W}_x) + \frac{1}{2\sigma_y^2}Tr(\mathbf{z}_n\mathbf{z}_n^T\mathbf{W}_y^T\mathbf{W}_y)]
\end{aligned}
$$

Take the expectations of $\mathbf{z}_n$ from distribution Q to each terms, and take the derivative to all parameters, and set to zero, we get the solutions in (5.15), (5.16) and (5.17).

REFERENCES

[1] Y. Xie, "Surface-enhanced hyper Raman and surface-enhanced Raman scattering: Novel substrates, surface probing molecules and chemical applications," Ph.D. dissertation, Hong Kong University of Science and Technology, 2007.

[2] J. R. Lombardi and R. L. Birke, "A unified approach to surface-enhanced Raman spectroscopy," *J. Phys. Chem. C*, vol. 112, pp. 5605–5617, 2008.

[3] S. Keren, C. Zavaleta, Z. Cheng, A. de la Zerda, O. Gheysens, and S. S. Gambhir, "Noninvasive molecular Imaging of small living subjects using Raman spectroscopy," *PNAS*, vol. 105, pp. 5844–5849, 2008.

[4] C. L. Zavaleta, B. R. Smith, I. Walton, W. Doering, G. Davis, B. Shojaei, M. J. Natan, and S. S. Gambhir, "Multiplexed imaging of surface enhanced Raman scattering nanotags in living mice using noninvasive Raman spectroscopy," *PNAS*, vol. 106, pp. 13 511–13 516, 2009.

[5] S. E. J. Bell and N. M. S. Sirimuthu, "Quantitative surface-enhanced Raman spectroscopy," *Chem. Soc. Rev.*, vol. 37, pp. 1012–1024, 2008.

[6] W. Cheung, I. T. Shadi, Y. Xu, and R. Goodacre, "Quantitative analysis of the banned food dye sudan-1 using surface enhanced Raman scattering with multivariate chemometrics," *J. Phys. Chem. C*, vol. 114, pp. 7285–7290, 2010.

[7] K. Lai, F. Zhai, Y. Zhang, X. Wang, B. A. Rasco, and Y. Huang, "Application of surface enhanced Raman spectroscopy for analyses of restricted sulfa drugs," *Sens. and Instrumen. Food Qual.*, vol. 5, pp. 91–96, 2011.

[8] A. D. Strickland and C. A. Batt, "Detection of carbendazim by surface-enhanced Raman scattering using cyclodextrin inclusion complexes on gold nanorods," *Anal. Chem.*, vol. 81, pp. 2895–2903, 2009.

[9] R. Stosch, A. Henrion, D. Schiel, and B. Guttler, "Surface-enhanced Raman scattering based approach for quantitative determination of creatinine in human serum," *Analytical Chemistry*, vol. 77, no. 22, pp. 7386–7392, 2005.

[10] R. J. Stokes, A. Macaskill, P. J. Lundahl, W. E. Smith, K. Faulds, and D. Graham, "Quantitative enhanced Raman scattering of labeled DNA from gold and silver nanoparticles," *Small*, vol. 3, pp. 1593–1601, 2007.

[11] D. Graham and K. Faulds, "Quantitative SERRS for DNA sequence analysis," *Chem. Soc. Rev.*, vol. 37, pp. 1042–1051, 2008.

[12] H. Zhang, M. H. Harpster, H. J. Park, P. A. Johnson, and W. C. Wilson, "Surface-enhanced Raman scattering detection of DNA derived from the west nile virus genome using magnetic capture of Raman-active gold nanoparticles," *Anal. Chem.*, vol. 83, pp. 254–260, 2011.

[13] J.-H. Kim, J.-S. Kim, H. Choi, S.-M. Lee, B.-H. Jun, K.-N. Yu, E. Kuk, Y.-K. Kim, D. H. Jeong, M.-H. Cho, and Y.-S. Lee, "Nanoparticle probes with surface enhanced Raman spectroscopic tags for cellular cancer targeting," *Anal. Chem.*, vol. 78, pp. 6967–6973, 2006.

[14] D. C. Kennedy, K. A. Hoop, L.-L. Tay, and J. P. Pezacki, "Development of nanoparticle probes for multiplex SERS imaging of cell surface proteins," *Nanoscale*, vol. 2, pp. 1413–1416, 2010.

[15] J. L. Abell, J. M. Garren, J. D. Driskell, R. A. Tripp, and Y. Zhao, "Label-free detection of micro-RNA hybridization using surface-enhanced raman spectroscopy and least-squares analysis," *Journal of the American Chemical Society*, vol. 134, no. 31, pp. 12 889 – 12 892, 2012.

[16] A. Loren, "Quantitative surface enhanced Raman spectroscopy," *Doktorsavhandlingar vid Chalmers Tekniska Hogskola*, no. 2090, pp. i+ 1–33, 2004.

[17] Y. Zhang, Y. Huang, F. Zhai, R. Du, Y. Liu, and K. Lai, "Analyses of enrofloxacin, furazolidone and malachite green in fish products with surface-enhanced Raman spectroscopy," *Food Chemistry*, vol. 135, no. 2, pp. 845 – 850, 2012.

[18] B. Liu, P. Zhou, X. Liu, X. Sun, H. Li, and M. Lin, "Detection of pesticides in fruits by surface-enhanced Raman spectroscopy coupled with gold nanostructures," *Food and Bioprocess Technology*, vol. 6, no. 3, pp. 710 – 718, 2013.

[19] K. E. Shafer-Peltier, C. L. Haynes, M. R. Glucksberg, and R. P. Van Duyne, "Toward a glucose biosensor based on surface-enhanced Raman scattering," *Journal of the American Chemical Society*, vol. 125, no. 2, pp. 588 – 593, 2003.

[20] J. D. Driskell, O. M. Primera-Pedrozo, Y. Z. Richard A. Dluhy, and R. A. Tripp, "Quantitative surface-enhanced Raman spectroscopy based analysis of microRNA mixtures," *Appl. Spectrosc.*, vol. 63, pp. 1107–1114, 2009.

[21] L. Zhang, Q. Li, W. Tao, B. Yu, and Y. Du, "Quantitative analysis of thymine with surface-enhanced Raman spectroscopy and partial least squares (PLS) regression," *Anal. Bioanal. Chem.*, vol. 398, pp. 1827–1832, 2010.

[22] M. Ratkaj, T. Biljan, and S. Miljanic, "Quantitative analysis of entacapone isomers using surface- enhanced Raman spectroscopy and partial least squares regression," *Applied Spectroscopy*, vol. 66, no. 12, pp. 1468 – 1474, 2012.

[23] C. Gobinet, V. Vrabie, M. Manfait, and O. Piot, "Preprocessing methods of Raman spectra for source extraction on biomedical samples: application on paraffin-embedded skin biopsies," *IEEE Trans. Biomed. Eng.*, vol. 56, pp. 1371–1382, 2009.

[24] F. Gan, G. Ruan, and J. Mo, "Baseline correction by improved iterative polynomial fitting with automatic threshold," *Chemometrics Intell. Lab. Syst.*, vol. 82, pp. 59–65, 2006.

[25] P. Du, W. A. Kibbe, and S. M. Lin, "Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching," *Bioinformatics*, vol. 22, pp. 2059–2065, 2006.

[26] R. G. Brereton, "Introduction to multivariate calibration in analyticalchemistry," *Analyst*, vol. 125, pp. 2125–2154, 2000.

[27] M. J. Pelletier, "Quantitative analysis using Raman spectroscopy," *Appl. Spec.*, vol. 57, pp. 20A–42A, 2003.

[28] G. H. Golub and C. F. V. Loan, *Matrix Computations (third edition).* The Johns Hopkins University Press, 1996.

[29] C. M. Bishop, *Pattern Recognition and Machine Learning*, M. Jordan, J. Kleinberg, and B. Scholkopf, Eds. Springer Press, 2006.

[30] A. E. Hoerl and R. W. Kennard, "Ridge regression: applications to nonorthogonal problems," *Technometrics*, vol. 12, pp. 69–82, 1970.

[31] R. Sundberg, "Small sample and selection bias effects in calibration under latent factor regression models," *J. Chemometr.*, vol. 21, pp. 227–238, 2007.

[32] R. Bro, "Multivariate calibration: What is in chemometrics for the analytical chemist?" *Anal. Chim. Acta*, vol. 500, p. 185194, 2003.

[33] I. T. Jolliffe, *Principal Component Analysis, Second Edition.* Springer, 2002.

[34] M. K.-S. Tso, "Reduced-Rank regression and canonical analysis," *J. R. Stat. Soc. Ser. B-Stat. Methodol.*, vol. 43, pp. 183–189, 1981.

[35] A. J. Burnham, R. Viveros, and J. F. MacGregor, "Frameworks for latent variable multivariate regression," *J. Chemometr.*, vol. 10, pp. 31–45, 1996.

[36] K. Worsley, J. B. Poline, K. J. Friston, and A. C. Evans, "Characterizing the response of PET and fMRI data using multivariate linear models," *Neuroimage*, vol. 6, pp. 305–319, 1997.

[37] L. Sun, S. Ji, S. Yu, and J. Ye, "On the equivalence between canonical correlation analysis and orthonormalized partial least squares," in *Int. Joint conf. Artificial Intelligence (IJCAI)*, 2009.

[38] J. Arenas-Garca, K. B. Petersen, and L. K. Hansen, "Sparse kernel orthonormalized PLS for feature extraction in large data sets," in *Adv. neural inf. proces. syst.*, 2007.

[39] H. Wold, *Path models with latent variables: the NIPALS approach*, H. M. B. et al., Ed. Academic, 1975.

[40] S. Wold, M. Sjstrma, and L. Eriksson, "PLS-regression: a basic tool of chemometrics," *Chemometrics Intell. Lab. Syst.*, vol. 58, pp. 109–130, 2001.

[41] H. Wold, "Soft modelling, the basic design and some extensions," *Systems Under Indirect Observation*, vol. 2, pp. 589–591, 1982.

[42] A. Hoskuldsson, "PLS regression methods," *J. Chemometr.*, vol. 2, pp. 211–228, 1988.

[43] J. A. Wegelin, "A survey of partial least squares (PLS) methods, with emphasis on the two-block case," University of Washington, Department of Statistics, Tech. Rep., 2000.

[44] R. Rosipal and N. Kramer, "Overview and recent advances in partial least squares," *LNCS*, vol. 3940, pp. 34–51, 2006.

[45] S. de Jong, "SIMPLS: an alternative approach to partial least squares regression," *Chemometrics Intell. Lab. Syst.*, vol. 18, pp. 251–263, 1993.

[46] C. J. F. ter Braak and S. de Jong, "The objective function of partial least squares regression," *J. Chemometr.*, vol. 12, pp. 41–54, 1998.

[47] S. Li, J. Gao, J. O. Nyagilo, and D. P. Dave, "Probabilistic partial least square regression: a robust model for quantitative analysis of Raman spectroscopy data," in *Proc. IEEE Int. conf. Bioinformatics and Biomedicine (BIBM)*, 2011, pp. 526–531.

[48] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educational Psychology*, vol. 24, pp. 417–441, 1933.

[49] D. R. Hardoon, S. R. Szedmak, and J. R. Shawe-Taylor, "Canonical correlation analysis: an overview with application to learning methods," *Neural Comput.*, vol. 16, pp. 2639–2664, 2004.

[50] J. R. Kettenring, "Canonical analysis of several sets of variables," *Biometrika Trust*, vol. 58, pp. 433–451, 1971.

[51] C. Dhanjal, S. R. Gunn, and J. Shawe-Taylor, "Efficient sparse kernel feature extraction based on partial least squares," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, p. 13471361, 2009.

[52] A. Kembhavi, D. Harwood, and L. S. Davis, "Vehicle detection using partial least squares," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, p. 12501265, 2011.

[53] A. Sharma and D. W. Jacobs, "Bypassing synthesis: PLS for face recognition with pose, low-resolution and sketch," in *IEEE conf. Computer Vision and Pattern Recognition (CVPR)*, 2011.

[54] A. Tishler, D. Dvir, A. Shenhar, and S. Lipovetsky, "Identifying critical success factors in defense development projects: a multivariate analysis," *Technol. Forecast. Soc. Change*, vol. 51, pp. 151–171, 1996.

[55] A. Tishlera and S. Lipovetsky, "Modelling and forecasting with robust canonical analysis: method and application," *Romput. Oper. Res.*, vol. 27, pp. 217–232, 2000.

[56] I. Daubechies, *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA., 1992.

[57] S.-J. Baek, A. Park, A. Shen, and J. Hu, "A background elimination method based on linear programming for raman spectra," *J. Raman Spectrosc.*, vol. 42, pp. 1987–1993, 2010.

[58] E. M. Qannari and M. Hanafi, "A simple continuum regression approach," *J. Chemometr.*, vol. 19, pp. 387–392, 2005.

[59] S. de Jong and H. A. Kiers, "Principal covariates regression part i. theory," *Chemometrics Intell. Lab. Syst.*, vol. 14, pp. 155–164, 1992.

[60] S. Bougearda, M. Hanafi, and E. Qannari, "Continuum redundancy-pls regression: A simple continuum approach," *Comput. Stat. Data Anal.*, vol. 52, pp. 3686–3696, 2008.

[61] R. Rosipal and N. Kramer, "Overview and recent advances in partial least squares," *LNCS*, vol. 3940, pp. 34–51, 2006.

[62] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. siam, 2000.

[63] H. A. L. Kiers and A. K. Smilde, "A comparison of various methods for multivariate regression with highly collinear variables," *Stat. Method. Appl.*, vol. 16, pp. 193–228, 2007.

[64] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 61, pp. 611–622, 1999.

[65] F. R. Bach and M. I. Jordan, "A probabilistic interpretation of canonical correlation analysis," 688, Department of Statistics, University of California, Berkeley, Tech. Rep., 2005.

BIOGRAPHICAL STATEMENT

Shuo Li was born in Tangshan, China, in 1983. He received his B.S. in Software Engineering and M.S.degree in Computer Science from Sichuan University, China, in 2006 and 2009, respectively, and began his Ph. D study in CSE in 2008 Fall. From 2008 to now, he is working in the BioMeCIS Lab with Dr. Jean Gao. His current research interest is multivariate analysis methods, Wavelet techniques, signal processing and Bayesian nonparametrics.