Imperial College of Science, Technology and Medicine
Department of Computing

Spectroscopy Motivated Probabilistic Latent Variable Models

James Odgers

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College
December 2024

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

Latent variable models are a powerful approach for capturing underlying structure in high dimensional data. One such setting for this high dimensional data is spectroscopy, which has an inherent structure due to the physical process generating the data. Spectroscopy has a wide range of applications, each of which results in specific sets of challenges, this thesis is motivated by the challenge presented by monitoring the manufacture of Pharmaceuticals.

This thesis is interested in how to use a probabilistic approach to latent variable models to the challenge through two main contributions focused on uncertainty quantification and encoding flexible, physically relevant priors. We first tackle uncertainty estimation in Partial Least Squares (PLS) regression, a widely used technique in spectroscopy. While existing methods rely on problematic linear approximations, we develop a bootstrap-based approach that naturally captures non-linear parameter interactions. We demonstrate its effectiveness across multiple pharmaceutical case studies, showing particular strength in Design Space identification where accurate uncertainty estimates are crucial.

The second part of the thesis introduces a novel probabilistic framework, the Weighted-Sum Gaussian Process Latent Variable Model (WS-GPLVM), which combines physical understanding from Beer-Lambert's law with flexible Gaussian Process models. This model allows variations in conditions which cause changes in the pure component spectra to be found via including additional latent variable. We develop this model over the course of two chapters of this thesis, resulting in a novel, flexible method with demonstrable real world effectiveness.

Acknowledgements

I have been unbelievably fortunate with the number of people who have helped me throughout my PhD. The number of people who have enriched my experience over the past 4 years is too large to be practical to write in this length of thesis, but as a partial and inadequate list I would like to thank:

- My supervisors Sarah and Ruth. The decision to do a PhD supervised by you both was truly one of the best decisions I have ever made. You have taught me so much about how to be an effective researcher, collaborator, and communicator. While I was learning all this from you a lot of life was happening and I truly could not have asked for better support from you both in all of this as well.

- Chryssa, I could not have asked for a better or kinder role model to aspire towards and I hope that I will be able to be half as good a post-doc to others as you were to me.

- Sal, the industry perspective you provided added a level of richness to my research which I would not have been capable of achieving otherwise and I regularly think that of the example that you set. The trips I took to Lilly were some of the most interesting highlights of my PhD.

- The members of 302 and the Computational Optimisation Group is the group which I have had the chance to share an office with and I could not have asked for a smarter or kinder group to have spent my PhD with. Jose who I look to as an example of how to make the most out of both life and work, Ruby whose collaboration I cannot speak highly enough of, Alex who is one of the funniest and smartest people I've had the chance to speak to, Daniel whose enthusiasm for research and willingness to discuss my questions brightened my days, Juan who gives the best hugs, Jixiang whose patience with my questions and work ethic are both unmatched, Shiqiang who has nothing but the best ideas on all topics, Toby who is always a pleasure to speak to and always commits to the bit, Gabriel who always made me laugh and had undoubtedly the best hair of the group, Phil who I had the most interesting conversations with, Yilin who is one of the sweetest people I have met, and Becky whose conversation was a delight.

- The Filippi Lab, the other research group I was lucky enough to be part of: Mike who I had one of the best weeks of my PhD in Vienna with, Ben whose explanations of bandits is the closest I think I'll ever get to understanding real math, and Guiomar whose networking skills I hope to be able to emulate a fraction of. Also to those I wish I had overlapped with more, Junyang and Theodore, I am delighted to see the group will continue to thrive after I have graduated.

- My flatmates, George, Rob, and David for putting up with me and for enlivening my evenings.

- To all those who I have fenced, travelled, cycled, skied, and celebrated with. The past few years have been filled with some of the best weeks of my life and I am so grateful to all those I have shared those experiences with.

- Olivia, who has massively improved my life over the last 3 years.

- Finally, to my family: Mum, Dad, Hester and Rosie. Without you I would never have made it to a PhD, and certainly would not have made it through.

Contents

Abstract **4**

Acknowledgements **5**

1 Introduction **1**

1.1 Motivation and Objectives 1

1.2 Contributions 4

1.3 Notation 5

1.4 Statement of Originality 5

1.5 Publications 6

2 Background **7**

2.1 Introduction 7

2.2 Background for Partial Least Squares 7

2.2.1 NIPALS Algorithm 10

2.2.2 Equivalent Linear Predictors to PLS 14

2.2.3 Partial Least Squares in this Thesis 15

2.3 Gaussian Processes 15

2.3.1 Introduction 15

2.3.2 Gaussian Process Latent Variable Models 17

2.3.3 Variational Inference for Gaussian Process Latent Variable Models 18

2.3.4 Gaussian Process Latent Variable Models in this Thesis 21

3 Uncertainty estimation in Partial Least Squares with Bootstrap 22

3.1 Introduction 22

3.1.1 Working Between Communities 22

3.1.2 Uncertainty in PLS 24

3.2 Existing Uncertainty Techniques in PLS 26

3.3 Method 31

3.4 Applications 34

3.4.1 Computation and Code 34

3.4.2 Setup 34

3.4.3 Identifying a Design Space 35

3.4.4 Simple Simulated Example 39

3.4.5 Michael Addition Reaction 42

3.4.6 Reductive Amination 43

3.4.7 High Shear Wet Granulation 45

3.4.8 Spectroscopy 46

3.5 Conclusions 47

4 Weighted-Sums of Gaussian Process Latent Variable Models 50

4.1 Introduction . . . 50

4.2 Weighted-Sum of Gaussian Process Latent Variable Models . . . 53

4.2.1 Extensions . . . 59

4.3 Implementation . . . 60

4.4 Experiments . . . 64

4.4.1 Toy Example . . . 64

4.4.2 Examples on Datasets from the Literature . . . 65

4.5 Discussion . . . 70

5 Variational Methods for Practical Probabilistic Variable Signal Spectroscopy 72

5.1 Introduction . . . 72

5.2 Method . . . 73

5.2.1 Unobserved Components . . . 73

5.2.2 Reformulating the ELBO . . . 75

5.2.3 Pure Component Spectra . . . 81

5.2.4 Structured Variational Inference for WS-GPLVMs . . . 85

5.3 Implementation Details . . . 87

5.3.1 Learning Dependent or Independent Latent Variables . . . 87

5.3.2 Selecting the Number of Unobserved Components . . . 88

5.3.3 Optimization . . . 89

5.3.4 Sampling the Inducing Points for Structured Variational Inference . . . 90

5.4 Related Works . . . 91

5.5 Experiments 92

5.5.1 Toy Example with Unobserved Components 92

5.5.2 Real Spectroscopy Datasets 100

5.5.3 Limits of the use of ELBO as an objective 104

5.6 Discussion 106

6 Conclusion **109**

6.1 Summary of Thesis Achievements 109

6.2 Applications 110

6.3 Future Work 111

Bibliography **112**

A Table of notation for Chapter 3 **126**

B Table of Notation for Chapter 4 **128**

C Table of Notation for Chapter 5 **130**

D Appendices for Chapter 4 **132**

D.1 Additional Results 132

D.2 ELBO Derivations 133

D.2.1 Full ELBO derivation 133

D.2.2 Derivation of the ELBO with Independent Measurements 139

D.3 Expressions for ξ_i 142

D.4 Code, Data and Computation 144

D.4.1 Comparison Methods 144

D.5 Implementation Details for Comparison Methods in Chapter 4 146

D.5.1 Inverse Linear model of Coregionalization 148

D.5.2 Classical Least Squares Gaussian Process 149

D.5.3 Classical Least Squares 149

D.5.4 Partial Least Squares 150

D.6 Additional Details for Experiment Setup 150

D.6.1 Toy Example 150

D.6.2 Near Infra Red Spectroscopy with varying temperatures 150

D.6.3 Oil Flow Regression 150

D.6.4 Remote Sensing Rock Classification 151

List of Tables

5.1 Results for the Dalby dataset with 90/10 training/test split. 102

5.2 Results for the Dalby dataset with 50/50 training/test split. 102

5.3 Results for the Wulfert dataset with 90/10 training/test split. 103

5.4 Results for the Wulfert dataset with 50/50 training/test split. 103

5.5 Time taken to optimise model. Note that the WS-GPLVM-ind. results are used as this corresponds to the model used in these chapters. The performance of the WS-GPLVM-ind. on the Wulfert dataset given in Table D.1, and is significantly below the results presented here. 103

D.1 Results of the Weighted-Sum GPLVM (WS-GPLVM), Multi-output Gaussian Process (ILMC), and Partial Least Squares (PLS). Regression results show Negative Log Predictive Density (NLPD) and Mean Squared Error (MSE) of the prediction. Each example was run 10 times and the mean is quoted here, along with two times the standard error of the mean in brackets. For results that deviate significantly from one, powers of ten were removed from all of the results and where applicable these are shown in square brackets at the top of the column. The best mean for each column is in bold. 133

D.2 Results of the Weighted-Sum GPLVM (WS-GPLVM), Multi-output Gaussian Process (ILMC), and Partial Least Squares (PLS). Classification results report the Log Predicted Probability (LPP), the Accuracy (Acc.) and the Area Under the Receiver Operator Curve (AUC). Each example was run 10 times and the mean is quoted here, along with two times the standard error of the mean in brackets. For results that deviate significantly from one, powers of ten were removed from all of the results and where applicable these are shown in square brackets at the top of the column. The best mean for each column is in bold. . . 133

D.3 Hyperparameter initializations for WS-GPLVM and WS-GPLVM-ind. When the initialization is a distribution, the initial value is drawn from that distribution. Initalization is 'grid' means a grid of points across the whole input domain. . . 147

List of Figures

3.1 Diagram of the proposed method for generating a probabilistic prediction from PLS using bootstrap. (a) The original data set is sampled with replacement to produce bootstrapped data sets. (b) The bootstraped parameters are estimated using the standard NIPALS algorithm described in Section 2.2.1. (c) The bootstrapped parameters are used to generate a prediction for the output of PLS using a normal distribution with mean given by the prediction from PLS and variance given by the mean squared error of prediction. (d) The estimates from each of the bootstrapped samples are averaged out to give an estimate for the overall probabilistic distribution of the output given the input and observed data. 32

3.2 Comparison of the two predictions of the probability density function from the zeroth order approximation method and the bootstrap prediction. The solid lines show the bootstrap method proposed here, the dashed line shows the prediction from the zeroth order approximation method, and the shaded area is the desired output range for y. As can be seen above, there is no difference in the predictions from the two points using the student-t method, whereas the bootstrap method applies significant weight to a far wider range of values. 40

3.3 Figure showing the Design Space for a simple simulated example. The dashed lines indicate the true probabilities of data generated at that point falling in the desired range. The red and purple points indicate points of interest discussed in Figure 3.2. Figure (a) shows the latent spaces (orange lines) found by the bootstrapped data sets, and the resulting Design Spaces using probabilistic bootstrap prediction – whose shape depends on the distance to the latent space. Figure (b) shows the latent space (orange line) found by the regular PLS model using the training data set, the resulting Design Spaces (blue and green solid lines), do not account for the distance to the latent space. 41

3.4 Design Spaces for the Michael addition reaction using bootstrapped and zeroth order methods – Figure (a) shows the Design Space from the bootstrap method, and Figure (b) showing the zeroth order method. The axes indicate different values for two independent inputs to the manufacturing process. The contours indicate the differing probabilities of given inputs being within the Design Space, with the solid lines indicating the probabilities from the PLS methods, while the dashed lines indicate the true probabilities of that input resulting in an acceptable output. 42

3.5 Projection of randomly sampled points in the input space to the two dimensional space of just reaction temperature and nitrogen addition temperature. The differently colored points indicate the different probabilities for the given input resulting in an acceptable output. Figure (a) shows the probabilities found using the bootstrap method proposed here, Figure (b) shows the probabilities using the zeroth order method. The same point appears to have different probabilities of being withing the Design Space as they have been projected down from a higher dimension - in reality these points are separated along one or more of the other dimensions not shown here. 44

3.6 Violin plot for the data given in Vemavarapu and Badawy [2019]. The blue shapes indicate the probability distribution found by the bootstrap prediction, whereas the orange shows the prediction from the zeroth order approximation method. The points are the prediction from PLS before uncertainty is included and the crosses are the true value of the % oversized particles. The blue line horizontal line is the mean value of the training data outputs. For the sake of clarity both violin plots are truncated at the 0.025 and 0.0975 quantiles of the plot to remove tales from both distributions. 45

3.7 The prediction from bootstrap prediction and predictions using a zeroth-order approximation. The blue shapes indicate the probability distribution found by the bootstrap prediction, whereas the red shows the prediction from the zeroth-order approximation method. The points are the prediction from PLS before uncertainty is included and the crosses indicate the observed values for the API fraction. The two predictions are indistinguishable for the 1000 point training data set, and nearly identical for the 50 point training data set. 49

4.1 Bayesian diagram for Weighted-Sum of Gaussian Process latent variable models (WS-GPLVM) with unobserved mixture weights r_{ic}. 51

4.2 Illustrative example of WS-GPLVM: by inputting a set of data (a) of training data *(top)*, with known mixture fractions, and test data *(bottom)*, with unknown mixture fractions, WS-GPLVM produces variational posterior estimates of (b) the fractional composition of the test data *(top left)*, the latent variables of both the training and the test data *(bottom left)*, and a Gaussian Process estimate, shown with a mean and 95% confidence interval, for the pure component signals *(right)*. 54

4.3 *(a)* Shows the training and test data after SNV was applied. Each orange line represents a training data point and each blue line is a test data point. *(b)* Shows estimated mass fractions. The blue points show the true values, the black crosses show the mean of our method's variational posterior, the orange points are 100 samples drawn from the variational posterior. *(c)* and *(d)* show the latent variables coloured by temperature (purple for $30°C$, yellow for $70°C$) which we expected to cause variation in the pure signals, and water fraction (purple for no water, yellow for pure water), where the variation is induced by the normalizing procedure. *(e)* Mean estimated pure component signals for the training and test spectra, evaluated at the mean of the learned latent variable and coloured by the temperature of the measurement (red for $30°C$, yellow for $70°C$). 65

4.4 Mean squared error (MSE) and negative log predictive density (NLPD) for 10 train-test splits of the a) near infra-red spectroscopy and b) oil flow datasets. . . 68

4.5 Training and test data for the remote sensing rock classification example. 69

4.6 Accuracy, log predictive probability (LPP) and area under receiver operator curve (ROC AUC) for remote sensing rock classification for a 10-fold cross-validation. 70

5.1 Gaussian process parameters for the toy example with unobserved components. The top row of each plot is the scale parameter of the GP kernel. The bottom two rows of each plot shows the reciprocal of the length-scale; a larger value indicates more variation in that dimension. Both the mean field and structured approaches are able to: (*i*) identify that only one unobserved component is required to fit the data, and (*ii*) identify when the two significant components share a single significant latent variable or two distinct latent variables. An interesting distinction in the optimisation between the mean field variational families (*left column*) and the structured families (*right column*) is that the the structured families learn shorter length scales for variation. This does not affect predictions, as for both the scale of that component is very close zero. However, it indicates that the structured family has a markedly different loss landscape than the mean field family which drives the variance of the learned function to zero much faster. 93

5.2 Samples from the variational posterior for the latent space of the test data $h_i^* \sim q(h_i^*)$. The latent spaces for the toy example introduced in this chapter for mean field (*left column*) vs. structured (*right column*) variational families, and variation caused by a single (*top row*) vs. independent (*bottom row*) latent variables. For models with a single latent variable, the distribution of the points resemble the prior along one dimension, indicating that the model has correctly identified that the model has correctly identified that only one latent variable is required for all data points. Meanwhile, for two independent latent variables the distribution of the point in in the latent space depends on the pure component weights for that sample. If a particular data point had very little of one fraction it is impossible to estimate the latent variable associated with that component can't be estimated accurately. This results in a distribution of points in the latent space which could have high uncertainty in either of the latent dimensions depending on the relative abundances of the components. This can be seen clearly in the bottom row of figures, where points with low component one (dark coloured points) have a high variance in x_1 and low variance in x_2, while points with high component one (light points) have low variance in x_i and high variance in x_2. The structured variational families appear to have a generally higher uncertainty than their mean field counterparts. 94

5.3 Pure component spectra mean and 95% credible interval estimated by the uncollapsed WS-GPLVMs. The estimates are taken at the origin of the latent space ($x = (0, 0)^T$) The left column shows the mean field estimates, while the right column shows the structured estimates. The top row shows the case where a single latent variable controlling both of the components, while the lower row shows the estimates for the two components independent. The signal associated with the observed component is shown as orange, while WS-GPLVM's two components associated with unobserved components are shown in blue and green. In the case where there is a single latent variable which controls both peaks the model incorrectly learns that pure component one has a second peak at the location of pure component two. This does not affect the predictions as the model correctly learns that there is also a peak for the unobserved component there, allowing the signals to cancel one another out, but it does indicate that the pure component spectra estimated by the model must be treated with scepticism. As in the latent space, the structured model does a better job of estimating the uncertainty and has appropriately learned to be sceptical about its own estimate for the pure component signal around the location of the peak with the unobserved signal. . 95

5.4 Plots of predictions for the pure component weights of the toy example introduced in this chapter for mean field *(left)* and structured *(right)*, and shared *(top)* and independent *(bottom)* latent variables. Both variational families are able to make good predictions for both conditions. The structured variational family has marginally larger error bars, however for this simple example it is not significant. This minor difference between the predictions of the structured and mean field families indicates that each spectra simply contains more information than is needed to produce a good estimate for the pure component weights, in a more complex example in which several wavelengths needed to be considered together in order to make good predictions we expect that there would be a more significant difference in the predictions between the two variational families. . . 96

5.5 Plots of ELBOs against MSEP for the Dalby dataset with differing numbers of inducing points (8,16) for a 90/10 split on the training/test data fractions. For the lower number of inducing points the increasing the ELBO predicts a decrease in the MSEP (Figure 5.5a), meaning for this setting selecting the model with the highest ELBO is a good criterion for model selection. However, in the model with a larger number of inducing points there appear to be multiple local optima and selecting the maximum ELBO always results in selecting a model with a high MSEP (Figure 5.5b). 107

Chapter 1

Introduction

1.1 Motivation and Objectives

This thesis emerged from collaborative work with a pharmaceutical company, focusing on addressing key challenges in manufacturing process monitoring and control. While drug development frequently captures the attention of the machine learning research community, the manufacturing process, though equally critical has received comparatively less attention. The manufacturing phase must meet extremely exacting standards - when purchasing a 200mg ibuprofen tablet, consumers rightfully expect both that the active pharmaceutical ingredient (API) amount closely matches the labeled 200mg and that the tablet contains only ibuprofen as its API. These stringent requirements drive the need for sophisticated analytical methods and careful process control.

A key technique for carrying out this quality control is spectroscopy. As a non-destructive analytical technique, spectroscopy can rapidly analyse materials by measuring their interaction with electromagnetic radiation across different wavelengths (λ). When electromagnetic radiation encounters a material, various interactions can occur - the radiation may be absorbed, transmitted, scattered, or reflected. Modern spectrometers measure these interactions simultaneously at hundreds or thousands of distinct wavelengths, typically denoted as $\lambda_1, \ldots, \lambda_M$. For each wavelength, the spectrometer records the intensity of the measured radiation after it

interacts with the sample. The resulting spectrum - the pattern of intensities across all measured wavelengths - serves as a characteristic fingerprint of the material being analysed. This fingerprint arises because different molecular structures and chemical bonds interact preferentially with radiation at specific wavelengths, meaning the measured intensities contain detailed information about the chemical composition and structure of the sample.

In pharmaceutical manufacturing, spectroscopic methods like Near-Infrared (NIR) and Raman spectroscopy are particularly valuable as they enable real-time monitoring without destroying samples. These techniques can quantify Active Pharmaceutical Ingredient (API) concentrations, monitor chemical reactions during synthesis, verify raw material quality, and detect potential contaminants - all critical capabilities for maintaining product quality and consistency.

The physical basis for spectroscopy comes from Beer-Lambert's law. This law arises from considering the absorption of light as it passes through a medium made up of C different chemical substances with concentrations $r = (r_1, \dots r_C)$ and describes the change of the intensity of light at a particular wavelength ($I(\lambda)$) as follows:

$$\frac{dI(\lambda)}{dx} = -\sum_{c=1}^{C} \left[\alpha_c(\lambda) r_c\right] I(\lambda) \tag{1.1}$$

where α_c is the absorption coefficient of the material. The solution to this differential equation gives Beer-Lambert's law, which states that the absorbance of light by a chemical species at a particular wavelength is directly proportional to both its concentration and the path length through which the light travels. After taking the logarithm of the solution to the differential equation the signal $d(\lambda)$ is given by

$$d(\lambda) = \sum_{c=1}^{C} r_c s_c(\lambda) \tag{1.2}$$

where $s_c(\lambda)$ is the pure component signal, that is the signal which would be observed if only one chemical was present. By combining measurements of the signal across many wavelengths $d = (d(\lambda_1), \dots, d(\lambda_M))$ Beer-Lambert allows the concentrations of the chemical species r to be calculated.

However, applying Beer-Lambert's law directly to pharmaceutical manufacturing presents several challenges. The pure component spectra - the characteristic absorption patterns of individual chemical species - often vary with environmental conditions like temperature and humidity. These variations mean that a component's spectroscopic signature is not fixed but rather changes based on the measurement conditions. Additionally, the presence of physical phenomena like light scattering and chemical interactions between components can lead to deviations from the simple linear mixing model. Signal interference from non-chemical sources, such as changes in sample presentation or instrument drift, further complicates the analysis.

Due to these complexities, directly applying the physics based approach is typically not taken in practice. Instead the problem is treated as being a linear regression problem in which the observed spectra at a set of wavelengths is treated as an input and the concentrations are treated as an output, so the predictions for the weights are given by

$$\hat{r} = \beta d, \tag{1.3}$$

where $\beta \in \mathbb{R}^{C \times M}$ is a matrix to be found.

Adding to the challenges of working in the pharmaceutical setting, the models for monitoring need to be developed at a very early stage of the manufacturing process when only a small amount of API is available. As a result only a relatively small number of training data points are available. Combining this with the fact that spectroscopy typically consists of measuring hundreds of wavelengths results in the necessity to use models which contain strong inductive biases to select relevant features.

The most popular of these techniques is Partial Least Squares (PLS) regression which addresses these challenges by combining dimensionality reduction with supervised learning. The key inductive bias in PLS is the assumption that both the input spectra and output concentrations can be described by a small number of underlying latent variables. This assumption is particularly well-suited to spectroscopic data as it aligns with the physical understanding that the hundreds of wavelength measurements are not independent, but rather reflect the underlying

molecular properties of the chemical components present in the sample. PLS finds these latent variables through an iterative process that maximizes the covariance between the input and output spaces, ensuring that the reduced dimensional representation captures variations relevant to predicting the concentrations.

This thesis is interested in applying probabilistic approaches to both of these approaches to spectroscopy. The first focuses on developing improved uncertainty quantification methods for existing analysis techniques. This work extends PLS regression to provide conditional probabilistic predictions through estimating parametric uncertainties.

The second research direction involves creating new probabilistic models specifically designed for spectroscopic analysis. These novel Gaussian Process-based models explicitly account for environmental variations while incorporating prior chemical knowledge into their structure. They are designed to handle cases with missing or partially observed component concentrations while providing principled uncertainty quantification through Bayesian inference. This combination of flexible probabilistic modelling with chemical domain knowledge enables more robust and interpretable analysis.

These research directions are motivated primarily by their application to pharmaceutical manufacturing, and are designed with the constraints imposed by this setting. However, the techniques developed can be applied beyond just that use case and in all of the chapters some additional use cases for the techniques are discussed. These additional use cases are chosen to illustrate the strengths and weaknesses of our proposed approaches.

1.2 Contributions

This thesis builds on two existing techniques, Partial Least Squares (PLS) and Gaussian Process Latent Variable Models (GPLVMs). In Chapter 2 we provide the background for understanding both of these techniques.

In Chapter 3 we describe a novel approach for calculating the uncertainty from a PLS model

based on the bootstrap technique. The approach that we develop improves upon the standard linear approaches in including the non-linear interactions between noise in the training data the parameters of the model. We apply this technique to various industrially relevant case studies, illustrating the differences between the predictions arising from our method and different existing methods.

In Chapter 4 we present a novel approach for tackling the problem of spectroscopy by building on the GPLVM technique. Specifically we consider adapting the common Classical Least Squares (CLS) model to the case where signals can vary from one data point to another. We allow the changes from one observation to another by modelling each pure component spectra as a GPLVM. We then demonstrate our approach on a variety of real-world applications.

In Chapter 5 we present further development of the work in Chapter 4. Specifically, we present improvements which simplify the optimization procedure and improve the accuracy of results, provide a method for working on data sets where not all of the signal weights are observed in the training data, and suggest an approach for improving the uncertainty estimation from our method based on Structured Variational Inference.

1.3 Notation

In this thesis a consistent notation was used when possible. We provide a table of notation which for each chapter separately in the Appendix.

1.4 Statement of Originality

The ideas presented in this thesis are the author's own work, unless stated otherwise. All concepts which are not original are properly cited and credited.

1.5 Publications

The work presented in Chapter 3 resulted in the publication Odgers et al. [2023] in the American Institute of Chemical Engineering journal.

The work in Chapter 4 is currently under review and available as a preprint [Odgers et al., 2024].

Additional work carried out as part of the same grant resulted in the publication Kappatou et al. [2023], to which the author of this thesis also contributed and was a co-author.

The author of this work also contributed to the publication Folch et al. [2023] which was published in the NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World in 2023.

Chapter 2

Background

2.1 Introduction

This chapter aims to provide the necessary background required for this thesis. In Section 2.2 we provide background on Partial Least Squares (PLS) which is required for Chapter 3. In Section 2.2.1 we explain the NIPALS algorithm which is used to make predictions for PLS, and in Section 2.2.2 we give some of the existing theory regarding the predictions arising from PLS.

Following this we provide background for Chapter 4 and 5. The approach taken relies on Gaussian Processes (GPs), and more specifically on Gaussian Process Latent Variable Models. We therefore provide a background on both of these techniques.

2.2 Background for Partial Least Squares

PLS is a widely used dual purpose technique where the goal of the technique is both to make accurate predictions for a model and provide an interpretable set of relationships between the input and output variations [Wold, 1975, Wold et al., 2001, Vinzi et al., 2010]. PLS links an input space ($x \in \mathbb{R}^M$) to an output space ($y \in \mathbb{R}^C$) via a low dimensional latent space ($t \in \mathbb{R}^A$). PLS finds this lower dimensional latent space, in a way which can both predict

the output space and describe a significant portion of the variance of the input space. This latent space allows PLS to perform particularly well when there are large numbers of highly correlated input variables, allowing the construction of an informative, lower dimensional latent space with $A \ll M$s.

There are many different variants of the PLS technique. The original model was introduced by Wold [1975], but there are many variants including SIMPLS [De Jong, 1993], Orthogonal-PLS [Trygg and Wold, 2002], kernel PLS [Rosipal and Trejo, 2001], tri-PLS [Bro, 1996], OUPLS [Puwakkatiya-Kankanamage et al., 2014] to name just a few. The version which we will use throughout this thesis is the original version introduced by Wold [1975], although the description is modified slightly to the more modern version which has fewer variables but makes identical predictions.

The PLS algorithm requires a training data set containing N observations, where the inputs and outputs are described by the matrices $X = (x_1, \ldots, x_N)^T \in \mathbb{R}^{N \times M}$ and $Y = (y_1, \ldots, y_N)^T \in \mathbb{R}^{N \times C}$. These two matrices are then described in terms of a matrix of latent variables $T = (t_1, \ldots, t_N)^T \in \mathbb{R}^{N \times A}$, where $A \ll M$. The latent variables are related to the original variables by a set of linear mappings $P \in \mathbb{R}^{M \times A}$ and $Q \in \mathbb{R}^{C \times A}$ which, along with the means $\bar{x} \in \mathbb{R}^M$ and $\bar{y} \in \mathbb{R}^C$, can be used to describe the observed data as

$$X = \mathbf{1_N}\bar{x}^T + TP^T + E, \tag{2.1}$$

and

$$Y = \mathbf{1_N}\bar{y}^T + TQ^T + F, \tag{2.2}$$

where $\mathbf{1_N} \in \mathbb{R}^{\mathbf{N}}$ is a N-dimensional vector of ones, $E = (e_1, \ldots, e_N)$ and $F = (f_1, \ldots, f_N)$ are the residuals, and $\bar{x} \in \mathbb{R}^M$ and $\bar{y} \in \mathbb{R}^C$ are the empirical means given by

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2.3}$$

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i. \tag{2.4}$$

This description of PLS highlights the similarities to other linear latent variable techniques, such as Canonical Correlation Analysis (CCA) [Hotelling, 1936] or Principal Component Regression (PCR) [Massy, 1965]. There are strong relationships between CCA, PCR and PLS [Burnham et al., 1996], which can intuitively be understood by considering the objective by which the subspace which X is projected to is found. CCA seeks to find the directions of maximum correlation between the input and output spaces, this can result in finding projections of the data which are not predictive of the output. PCR relies on Principle Component Analysis (PCA) to find the subspace which maximises the described variance of the input data without reference to the output data, which can result in ignoring signals in the input space if the size of the signal associated with the latent variable which connects to the output space is smaller than variances which are only relevant for describing the inputs space. PLS seeks to balance these two approaches using a method which is sometimes described as *maximising the covariance*, balancing the goal of finding latent variables which are descriptive of the output variable and finding directions for projection with large variance in the input space. Although this description is intuitively helpful to describe the goal of PLS, it is somewhat misleading. The parameters of PLS are not described by a single mathematical objective, instead they are calculated using the Non-linear Iterative PArtial Least Squares (NIPALS) algorithm, which deterministically calculates values for P, Q, and another crucial parameter W which relates the input to the latent space. The description of maximising the covariance applies to the parameter W, not to P, and even with this caveat the description misses an underlying complexity of the system in which new covariances are added to the data as the algorithm progresses. The NIPALS algorithm, as well as the subtlety of the covariance objective is discussed further in Section 2.2.1.

Another useful method for analysing PLS is by only considering the predictions which arise from it, in other words considering PLS as a pure regression model [Wold et al., 1984, Helland, 1988, Elden, 2004]. This approach considers linear predictors directly from the input to the output which is useful for analysing uncertainties from the predictions [Denham, 1997]. This approach is described in Section 2.2.2.

2.2.1 NIPALS Algorithm

As mentioned above, the description of PLS is not given by a single objective function, instead an algorithm is used to find the relevant parameters. In this section we first describe the algorithm to find the parameters, then we describe how to use the NIPALS algorithm to make predictions, finally we discuss why two different relationships between the input space and the latent space are preferable in the PLS algorithm.

Finding the Parameters Using the NIPALS algorithm

NIPALS is an iterative approach, in which the algorithm alternates between maximising covariance and deflating the input data until A latent variables have been found. Prior to starting this iterative procedure the data matrices X and Y are mean centred. The empirical means are calculated using Equations (2.3) and (2.4). Using these empirical means the training data matrices are mean centred with

$$X_1 = X - \mathbf{1}_N \bar{x}^T \tag{2.5}$$

$$\tilde{Y} = Y - \mathbf{1}_N \bar{y}^T. \tag{2.6}$$

Note that the algorithm requires updating the input at each iteration so needs an index a, which is set to be one in the equation above.

Following this preprocessing, for each $a = 1, \ldots, A$ the following three steps are carried out:

1. Calculate $w_a \in \mathbb{R}^M$, which is the projection of the input data, *i.e.* $X_a w_a$, which has maximum covariance with Y. This can be solved using the following maximisation problem:

$$\max_{w_a} \quad w_a^T X_a^T \tilde{Y} \tilde{Y}^T X_a w_a \quad s.t. \quad ||w_a|| = 1, \tag{2.7}$$

   This is equivalent to finding the first left singular vector of $X_a^T \tilde{Y} \tilde{Y}^T X_a$.

2. Calculate the scores using

$$t_a = X_a w_a. \tag{2.8}$$

3. Calculate the ordinary least squares regressors ($p \in \mathbb{R}^{1\times M}$ and $q \in \mathbb{R}^{1\times C}$)for X_a and $\tilde{Y}$ given the latent variable $t_a = [t_{1,l}, \ldots, t_{N,l}]$ using

$$p_a = \frac{t_a^T X_a}{t_a^T t_a} \tag{2.9}$$

$$q_a = \frac{t_a^T \tilde{Y}}{t_a^T t_a}, \tag{2.10}$$

where $t_a \in \mathbb{R}^N$ is a matrix of the ath latent variable of the training data.

An important property of PLS for this step is that the latent variables have zero covariance between different values for l Gustafsson [2001], so the matrix $T^T T$ containing all latent variables is diagonal. Due to this lack of covariance the least squares estimator of Q can be estimated iteratively in this way.

4. Finally, the input data is deflated to produce the input for the next iteration:

$$X_{a+1} = X_a - t_a p_a^T \tag{2.11}$$

After this iterative procedure is finished the full matrices W, P, and Q are found by concatenating w_a, p_a and q_a

$$W = [w_1, w_2 \ldots, w_A] \qquad P = [p_1^T, p_2^T \ldots, p_A^T]^T \qquad Q = [q_1^T, q_2^T \ldots, q_A^T]^T. \tag{2.12}$$

Predicting a Response Given an Input Value and PLS parameters

Given this set of parameters PLS is able to predict the response (y) from a new input ($x \in \mathbb{R}^M$). Here the estimate for $y \in \mathbb{R}^C$ is denoted using $\hat{y}(x; W, P, Q)$, and is calculated in the following way:

1. Set $a = 1$ and $x_1 = x - \bar{x}$.

2. Calculate the scalar t_l using

$$t_a = w_a^T x_a. \tag{2.13}$$

3. If $a < A$ transform the input variable using

$$x_{a+1} = x_a - t_a p_a^T, \tag{2.14}$$

set $l = l + 1$ and return to Step 2.

4. Calculate the value for y using

$$\hat{y}(x; W, P, Q) = \bar{y} + Qt, \tag{2.15}$$

where $t = [t_1, t_2, \ldots t_A]^T \in \mathbb{R}^A$.

Importance of the Deflation Step

The most surprising step of the PLS is the deflation step. The approach is very reminiscent of delation as a method in Principle Component Analysis (PCA), where the first principle component is found and then removed using Hotelling's deflation method. This is clearly explained in Mackey [2008], amongst other places. However, for PCA deflation method only a single vector is used for deflation which would be equivalent to using w_a instead of p_a in Equation (2.11). Unlike other latent variable models the deflation step requires two different vectors $w_a, p_a \in \mathbb{R}^M$ for each latent variable, one for mapping to the latent space and one for mapping back to the observed space. The benefit of using this approach is that it allows more than C latent variables to be found, despite the fact that $X_1^T \tilde{Y} \tilde{Y}^T X_1$ has at most C non-zero eigenvalues. If the approach used only one vector then this would be equivalent to applying PCA to the matrix $X_1^T \tilde{Y} \tilde{Y}^T X_1$, would give at most the C non-zero meaningful projections of the data, however this can be avoided by deflating using a new loading p_a.

This can be illustrated by considering the $C = 1$ case, a common occurrence when only one component is of interest. For this $X_1^T \tilde{Y} \tilde{Y}^T X_1$ has a single non-zero eigenvalue, σ_1, associated with the eigenvector w. The standard procedure for deflation would leave this with no non-zero

eigenvalues. The deflation for PLS is carried out by

$$X_2 = X_1 - t_1 p_1^T = X_1 - X_1 w_1 p_1^T = X_1(\mathbb{I}_M - w_1 p_1^T). \tag{2.16}$$

As a result for the next iteration, where $a = 2$, the matrix for which the maximum eigenvalue needs to be found is

$$\begin{aligned} X_2^T \tilde{Y}\tilde{Y}^T X_2 &= (\mathbb{I}_M - w_1 p_1^T)^T X_1^T \tilde{Y}\tilde{Y}^T X_1 (\mathbb{I}_M - w_1 p_1^T) \\ &= X_1^T \tilde{Y}\tilde{Y}^T X_1 - p_1 w_1^T X_1^T \tilde{Y}\tilde{Y} X_1 - X_1^T \tilde{Y}\tilde{Y}^T X_1 w_1 p_1^T + p_1 w_1^T X_1^T \tilde{Y}\tilde{Y}^T X_1 w_1 p_1^T \\ &= X_1^T \tilde{Y}\tilde{Y}^T X_1 - \sigma_1 p_1 w_1^T - \sigma_1 w_1 p_1^T + \sigma_1 p_1 p_1^T, \end{aligned} \tag{2.17}$$

which has a new non-zero eigenvalue, σ_2, associated with the eigenvector $(p_1 - w_1)$. Showing this requires the results

$$(p_1 - w_1)^T w_1 = 0. \tag{2.18}$$

$$X_1^T \tilde{Y}\tilde{Y}^T X_1 (p_1 - w_1) = 0. \tag{2.19}$$

The first result can be found by noting that $p_1^T w = 1$ and the second can be found by using the fact that $(p_1 - w_1)$ is orthogonal to w_1, and the eigenvectors of $X_1^T \tilde{Y}\tilde{Y}^T X_1$ are orthogonal with only one non-zero values associated with the eigenvector w_1. Substituting these results into the expression in Equation (2.17) gives

$$X_2^T \tilde{Y}\tilde{Y}^T X_2 (p_1 - w_1) = \sigma_1 p_1^T (p_1 - w_1)(p_1 - w_1). \tag{2.20}$$

Similar arguments can be applied iteratively meaning that the matrix $X_a^T \tilde{Y}\tilde{Y}^T X_a$ has non-zero eigenvalues as long as $p_a \neq w_a$. This condition is satisfied as long as the eigenvector of $X_a^T \tilde{Y}\tilde{Y}^T X_a$ does not exactly align with the eigenvector of $X_a^T X_a$.

For the case where $C > 1$ these relationships are more complex, specifically due to the relationship in Equation (2.19) no longer being true, but the fact that new covariances are induced in $X_a^T \tilde{Y}\tilde{Y}^T X_a$ at each value of a is still true. The goal of this example is to provide an intuitive understanding of how incorporating two different relationships between the input and the latent

space allows the advantages of covariance, selecting components in the input space with both a high variance and large contribution to the predictions of the output, while avoiding the issue of limiting the number of linear subspaces of X which can be used in the prediction.

2.2.2 Equivalent Linear Predictors to PLS

While an optimization approach with constraints is not straightforward as a method to understand the paraments of PLS [Burnham et al., 1996], it is possible to construct a linear map which produces the same predictions as the PLS model. The equivalent linear predictor to PLS relies on the additional relationship between the input and latent variables [Helland, 1988] of

$$T = XW(PW^T)^{-1}, \tag{2.21}$$

hence predictions are

$$\hat{Y} = XW(PW^T)^{-1}Q^T. \tag{2.22}$$

We define $\beta_{PLS} = Q(W^TP)^{-1}W^T$ as the linear map which produces equivalent predictions to the PLS algorithm.

This problem has been studied with special focus on the case where $C = 1$, in which it has been shown that the estimator β_{PLS} is the least squares estimate projected into the span of the Krylov subspace $K_A(X_1^TX_1, X_1^T\tilde{Y}) = span(X_1^T\tilde{Y}, (X_1^TX_1)X_1^T\tilde{Y}, \ldots, (X_1^TX_1)^{A-1}X_1^T\tilde{Y})$ [Wold et al., 1984, Helland, 1988, Elden, 2004]. This linear equivalent for predictions in the $C = 1$ case is widely used in prior literature on the uncertainty estimation of PLS [Phatak et al., 1993, Faber and Kowalski, 1996, 1997, Faber, 2002, Olivieri et al., 2006]. It is important to emphasise that this approach discards much of the interpretability of PLS [Burnham et al., 1996] which is highly relevant for why practitioners use PLS and is limited to the case where $C = 1$.

2.2.3 Partial Least Squares in this Thesis

This thesis addresses uncertainty quantification in PLS. While PLS's latent variable structure provides effective dimensionality reduction, the non-linear relationship between model parameters and predictions complicates uncertainty estimation. In Chapter 3, we develop a bootstrap-based approach that naturally captures these non-linearities without requiring problematic linear approximations. This method is particularly effective for Design Space identification where accurate uncertainty bounds are essential for regulatory compliance and manufacturing reliability.

2.3 Gaussian Processes

2.3.1 Introduction

Gaussian Processes (GPs) are a non-parametric Bayesian Regression technique [Rasmussen, 2003] . The simplest set up for this assumes that there is an observed training dataset $\mathcal{D}_{GP}$ containing N input variables $X = \{x_i \in \mathbb{R}^A | i = 1, \ldots, N\}$ and corresponding output variables $y = \{y_i \in \mathbb{R} | i = 1, \ldots, N\}$. These input-output pairs are linked via some unknown function $f(\cdot)$ with some noise which is typically treated as being additive and homoskedastic $\epsilon_i \sim N(0, \sigma)$. This allows the relation between the input and output data to be expressed as

$$y_i = f(x_i) + \epsilon_i. \tag{2.23}$$

GPs approach the challenge of learning $f(\cdot)$ in a non-parametric Bayesian way. For the purpose of this thesis, we are interested in specifically estimating the unobserved function values at the input locations, *i.e.* the function values before noise is added. This is done by treating the function values $f = (f(x_1), \ldots, f(x_N)) \in \mathbb{R}^N$ as random variables and placing a prior on them,

$$f \sim N(0, K_{XX}), \tag{2.24}$$

where $K_{XX} \in \mathbb{R}^{N \times N}$ is the kernel matrix. With this prior specified the posterior distribution can be found to using

$$p(f|\mathcal{D}_{GP}) = \frac{p(\mathcal{D}_{GP}|f)p(f)}{p(\mathcal{D}_{GP})}. \tag{2.25}$$

Provided that the additive noise is Gaussian, this has a closed form

$$p(f|\mathcal{D}_{GP}) = N(\mu^f, \Sigma^f), \tag{2.26}$$

where

$$\mu^f = K_{XX}\left(K_{XX} + \sigma^2 \mathbb{I}_N\right)^{-1} y \quad \text{and} \quad \Sigma^f = K_{XX} - K_{XX}\left(K_{XX} - \sigma^2 \mathbb{I}_N\right) K_{XX}. \tag{2.27}$$

The key to the effectiveness of GPs is their ability to concentrate their probability mass to a small number of possible functions. As non-parametric models GPs are capable of modelling highly complex relationships between the input and the output. However, highly complex relationships can suffer from poor generalisation due to their inability to rule out unlikely models. GPs strike a balance between simple models which generalise well and flexible models which fit the data through the selection of a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that

$$[K_{XX}]_{ij} = k(x_i, x_j). \tag{2.28}$$

This kernel function is required to produce a positive semi-definite kernel matrix [Rasmussen, 2003]. In this thesis we are interested in functions $f(\cdot)$ which vary smoothly over an input $x_i \in \mathbb{R}^A$. We do this by using a squared exponential kernel

$$k(x_i, x_j) = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{a=1}^{A} \frac{(x_{ia} - x_{ja})^2}{\beta_a}\right). \tag{2.29}$$

This kernel function requires defining a scale hyper-parameter σ_f^2 and a set of length-scales $\beta = \{\beta_a | a = 1, \ldots, A\}$. This can be done in a principled, automatic way by finding the model which balances the complexity of the model with the fit of the data. This is achieved by finding

the hyper-parameters which maximize the marginal likelihood of the observed data:

$$\log p(Y|X) = -\frac{1}{2}y^T(K_{XX} + \sigma^2\mathbb{I}_N)^{-1}y - \frac{1}{2}\log(\det(K_{XX} + \sigma^2\mathbb{I}_N)) - \frac{N}{2}\log(2\pi). \tag{2.30}$$

2.3.2 Gaussian Process Latent Variable Models

The work in this thesis builds on Gaussian Process Latent Variable Models (GPLVMs), which are a powerful method for dimensionality reduction that combines probabilistic modeling with non-linear mappings. GPLVMs invert the typical GP regression setup by treating the input locations X as unobserved latent variables that need to be inferred, while only observing the high-dimensional output data Y. Specifically, given a dataset of N observations where each observation is M-dimensional, $\mathcal{D}_{GPLVM} = \{y_j \in \mathbb{R}^N | j = 1, \ldots, M\}$, GPLVMs aim to find both a low-dimensional latent representation $X = \{x_i \in \mathbb{R}^A | i = 1, \ldots, N\}$ for each data point and a set of mappings from the latent space to the observed output space $F = \{f_j \in \mathbb{R}^N | j = 1, \ldots, M\}$ where $f_m = (f_m(x_1), ..., f_m(x_N)) \in \mathbb{R}^N$. The standard assumption for this is to use a spherical normal prior so $p(X) = \prod_{i=1}^N N(0, \mathbb{I}_A)$ and use an independent GP mapping sharing the same kernel for each of the output dimensions so $p(F|X) = \prod_{i=1}^M N(0, K_{XX})$.

GPLVMs are an attractive model to use as the objective involves integrating out the possible function values, providing implicit regularization for the mapping. Ideally, we would find the posterior distribution for the unobserved variables using

$$p(X, F|Y) = \frac{p(Y|F)p(F|X)p(X)}{p(Y)} \tag{2.31}$$

and the GP hyper parameters could be found by maximising the log-marginal-likelihood ($\log(p(Y))$). Unfortunately, this is not possible as calculating $p(Y) = \int p(Y|F)p(F|X)p(X)dXdF$ requires an intractable integral. This is because the expression for the conditional distribution for the function values which appears in the integral is given by

$$p(F|X) \propto \prod_{j=1}^{M} \exp(-\frac{1}{2}f_j^T K_{XX}^{-1} f_j), \tag{2.32}$$

which requires the inverse of the covariance matrix. This inverse covariance matrix results in a non-tractable integral.

This problem was first addressed in Lawrence [2005], where the uncertainty of the position of the inputs x_i was ignored. By ignoring the uncertainty the need to perform the intractable integral was avoided. Both the estimate for the position of the latent variables and the GP is given by the type-II MLE estimate. Concretely, the objective to be optimized is

$$\log\left(p(Y|X)\right) = -\frac{M}{2}\log(\det(K_{XX}+\sigma^2\mathbb{I}_N)) - \frac{N(M+A)}{2}\log(2\pi) - \sum_{j=1}^{M}\frac{1}{2}y_j^T(K_{XX}+\sigma^2\mathbb{I}_N)^{-1}y_j - \sum_{i=1}^{N}x_i^T x_i. \quad (2.33)$$

and the variables to be optimized are the input values X and the kernel hyper parameters.

While it is possible to work with this approach, the maximum likelihood estimate for the latent variables can make is susceptible to overfitting. In the next section we describe a method to achieve a better estimate for the posterior.

2.3.3 Variational Inference for Gaussian Process Latent Variable Models

Variational Inference (VI) is a method in machine learning and statistics that approximates complex probability distributions by finding a simpler distribution that approximates the target [Blei et al., 2017]. Rather than trying to compute exact posterior distributions, which is often intractable for complex models, VI turns the inference problem into an optimization problem. It does this by defining a family of simpler distributions (called variational distributions) $q(\cdot)$ and then finding the member of that family that minimizes the Kullback-Leibler (KL) divergence with respect to the true posterior distribution $p(\cdot|\cdot)$. The objective of this optimization which minimises the KL divergence between the true posterior and the variational posterior is the Evidence Lower BOund (ELBO).

Variational Inference was first applied to GPs by Titsias [2009] who studied the problem de-

scribed in Section 2.3.1 to produce Sparse GPs. The key insight which allows VI to be applied effectively is that as GPs additional points not associated with observations can be included and, provided the distribution of values at those points is drawn from the same GP prior, the prior on the observed points remains the same. These new data points are referred to as inducing points $u = (u_1, \ldots, u_L) \in \mathbb{R}^L$ corresponding to input locations $V \in \mathbb{R}^{L \times A}$ and are drawn from the same GP as the function values, *i.e.*

$$(u_1, \ldots, u_L, f_1, \ldots, f_N) \sim N(0, K_{(XV)(XV)}) \quad \text{where} \quad K_{(XV)(XV)} = \begin{pmatrix} K_{VV} & K_{XV} \\ K_{VX} & K_{XX} \end{pmatrix}. \tag{2.34}$$

As the inducing point output values u and the function values f are drawn from the same process it is possible to write down the conditional distribution for the function values at the location of the observations given the inducing points using

$$p(f|u, X) = N(\mu^{f|u}, \Sigma^{f|u}) \tag{2.35}$$

where

$$p(f|u) = \mathcal{N}(f|\mu^{f|u}, \Sigma^{f|u}) \tag{2.36}$$

where the mean and covariance are given by

$$\mu^{f|u} = K_{XV} K_{VV}^{-1} u \quad \text{and} \quad \Sigma^{f|u} = K_{XX} - K_{XV} K_{VV}^{-1} K_{VX} \tag{2.37}$$

and therefore the full posterior for the inducing points and function values can be written

$$p(f, u|y) = \frac{p(y|f, X)p(f|u)p(u)}{p(X, Y)}, \tag{2.38}$$

and is well defined.

Titsias [2009] noted that the conditional distribution for the function values given the inducing variables could be used in the variational posterior by choosing a variational distribution of the form $q(f, u) = p(f|u)q(u)$, where $q(u) = N(\mu^u, \Sigma^u)$. In this setting the variational posterior only

needs to be defined over the inducing points, from which the posterior of the other variational posterior function values can be calculated. Using this variational family the ELBO is

$$\begin{aligned}\mathcal{L}_{GP} &= \int \log\left(\frac{p(y|f)\cancel{p(f|u,X)}q(u)}{\cancel{p(f|u,x)}q(u)}\right) p(f|u,X)q(u)dudf \\ &= \langle p(y|f)\rangle_{q(f,u)} - D_{KL}(q(u)||p(u)),\end{aligned} \tag{2.39}$$

which has a closed form solution for a Gaussian likelihood. Note that we have introduced a piece of notation here $\langle\cdot\rangle_{q(\cdot)}$, which is the expectation of the expression inside the angular brackets with respect to the distribution $q(\cdot)$.

In addition to the computational advantages in the traditional GP setting, the introduction of inducing points also makes it possible to introduce a variational family for the position of the points in the latent space in the GPLVM setting [Titsias and Lawrence, 2010]. In the original GPLVM paper, whose objective is given in Equation (2.33), the inverse kernel matrix is associated with the unknown positions of the points in latent space. In the sparse formulation this expression no longer appears in the expression for $\mathcal{L}_{GP}$. Instead the term is $p(f|u,X)$ which, as can be seen in Equation (2.37), does not require the use of the inverse of the kernel matrix of the data points (K_{XX}^{-1}), instead the kernel matrices which appear are K_{XV} and K_{VV}^{-1}. The inverse matrix now appears in the kernel matrix for the input locations of the inducing points (K_{VV}^{-1}). The inducing point locations are known exactly and can be selected arbitrarily without impacting the prior of the system, thereby avoiding the intractable term of the integral.

By introducing inducing points to the setting described in Section 2.3.2 requires a set of inducing points for each of the output dimensions, so the inducing point outputs are given by $U = \{u_j \in \mathbb{R}^L | j = 1,\ldots,M\}$. The inputs to the inducing points V are assumed to be shared across all of the outputs dimensions. With this the full posterior we are interested in is $p(F,U,X|Y)$, which is approximated by optimizing the ELBO

$$\mathcal{L}_{GPLVM} = \sum_{j=1}^{M}\left[\langle p(y_j|f_j)\rangle_{q(f_j,u_j,x)} - D_{KL}(q(u_j)||p(u_j))\right] - \sum_{j=1}^{M} D_{KL}(q(x_i)||p(x_i)). \tag{2.40}$$

For the Automatic Relevance Determination (ARD) kernel we are interested in for this thesis

and the mean field variational family $q(X) = \prod_{i=1}^{N} N(x_i|\mu_i^x, \Sigma_i^x)$, with Σ_i^x being diagonal, then the expectation $\langle p(y_j|f_j)\rangle_{q(f_j,u_j,x)}$ can be calculated analytically and the optimal distribution $q(u_j)$ can be calculated analytically for each of the M output dimensions.

2.3.4 Gaussian Process Latent Variable Models in this Thesis

This thesis extends GPLVMs to work in the signal separation setting implied by Beer-Lambert's law. The motivation for the GP prior provides crucial regularization through kernel hyper-parameter optimization, automatically determining the complexity and smoothness of spectral variations without requiring explicit parametric constraints. In Chapter 4 we demonstrate the practicability of this approach, and in Chapter 5 we provide multiple extensions making the model more practical.

Chapter 3

Uncertainty estimation in Partial Least Squares with Bootstrap

This chapter is associated with the publication Odgers et al. [2023]. During the course of producing this work I collaborated closely with Dr. Chryssa Kappatou and Dr. Salvador García-Muñoz who helped with various comments on the writing and figures.

3.1 Introduction

3.1.1 Working Between Communities

What makes a good model? It is easy to write down a wish-list of attributes which make a good model. For example, everyone would typically agree that they want a model to be accurate, is easy to interpret, computationally cheap, and it provide provable statistical guarantees. These are all desirable qualities, but which one is the most desirable and how much of one are you willing to give up to improve another? This is obviously not a question which has a simple answer. Which trade-offs are reasonable to make will depend entirely on what problem which you are trying to solve: the criteria for a good model to deciding which online advert to show

someone is very different than the criteria to decide whether a medicine is safe and effective for human consumption.

This more philosophical question is often unhelpful, leading to debates which take more time than the degree of clarity they provide justifies. However, there are some cases where it is necessary to consider which attributes of a model are important—one of which is when a piece of work falls between two communities. Different communities can arrive at different answer to the question which desirable features of a model are important and which can be ignored. The answer to this question is often left implicit by communities and, as many interesting pieces of research lie on the boundaries of communities, it can be very unclear which criteria should be used to judge a model.

This project takes the perspective of the pharmaceutical manufacturing community. A very attractive feature of this community is that it has a clear goal: *how does this model help produce the medicines needed to save lives?* One of the key differences between this and more theoretically motivated communities is that this problem is very large. As an example, In my PhD I was lucky enough to collaborate with Eli Lilly who, at the time of writing, employ nearly 50 thousand people. Obviously not all of these people are working on the manufacturing problem, but, even with the relatively small fraction of people who are focused on this area, a major consideration becomes how to ensure that everyone shares an understanding of how a model works. This is especially important for this safety critical application as every model needs to be approved, so any change in the model also requires educating regulators in several countries. Given this the required level of improvement to justify changing a model is very significant and the most valuable contributions which can be made stay close to existing methods.

This is the context in which this work takes place. The challenge for this work was that the objectives of the pharmaceutical manufacturing community are not the same as those of more theoretically motivated communities. The use of the PLS model is ubiquitous and the role this Chapter is not to suggest alternative models, which may be easier to justify to statisticians, it is to expand the possible uses for this practical and accurate model.

3.1.2 Uncertainty in PLS

This chapter develops uncertainty estimates for PLS. The challenge of this approach is that, as stated in Section 2.2.1, the PLS model, is not motivated by a single statistical objective but instead described by an algorithm. Each step of the algorithm is theoretically motivated, but taken together it is hard to analyse. In this Chapter we restrict ourselves to work precisely with the standard PLS model with no modifications. We have written our approach in such a way that we hope it is understandable to both statisticians and practitioners who use the model and so have written the Chapter as if PLS was a probabilistic model, despite the fact that it is a hard model to fit into the probabalistic framework. For the algorithmic approach, we refer to Section 2.2.1.

There are many reasons that the predictions from a model will not match up with the true values, for example there will be noise in the quantities being used for the training data, the PLS prediction may be a linear approximation for a system which is not exactly linear, or the system being predicted may contain a certain inherent amount of unavoidable variation. PLS, and the uncertainty prediction method that we propose to build for it, does not differentiate between these different sources of variability. Our approach incorporates all of these real sources of uncertainty into two categories. The first one is a random, unpredictable variation that we describe as a Gaussian noise on the output. The other source is uncertainty in the PLS parameters, which arises because the training data is finite. As there are a finite number of data points, a number of different parameters of PLS could match the training data quite well and could be capturing the underlying relationship between inputs and outputs more accurately. The novelty of our method is in how to approach generating a distribution for the PLS parameters.

Calculating prediction uncertainty in PLS is non-trivial because of a complicated relationship between noise in the output and the predictions coming from PLS [Denham, 1997]. We discuss some of the many existing approaches to overcome this difficulty later in this chapter. Much of the previous work was done on PLS in the 1980s and 1990s [Helland, 1988, Denham, 1997, Höskuldsson, 1988, Faber and Kowalski, 1997], when limited computational power was available.

This required the use of approximations with known limitations, but that were computationally quick. We use the massive increase in computational power to apply a Monte Carlo Bootstrap method, as described by Fushiki [2005], to PLS. By creating synthetic data sets by sampling with replacement, our method allows the non-linearities in the PLS parameters to be included in any prediction.

Our method produces a continuous distribution that is free from the approximation that the change in the prediction of PLS can be modelled using linear expressions. Instead of assuming linearity, we allow variation in the parameters of PLS to be approximated by variation in the data used to train the PLS model. Explicitly, the assumptions of the method that we propose below are that the noise in the prediction error can be modelled by a normal distribution, that PLS is a reasonable predictive model for the data of interest, and that the training data is drawn from a similar distribution to the data for which the model will be used to make future predictions.

This chapter first overviews the different existing methods for uncertainty prediction for PLS. Next, we develop a novel bootstrapping method to provide a probabilistic prediction density for PLS. Finally, we demonstrate the applicability of our approach in five case studies. The first three of these case studies focus on Design Space identification [ICH Expert Working Group, 2009], i.e., identifying process parameters which are likely to result in a desired output. The final two case studies focus on generating prediction intervals for real data, with one case study performing similarly to existing techniques and one other performing in a significantly different way. Finally, the main outcomes of these case studies are summarised and potential avenues of further research are highlighted.

The parameters which we are interested in are the PLS parameters introduced in Section 2.2. Recall that, for a single point, the relationships between the space ($x \in \mathbb{R}^M$), the output space ($y \in \mathbb{R}^C$) and the latent space ($t \in \mathbb{R}^A$) can be expressed as

$$x = \bar{x} + Pt + e \tag{3.1}$$

$$y = \bar{y} + Qt + f, \tag{3.2}$$

and

$$t = (W^T P)^{-1} W^T x. \tag{3.3}$$

3.2 Existing Uncertainty Techniques in PLS

The focus of this chapter is calculating prediction intervals for PLS conditional on the inputs. The standard method for making predictions with uncertain parameters is to find a distribution that describes the parameters, and then to marginalise out the uncertain parameters. This is achieved by making a prediction using

$$p(y|x, \mathcal{D}) = \int p(y|x, \omega) p(\omega|\mathcal{D}) d\omega, \tag{3.4}$$

where $x \in \mathbb{R}^M$ is the input, $y \in R^C$ is the output that we are trying to predict, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ is a set of training data, and ω are a set of model parameters needed for prediction - for PLS model with normally distributed noise $\omega = \{W, P, Q, \sigma^2\}$, where W, P and Q are the parameters of PLS and σ^2 is an estimate of the variance of the noise.

In many situations Bayesian methods can be used to find the distribution of parameters. This combines prior information, $p(\omega)$, with the likelihood of a set of parameters leading to the training data, $p(\mathcal{D}|\omega)$, to find a distribution known as the posterior for the parameters ω. In order to maintain normalisation, the product of the likelihood and prior needs to be divided by a constant, $p(\mathcal{D})$, which is known as the evidence. The expression for the posterior is therefore

$$p(\omega|\mathcal{D}) = \frac{p(\mathcal{D}|\omega) p(\omega)}{p(\mathcal{D})}. \tag{3.5}$$

Multiple papers have applied Bayesian techniques to data-sets with low dimensional structures, such as those that PLS excels at. This has lead to a technique referred to as Probabilistic PLS [Nounou et al., 2002, Chen et al., 2007a, Zheng et al., 2016, Li et al., 2011, el Bouhaddani et al.,

2018, del Castillo and Reis, 2020], where the data is assumed to be generated according to

$$x = P_{shared}t_{shared} + P_{shared}t_{private} + e \tag{3.6}$$

$$y = Q_{shared}t_{shared} + f, \tag{3.7}$$

and the latent variables are now factorised into a shared and private set, with certain latent variables only describing the input variables. This structure of problem can be solved using standard Bayesian estimation techniques and allows researchers to tackle many types of problems. However, this formulation does not provide the same solution as PLS using the NIPALS algorithm. Many practitioners will select PLS for good reasons, such as interpretability, familiarity, memory efficiency and accuracy of the predictions, and it is not reasonable to expect a change in the model being used in order to provide a conditional uncertainty estimate. The work in this chapter aims to address this by providing a predictive distribution which can be found after the PLS model has been selected.

There are several challenges to applying Bayesian techniques to the standard NIPALS PLS. One is that there are two relationships between the input and latent spaces - W for mapping from the input to the latent space and P for mapping from the latent space to the input space, as is discussed in Section 2.2. When inferring t, it is not clear which of these should be preferred. The Probabilistic PLS papers described above avoid this problem by dropping W. Another challenge is that the parameters W and P are generated iteratively by column, with each column being found from fitting an estimate of a different data set which was generated during the NIPALS PLS algorithm. Given that only the first column of P and W are fitting the observed data it is not clear how to define the likelihood of the later columns of P and W. These problems are not easy to solve within a Bayesian framework.

Due to the challenges of applying Bayesian techniques to the complete set of parameters from PLS, a different approach has generally been preferred for calculating uncertainty. A different approach builds on the fact that there exists a linear estimator which will make the same

predictions as the PLS model, as described in Section 2.2.2, which has lead several researchers to investigate the possibility of treating PLS like other linear estimators [Faber and Kowalski, 1997, Phatak et al., 1993, Faber and Kowalski, 1996, Faber, 2002, Olivieri et al., 2006]. The key idea shared by all of these estimations is to make the simplification of the relationship between the equivalent linear predictor to PLS, β_{PLS}, to be linearly related to the added noise, typically focusing on the outputs. This type of model arises from considering a system with additive nose of the form $\epsilon \sim N(0, \Sigma)$, with the training data ($\mathcal{D} = \{X, Y\}_{i=1}^{N}$) being described by

$$Y = 1_N \bar{y}^T + X\beta_{true}^T + E. \tag{3.8}$$

Supposing that there exists a linear transform R, which when applied to the mean centered output $\tilde{Y} = Y - 1_N \bar{y}^T$ which gives an unbiased estimate for β_{true}. For example, an estimate form of estimate exists for Ordinary Least Squares for which has $R_{OLS} = ((X - 1_N \bar{x}^T)^T (X - 1_N \bar{x}^T))^{-1} (X - 1_N \bar{x}^T)^T$, giving an estimate $\hat{\beta} = R_{OLS} Y$. For any estimate of this type the errors of the linear predictor is distributed with covariance $S = R^T \Sigma R$ so

$$\beta_{true} - \hat{\beta} \sim N(0, S). \tag{3.9}$$

Under these assumptions the error of the $\hat{\beta}$, the mean and the noise would all be normally distributed with mean zero. If the covariance of the noise was known then the difference between the estimate $\hat{y}$ and the true value y for a given input x will be given by

$$y - \hat{y}|x \sim N(0, s(x)) \quad \text{where} \quad s(x) = \left(\Sigma + \frac{\Sigma}{N} + x^T S x \right). \tag{3.10}$$

For real data the variance of the noise is also estimated from the data the uncertainty from this source needs to be combined with the variance from the above expression. For a one dimensional output, which is the focus of much prior work [Zhang and García-Muñoz, 2009], this results in a student-t distribution. This allows us to write the prediction, conditional on

the input x and a training data set, as

$$p(y|x,\mathcal{D}) = \frac{1}{s(x)}\,\mathcal{T}\left(\frac{y-\hat{y}}{s(x)}; N-n_l\right), \tag{3.11}$$

where $\mathcal{T}(z,\nu)$ denotes the p.d.f. of a student distribution with ν degrees of freedom evaluated at z.

These methods rely heavily on the fact that an equivalent linear predictor to PLS exists, $\hat{\beta}_{PLS}$, as described in Section 2.2.2. This means that provided the errors in the linear predictors is approximated to be normally distributed then the output prediction will also be normally distributed. An issue for this approach is that the relationship between the training observations Y and the predictions from PLS is non-linear Helland [1988]. This means that distribution for the errors of $\hat{\beta}_{PLS}$ is not normally distributed, hence various approaches are possible to take for the best method of S to achieve a good approximation to the true error.

One method approximating the relationship between the noise in the output, while maintaining a remaining linear relationship between the noise and the parameters of the prediction, is to use a Taylor expansion of the relationship between the output of the training data Y and $\hat{\beta}_{PLS}$ [Phatak et al., 1993, Denham, 1997]. This idea was later extended by Faber and Kowalski [1996] to include perturbations in the input space as well, which PLS also has a non-linear relationship to [Elden, 2004].

One popular method is to assume that the span of the latent space in the input does not vary. This leads to an estimate of the covariance sometimes called the zeroth-order approximation [Phatak et al., 1993]. This allows $s(x)$ to be calculated as

$$s(x) = \sigma(1 + \frac{1}{N} + h(x)), \tag{3.12}$$

with

$$h(x) = \frac{t(x)^T(T^TT)^{-1}t(x)}{N-1} \quad \text{and} \quad \sigma^2 = \frac{1}{N-n_l}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2. \tag{3.13}$$

These expressions are used in combination with Equation (3.11) to generate a probabilistic

distribution to compare our results to.

Another method for estimating the covariance matrix for the linear transformation is to use bootstrap techniques. Faber [2002] investigated both bootstrap by residuals and bootstrap by objects performance in spectroscopy and preferred bootstrap by residuals for estimating S when using PLS for spectroscopy, noting the similarity of its performance to the zeroth order method. This chapter removes the assumption that the parametric uncertainty in the predictions from PLS can be modelled by a normal distribution, which the methods discussed in Faber [2002] do not address.

Other authors have also considered methods which do not assume that the prediction uncertainty from PLS can be modelled by a normal distribution. This was first examined by Denham [1997], who proposed a method to form predictive intervals for PLS prediction using bootstrap by residuals (see Efron and Tibshirani [1993] for a detailed description of bootstrap by residuals). In Denham's method, synthetic data sets are generated by adding random noise, following the the empirical distribution of the prediction errors, to the predictions from PLS. These synthetic data sets are then used to generate different sets of PLS parameters. For a new input, the set of parameters are used to calculate a a distribution of predictions from PLS, which can be convoluted with the empirical distribution of errors to generate confidence intervals. A similar method for estimating prediction intervals was also used by Reis and Saraiva [2012], but with the focus on situations with high dimensional outputs that contain a structure that is being found by PLS. One aspect not considered by these methods is the effect of variations in the input. If the inputs are of full rank and go directly to the outputs then simply considering the effects of perturbing the output considers all of the noise in the system and can lead to better estimates for the uncertainty [Efron and Tibshirani, 1993]. However, latent variable models, such as PLS, assume that only a portion of the input is of interest to predict the output and estimates which portion should be used from the data. As this is a quantity measured from the data it is important to consider that the wrong portion of the inputs has been selected. The intuition for this can be seen clearly in probabilistic PLS, where ignoring the uncertainty in P_{shared} and $P_{private}$ is clearly unjustified. In PLS the portion of the input which is selected depends on both the input and the output data, so by not considering possible changes in

the input an important source of uncertainty is missed by bootstrap by residuals, our method captures this by using bootstrap by pairs.

The approach proposed in this chapter builds on the approach proposed by Denham [1997]. Our work applies the method proposed by Fushiki et al. [2005], where the distribution for the parameters $p(\omega|\mathcal{D})$ used for making predictions is not estimated by the posterior. Instead the distribution is given by a set of Maximum Likelihood Estimates (MLEs) for different possible data sets that are approximated using the bootstrap method. Our method lends itself to the NIPALS algorithm as each parameter can individually be seen as a Maximum Likelihood Estimate [Gustafsson, 2001] although we are unaware of any result that indicates that the parameters of PLS are a joint maximum likelihood estimate. Our approach uses the regular NIPALS algorithm, does not assume any parametric form of uncertainty from the predictions of PLS, and considers all sources of uncertainty.

3.3 Method

Given a training data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, we wish to predict the output for a new observation x using a Partial Least Squares (PLS) regression model. More specifically, we want to construct a predictive distribution $p(y|x, \mathcal{D})$ taking into account the uncertainty in the estimation of the PLS model parameters due to finite training data. To do so we, use bootstrapping, a widely used technique for uncertainty quantification of statistical inference procedure. Efron and Gong [1983] present a good overview of bootstrapping.

Figure 3.1 depicts our approach to construct the predictive distribution $p(y|x, \mathcal{D})$ using bootstrapping. The key idea is that the distribution of the training data set is a good approximation of the distribution that would be observed in any future observations. Therefore, the parametric uncertainty of the values found in any parameters that arise from the data can be evaluated by examining the distribution estimated parameters which is generated by the distribution of training data. The first step of our method therefore consists of randomly simulating B new synthetic data sets of N data points, denoted by $\mathcal{D}^*_{(b)} = \{(x_i, y_i)^*_{(b)}\}_{i=1}^N$, $b = 1 \dots B$, by sampling

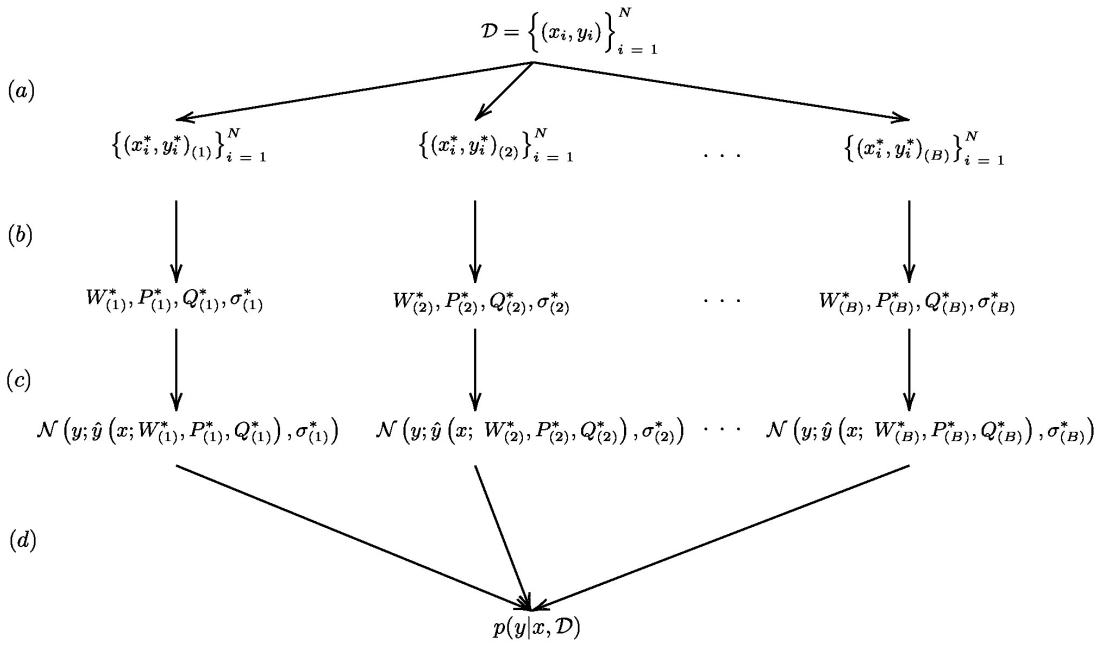


Figure 3.1: Diagram of the proposed method for generating a probabilistic prediction from PLS using bootstrap. (a) The original data set is sampled with replacement to produce bootstrapped data sets. (b) The bootstraped parameters are estimated using the standard NIPALS algorithm described in Section 2.2.1. (c) The bootstrapped parameters are used to generate a prediction for the output of PLS using a normal distribution with mean given by the prediction from PLS and variance given by the mean squared error of prediction. (d) The estimates from each of the bootstrapped samples are averaged out to give an estimate for the overall probabilistic distribution of the output given the input and observed data.

from the empirical distribution. Note that generating a N sample synthetic data set from the empirical distribution simply consists in sampling with replacement N data points from the training data set. For each of these new synthetic data sets $\mathcal{D}^*_{(b)}$, we estimate the PLS parameters $W^*_{(b)}$, $P^*_{(b)}$, and $Q^*_{(b)}$ as well as the Maximum Likelihood Estimate (MLE) for the variance of the prediction noise defined by

$$\sigma^{*2}_{(b)} = \frac{1}{N}\sum_{i=1}^{N}\left(y^*_{i,(b)} - \hat{y}\left(x^*_{i,(b)}; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right)\right)^2, \tag{3.14}$$

where $\hat{y}\left(x^*_{i,(b)}; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right)$ is the prediction from the PLS algorithm for input $x^*_{i,(b)}$ using parameters $W^*_{(b)}$, $P^*_{(b)}$, and $Q^*_{(b)}$. Secton 2.2.1 describes the method for finding this $\hat{y}$ prediction. Given these estimates, we compute the prediction probability of y for a new observation x for each of the B sets of parameters. Here we assume that the prediction error is normally

distributed, so

$$p\left(y \mid x, W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}, \sigma^*_{(b)}\right) = N\left(y; \hat{y}\left(x; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right), \sigma^*_{(b)}\right) \tag{3.15}$$

where $N(y; \mu, \sigma)$ denotes the probability density function (p.d.f) of a normal distribution with mean μ and variance σ^2 evaluated at y. The overall prediction is found by averaging the predictions of the B individual bootstrapped predictions. Algorithm 1 presents the method.

Mathematically, our method constructs a bootstrap predictive distribution by computing the expectation of the predictive distribution over different possible bootstrap data sets, similarly to Fushiki et al. [2005]. The resulting predictive distribution is given by

$$p(y|x, \mathcal{D}) = E_{\mathcal{D}^*}\left[p\left(y|x, \hat{\omega}(\mathcal{D}^*)\right)\right] = \int p\left(y|x, \hat{\omega}\left(\mathcal{D}^*\right)\right)\ p\left(\mathcal{D}^*|\hat{p}\right)\ d\mathcal{D}^*, \tag{3.16}$$

where $\hat{\omega}(\mathcal{D}^*) = (P^*, Q^*, W^*, \sigma^*)$ denote the estimated parameters for a given data set $\mathcal{D}^*$. The expectation is taken over the bootstrap data set $\mathcal{D}^*$, which consists of N samples drawn independently from the empirical distribution

$$\hat{p}\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta\left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x_i \\ y_i \end{pmatrix}\right) \tag{3.17}$$

where $\delta(\cdot)$ is the Dirac delta function. The integral in Equation (3.16) is approximated using Monte Carlo integration as follows

$$p\left(y|x, \mathcal{D}\right) \approx \frac{1}{B}\sum_{b=1}^{B} p\left(y|x, \hat{\omega}\left(\mathcal{D}^*_{(b)}\right)\right) \tag{3.18}$$

for both the where $\mathcal{D}^*_{(b)}$, $b = 1 \ldots B$, are bootstrapped data sets.

One aspect to note about our method is that the standard deviations of any individual predictions do not depend on the location of the input. Instead, we achieve a prediction uncertainty that is dependant on the inputs by changes in the mean of the predicted distributions. Points further from the mean of the training data will produce prediction means that are further

Algorithm 1 Method for calculating probabilistic bootstrap PLS predictions

1. For b = 1 to B
   (a) Generate a data set $\mathcal{D}^*_{(b)} = \left\{(x^*_i, y^*_i)_{(b)}\right\}_{i=1}^N$ from the empirical distribution by sampling with replacement from the training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$
   (b) Calculate the PLS parameters $\left(P^*_{(b)}, Q^*_{(b)}, W^*_{(b)}\right)$ and estimate of the variance of the noise from a single bootstrapped data set $\sigma^*_{(b)}$ using

$$\sigma^{*2}_{(b)} = \frac{1}{N}\sum_{i=1}^{N}\left(y^*_{i,(b)} - \hat{y}\left(x^*_{i,(b)}; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right)\right)^2, \tag{3.19}$$

2. Calculate the predictive distribution

$$p\left(y|x, \mathcal{D}\right) \approx \frac{1}{B}\sum_{b=1}^{B}\mathcal{N}\left(y; \hat{y}\left(x; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right), \sigma^*_{(b)}\right), \tag{3.20}$$

   where $\mathcal{N}(y; \mu, \sigma)$ is the p.d.f. of a normal distribution with mean μ and standard deviation σ evaluated at position y, and $\hat{y}\left(x; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right)$ is the prediction from the PLS algorithm for input x using parameters $W^*_{(b)}$, $P^*_{(b)}$, and $Q^*_{(b)}$

from one another, resulting in a wider overall distribution for $p(y|x, \mathcal{D})$, despite each individual prediction having constant variance.

3.4 Applications

3.4.1 Computation and Code

All of the computations were carried out on a 2020 M1 MacBook Pro with 16GB of ram. The code can be found at . The code for the third Design Space example and the spectroscopy example, is not included for confidentiality reasons.

3.4.2 Setup

This section illustrates the performance of the proposed approach (see Section 3.3) on different case studies. We claim that the bootstrapping approach is especially strong in generating good

probabilistic predictions in low-data settings that do not follow a standard linear model, i.e., a more complex model than ordinary least squares regression. These low-data, non-standard settings frequently arise in practice. To show the potential of the bootstrapping approach, we proceed by demonstrating its performance on a series of examples: the purpose of these examples is to show that the probabilistic bootstrapping approach performs how we would expect in straightforward examples and offers compelling advantages for real-world test instances.

This section studies two major applications for the bootstrapping approach, which constructs a probabilistic prediction for the output y given an input x and a set of training data $\mathcal{D}$. The first application is identifying a Design Space, i.e. the multidimensional set of input variables, including both material and process parameters, that results in a product of the desired quality [ICH Expert Working Group, 2009]. Our simple example and two industrially-relevant case studies in Michael Addition and Reductive Amination consider Design Space identification. The second application, calculating probabilistic prediction densities for the output y space, is demonstrated for applications in High Shear Wet Granulation and spectroscopy.

We compare the bootstrapping approach to the results achieved using the zeroth order approximation in Equations (3.11) and (3.12). Due to the linear approximation in the zeroth order method, authors have typically used PLS only as an intermediate step to finding the Design Space [Facco et al., 2015, Bano et al., 2017, 2018, Palací-López et al., 2019]. However, we compare to the zeroth order method because it is widely used when considering uncertainty in PLS [Zhang and García-Muñoz, 2009, Facco et al., 2015, Bano et al., 2017]. We use these comparisons to illustrate the differing performance of our method under a range of different circumstances.

3.4.3 Identifying a Design Space

The Design Space is the set of input values to a manufacturing process which will produce a desired set of output parameters with a probability above a given threshold based on a given model and training data set. Peterson [2008] defines the Design Space mathematically and

probabilistically as

$$DS = \{x \in \mathbb{R}^{n_x} | p\left(y \in y_{des} | x, \mathcal{D}\right) > 1 - \alpha\}, \tag{3.21}$$

where $1 - \alpha$ is a minimum acceptable probability. Sections 3.4.4–3.4.6 consider applications identifying a *Design Space*, i.e. the multidimensional set of input variables, including both material and process parameters, that results in a product of the desired quality.

Identifying Design Spaces has received much study within the literature [Ochoa et al., 2021, Laky et al., 2019, Lebrun et al., 2013, Kusumo et al., 2019]. There has also been notable work using PLS to help identify the Design Space, e.g. by identifying a region, known as the Experimental Space, within which the Design Space is predicted to fall [Facco et al., 2015, Bano et al., 2017, Palací-López et al., 2019]. Another Design Space identification example uses PLS to perform dimensionality reduction to find suitable samples to examine using a Bayesian model [Bano et al., 2018].

Applying the bootstrapping approach to Design Space identification requires integrating the probability distribution function over all acceptable values. When the output is one dimensional, Equation (3.20) can be simply adapted to replace the normal probability distribution functions with the difference between the cumulative distribution function (c.d.f.) values at the minimum and maximum acceptable values for the output, resulting in the equation

$$p\left(y|x, \mathcal{D}\right) \approx \frac{1}{B} \sum_{b=1}^{B} \Phi\left(y_{max} | \hat{y}\left(x; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right), \sigma^*_{(b)}\right) - \Phi\left(y_{min} | \hat{y}\left(x; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right), \sigma^*_{(b)}\right), \tag{3.22}$$

where $\Phi(\cdot|\hat{y}_b, \sigma_b)$ is the c.d.f. of the normal distribution with mean $\hat{y}_b$ and standard deviation σ_b.

For multi dimensional outputs a Monte Carlo approach is used, where samples are randomly drawn from the predicted probability distribution and the fraction of the samples which are within the desired output range is taken as the probability. To draw random samples first a random integer (b') between 1 and B is drawn, then a point is drawn from the p.d.f. of a multivariate normal distribution using the parameter values found using the b'th parameter values. In order to fully define this multivariate normal distribution the variance $\sigma^*_{(b)}$ needs to

be replaced by the covariance matrix $\sigma^*_{(b)}$ with elements defined by

$$\sigma^*_{(b),kl} = \frac{1}{N}\sum_{i=1}^{N}\left(y^*_{i,(b),k} - \hat{y}_k\left(x^*_{i,(b)}; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right)\right)\left(y^*_{i,(b),l} - \hat{y}_l\left(x^*_{i,(b)}; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right)\right), \quad (3.23)$$

where k and l indicate the output dimension that is being referred to. This process is described in Algorithm 2.

One point which we would like to acknowledge is that Design Space identification problems often do not follow the latent structure implicitly assumed by PLS. We believe that considering PLS in this type of tasks is reasonable, as in these cases the dimensionality reduction acts as a type of variable selection, which can allow PLS to perform well. In fact, PLS has been studied in the literature for similar applications for a significant period of time [Jaeckle and MacGregor, 2000]. However, there is a requirement that any practitioner interested in applying any PLS uncertainty technique first considers whether a PLS model is appropriate for a given task, or whether a task specific model - such as the ones put forward by Reis et al. [2019] - performs better.

Defining the knowledge space

When using a PLS model to find the model space, the model is only considered valid for data "close to" the training data. The region for which the model is assumed to be valid is found by assuming training data is drawn from a multivariate normal distribution, then finding confidence limits that distribution. If the new input point falls outside the region defined by these confidence intervals the model would not be considered vaid as a method for prediction.

There are two statistics used for this: T^2 and Squared Prediction error (SPE): the T^2 statistic and the SPE statistic. The T^2 statistic measures the position within the latent space. The SPE measures the distance of a point from the latent space. In order for a point to be considered to have been drawn from the same region the SPE and the T^2 statistic need to both be below pre-defined thresholds.

The T^2 statistic for an input x^* is given by

$$T_i^2 = \sum_{j=1}^{n_l} \frac{t_j^{*2}}{\lambda_j}, \tag{3.24}$$

where n_l is the number of latent variables, t_j^* is the value of the data point x^* along the jth latent variable and λ_j is the variance along the jth latent variable.

The value of a points Squared Prediction Error (SPE) for a data point x^* is given by

$$SPE = e^{*T} e^* \tag{3.25}$$

where e^* is the portion of x^* that was discarded by PLS model.

The upper confidence limit for T^2

$$UCL\left(T^2\right)_\alpha = \frac{n_l\left(N^2-1\right)}{N(N-n_l)} F_{(n_l,(N-n_l)),\alpha} \tag{3.26}$$

where N is the number of samples in the training data, α is a desired confidence level, and $F_{(n_l,(N-n_l)),\alpha}$ is the α quantile of F distribution with n_l and $N-n_l$ degrees of freedom.

The upper confidence limit for SPE is given by

$$UCL(SPE)_\alpha = \frac{v}{2b} \chi^2_{(2b^2/v),\alpha}, \tag{3.27}$$

where b is the mean of the training points, v is the variance, $\chi^2_{(2b^2/v),\alpha}$ is the α quantile of a χ^2 distribution with $2b^2/v$ degrees of freedom

Calculating probabilistic prediction densities

One downside of using real data for Design Space identification is that we are unable to compare to the true Design Space, as this is an unknown quantity that we are trying to estimate. To show the validity of our method for real data, Sections 3.4.7–3.4.8 compare the calculation of probabilistic prediction densities for the output y to the observed outputs in High Shear Wet

Granulation and Spectroscopy. This comparison allows us to show that our method is able to make predictions that meaningfully capture the uncertainty of the model for real data.

Algorithm 2 Monte Carlo method for calculating the probability of an output falling in the desired range

1. For m = 1 to M:
   (a) Randomly select integer b' between 1 and B
   (b) Draw a random number $y'_m$from the multivariate normal $N\left(\hat{y}\left(x^*_{i,(b)}; W^*_{(b)}, P^*_{(b)}, Q^*_{(b)}\right), \Sigma^*_{(b')}\right)$
2. The probability can then be calculated using

$$p = \frac{1}{M}\sum_{m=1}^{M}\prod_{k=1}^{n_y} I\left(y_{min,k} < y'_m < y_{max,k}\right), \tag{3.28}$$

   where $I(\cdot)$ is the indicator function and the subscript k indicates which output dimension is being referred to

3.4.4 Simple Simulated Example

A very simple Design Space identification example illustrates the technique. In this example, a two dimensional input (x_1, x_2) generates a one dimensional output (y), via the equation

$$y = x_2 + \epsilon, \tag{3.29}$$

where ϵ is a Gaussian noise term with variance 0.2. A training data set $\mathcal{D}$ with $N = 40$ samples was generated by sampling input values from a multivariate normal distribution $(x_1, x_2) \sim N((5,5)^T, 5\,\mathbb{I})$ and the output y was generated using Equation (3.29).

To identify the Design Space for $y_{\text{des}} = [8, 12]$ with $\alpha = 0.9$ based on the training data $\mathcal{D}$, we estimate the bootstrap predictive distribution using $B = 1000$ bootstrap data sets and construct the Design Space. Figure 3.3 represents the resulting Design Space. The contours depicted in Figure 3.3 come from calculating the probability of each point in a 100×100 grid in the input space resulting in the desired output. This grid was interpolated in order to produce the contours. The full details of the implementation can be found in the provided code. An

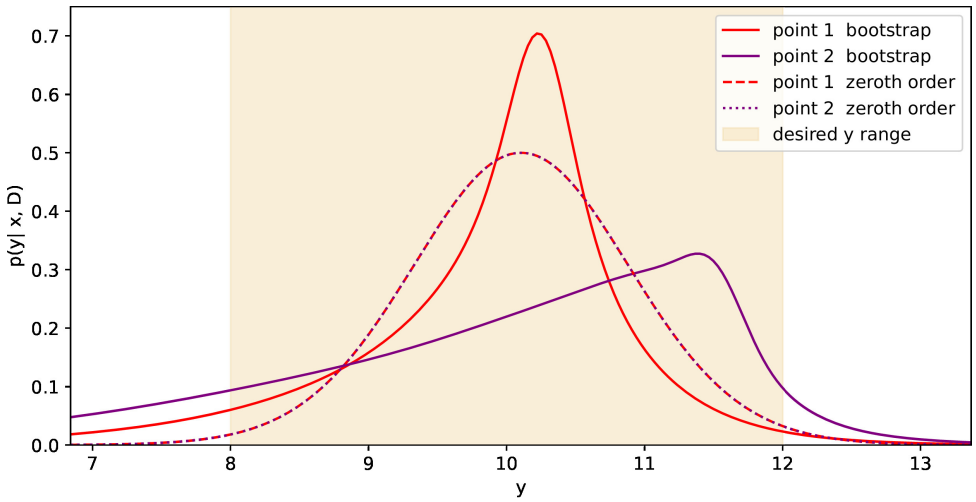


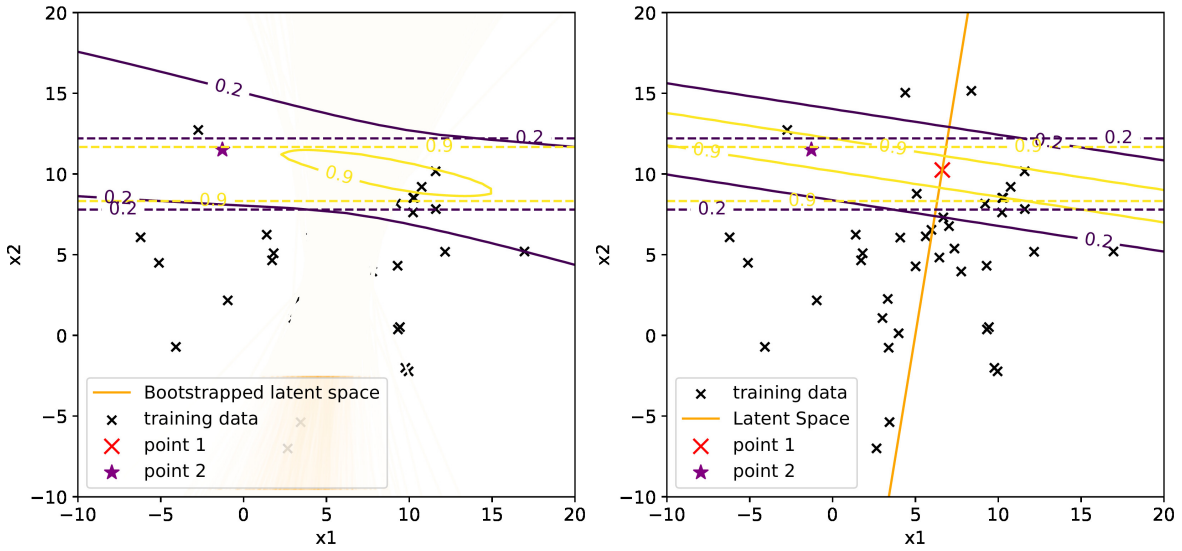
Figure 3.2: Comparison of the two predictions of the probability density function from the zeroth order approximation method and the bootstrap prediction. The solid lines show the bootstrap method proposed here, the dashed line shows the prediction from the zeroth order approximation method, and the shaded area is the desired output range for y. As can be seen above, there is no difference in the predictions from the two points using the student-t method, whereas the bootstrap method applies significant weight to a far wider range of values.

important feature of our approach is that it considers uncertainty in the latent space through bootstrapped data sets. Figure 3.3 illustrates how the latent space estimated using all the training data differs from the latent spaces estimated using each of the 1000 bootstrapped data sets. We observe that, by incorporating the uncertainty via bootstrapping, the Design Space constructed using our approach is naturally restricted to be close to the latent space.

We now compare the Design Space found by our method to the one obtained using the zeroth order method as well as to the so-called true Design Space, that is the Design Space if one knew the data-generating process, i.e.

$$\left\{x \in \mathbb{R}^2 | p\left(y \in [8, 12] | x\right) > 1 - \alpha\right\} = \left\{x \in \mathbb{R}^2 | p\left(8 - x_2 \leq \epsilon \leq 12 - x_2\right) > 1 - \alpha\right\}.$$

Figure 3.3 shows that the Design Space constructed using the probabilistic bootstrap approach is included within the true Design Space, while the Design Space constructed using the zeroth order method only overlaps with the true Design Space. Recall that the zeroth order method makes predictions that are invariant with respect to the distance of the point from the latent space. In other words, the zeroth order method assigns equal probability to all points within the input space (x) that map to the same point in the latent space (t). Therefore, the Design Space



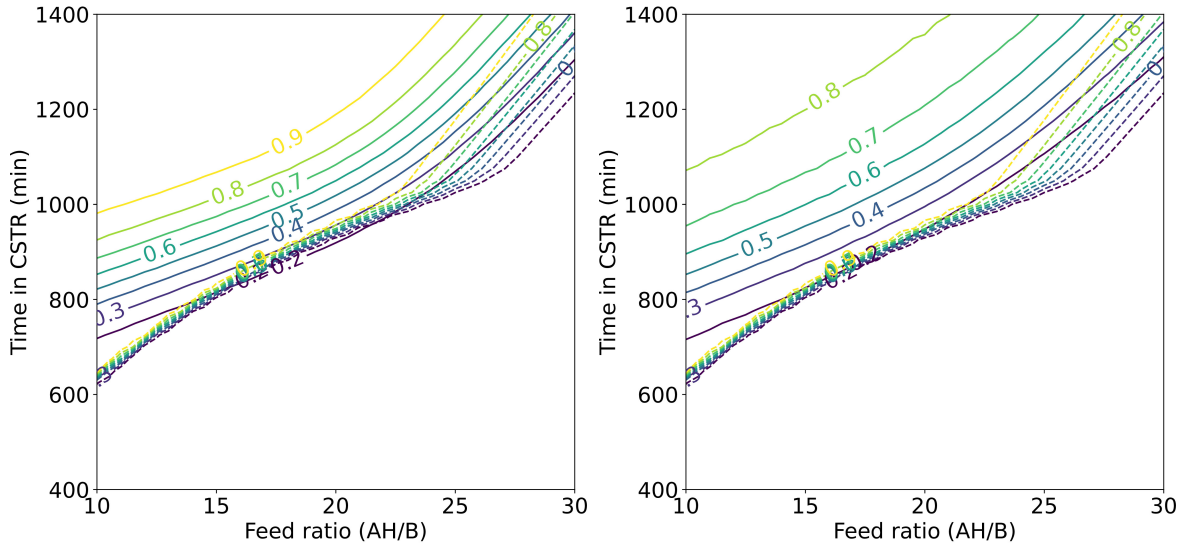
(a) Bootstrap Method (b) Zeroth Order Method

Figure 3.3: Figure showing the Design Space for a simple simulated example. The dashed lines indicate the true probabilities of data generated at that point falling in the desired range. The red and purple points indicate points of interest discussed in Figure 3.2. Figure (a) shows the latent spaces (orange lines) found by the bootstrapped data sets, and the resulting Design Spaces using probabilistic bootstrap prediction – whose shape depends on the distance to the latent space. Figure (b) shows the latent space (orange line) found by the regular PLS model using the training data set, the resulting Design Spaces (blue and green solid lines), do not account for the distance to the latent space.

resulting from the zeroth order method is a band orthogonal to the estimated latent space. In contrast, the probabilistic bootstrap approach produces curves that reflect the uncertainty arising from the portion of the input not typically considered by a PLS model.

Figure 3.2 shows the estimated predictive distributions $p(y|x, \mathcal{D})$ for two different values of x, indicated by the purple and red points in Figure 3.3, using both the zeroth order and probabilistic bootstrap prediction methods. The two values of x have identical positions in the latent space. As the zeroth order method described in Equations (3.11) and (3.12) only considers the projections of the two points on the latent space, the two posterior predictive distribution using the zeroth order method are identical. In contrast, the bootstrap method, which does consider the uncertainty in the position of the latent space, produces two distinct posterior predictive distribution. The probabilistic bootstrap approach emphasizes the importance of the distance to the latent space as a source of uncertainty.

The prediction arising from the bootstrap method is strongly negatively skewed. This is in



(a) Bootstrap Method (b) Zeroth Order Method

Figure 3.4: Design Spaces for the Michael addition reaction using bootstrapped and zeroth order methods – Figure (a) shows the Design Space from the bootstrap method, and Figure (b) showing the zeroth order method. The axes indicate different values for two independent inputs to the manufacturing process. The contours indicate the differing probabilities of given inputs being within the Design Space, with the solid lines indicating the probabilities from the PLS methods, while the dashed lines indicate the true probabilities of that input resulting in an acceptable output.

contrast to any uncertainty method for PLS models that relies on estimating parameters for the student-t distribution, including the zeroth order method. As discussed earlier, the student-t distribution arises from assuming that the distribution can be described by just an unknown mean and variance. By relaxing the assumption that uncertainty in PLS parameters, plus Gaussian noise with unknown variance results in a student-t distribution, we see that the resulting distribution is markedly skewed and different from any student-t distribution. This skew is due to the known non-linearities in the parameter estimation of PLS.

3.4.5 Michael Addition Reaction

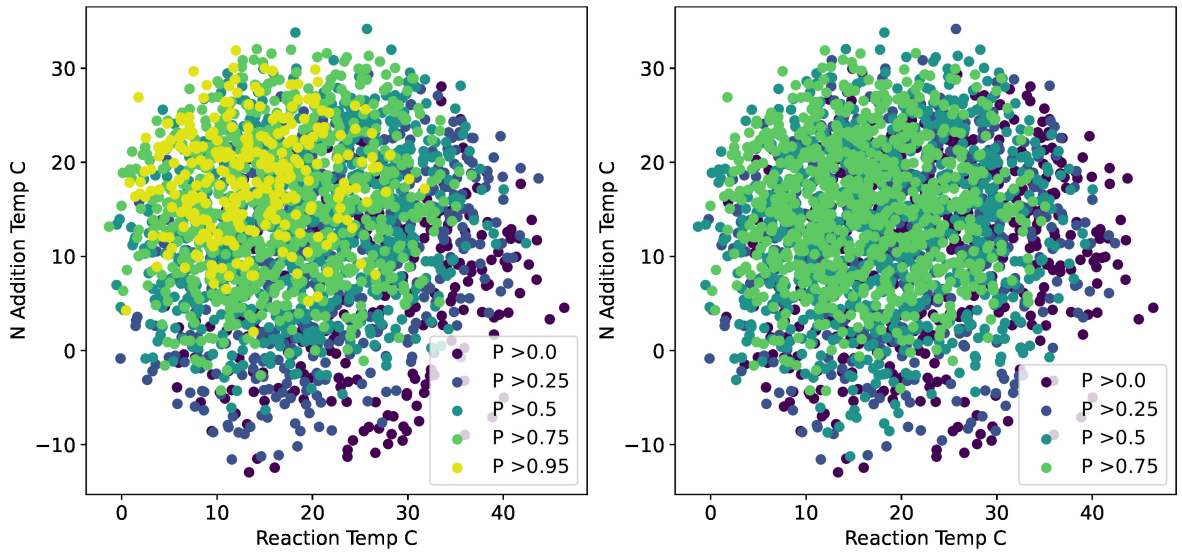
The next example explores Design Space identification for the Michael Addition reaction. The full details on the simulation of this reaction are given by Kusumo et al. [2019]. For this chapter, the real data generating process will be treated as a black box that accepts two process inputs (the molar ratio of two reactants, and time in a continuous stirred reactor vessel) and predicts two outputs (a greater than 90% conversion of the carbon, and a residual molar concentration

of an intermediate product). A number of 25 training data points were simulated in a grid arrangement across the inputs, with molar ratio varying between 10 and 30, and the time in the continuous stirred reactor vessel was varied between 400 and 1400 minutes. Training data was initially generated without noise, then a Gaussian error term was added. The standard deviation of the error term was set as 1% of the mean of the training data before the noise was added.

This data was fit with a two component PLS model, which was used to calculate a Design Space with a greater than 90% conversion of carbon and a residual molar concentration of less than 2 mmol per liter. The results of these are shown Figure 3.4, with the true probabilities of the outputs being in the Design Space shown as dashed lines, and the Design Spaces found by the two models shown in solid lines. In this case it is obvious that bootstrap method performs significantly better, correctly capturing a significant portion of the inputs in this range as having a greater than 90% chance of being within the Design Space, while the zeroth order approximation fails to predict any region of these inputs as having a greater than 90% chance of being within the Design Space. While it is not possible to identify exact causes of the significant difference in the performance between the two methods, previous results from Fushiki [2005] have shown that in general bootstrap prediction method performing remarkably well in cases where no set of input parameters for the model would recapitulate the true data generating process, as is the case here.

3.4.6 Reductive Amination

The final Design Space identification example is an industrial case study. In this example there are 15 input dimensions and two output dimensions of interest, one conversion percentage for outputs with a minimum required conversion rate and one contaminant with a maximum allowable presence in the final product. Due to the high dimensional inputs, grid sampling, as was used in the Michael Addition reaction is not possible, so instead a Markov Chain Mote Carlo (MCMC) was used to sample the areas of higher probability of being within the Design Space. In order to constrain the samples to be close to the historic data, samples were only



(a) Bootstrap Method (b) Zeroth Order Method

Figure 3.5: Projection of randomly sampled points in the input space to the two dimensional space of just reaction temperature and nitrogen addition temperature. The differently colored points indicate the different probabilities for the given input resulting in an acceptable output. Figure (*a*) shows the probabilities found using the bootstrap method proposed here, Figure (*b*) shows the probabilities using the zeroth order method. The same point appears to have different probabilities of being withing the Design Space as they have been projected down from a higher dimension - in reality these points are separated along one or more of the other dimensions not shown here.

drawn from the Knowledge Space defined by Facco et al. [2015], and described in more detail in Section 3.4.3.

The results of both the bootstrap and the zeroth order approximation are shown in Figure 3.5. This plot shows the projection of points down to a two dimensional subspace of the true input dimension – a temperature of a reactant when added and the temperature of the reaction. These points are color coded by the probabilities of the given inputs producing an output fulfilling both a greater than 90% conversion to the desired output, and a less than 1% presence of an impurity generated by the reaction. For these acceptance criteria the zeroth order method fails to find any points that would constitute a Design Space at a confidence level of 90%. Meanwhile the bootstrap method proposed here predicts that a large number of the points will fulfill the required Design Space criteria. While it is not possible to comment on the accuracy of this Design Space, there are many situations in which it is preferable to have a larger Design Space, even at an inevitable risk of incorrectly assigning some points that should not truly be in the

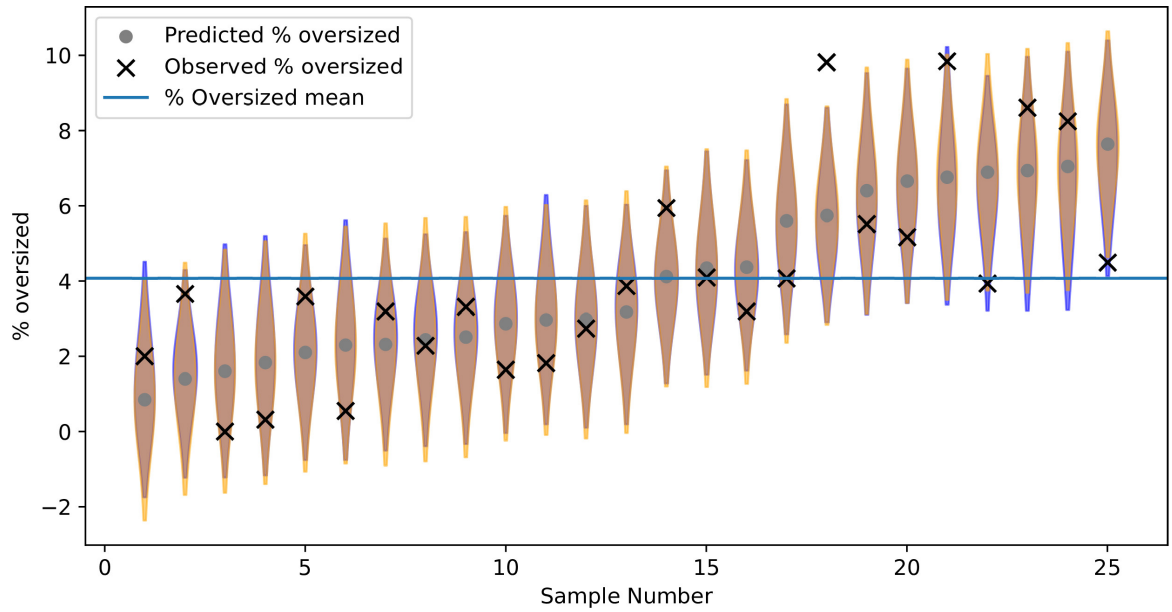


Figure 3.6: Violin plot for the data given in Vemavarapu and Badawy [2019]. The blue shapes indicate the probability distribution found by the bootstrap prediction, whereas the orange shows the prediction from the zeroth order approximation method. The points are the prediction from PLS before uncertainty is included and the crosses are the true value of the % oversized particles. The blue line horizontal line is the mean value of the training data outputs. For the sake of clarity both violin plots are truncated at the 0.025 and 0.0975 quantiles of the plot to remove tales from both distributions.

Design Space.

3.4.7 High Shear Wet Granulation

This data set arises from a wet granulation process [Vemavarapu and Badawy, 2019]. Vemavarapu and Badawy [2019] consider how six input dimensions: aqueous solubility, contact angle, water holding capacity, two distinct measures of particle size, and surface area affected a number of response variables. In this chapter we only consider the percentage of particles larger than 1.4mm as the response of interest. We abstract away specifics and refer to the inputs as $x_j,\ j \in \{1, \ldots, 6\}$ and the output as y. We scale the data as described by Vemavarapu and Badawy [2019] in order to improve the model performance, i.e. we apply a logarithm to x_j for $j \in \{1, 3, 4, 5\}$ and a cube root to x_6. For each prediction a different training set, excluding the point that is being predicted, but included all other points was created. This new data set was used to train a PLS model and find the uncertainty predictions.

Figure 3.6 shows a violin plot of the prediction density between the 0.025 and 0.975 quantiles of the predicted distribution, along withe the true observed value and the prediction from using the deterministic prediction form PLS. The prediction from the zeroth order method is also included for the sake of comparison. Both of these methods produce credibility intervals that contain the majority of data points, with the bootstrap prediction only failing to predict the interval for sample number 18 and the zeroth order approximation missing samples 18 and 25. The primary difference between these predictive densities is that the bootstrap prediction generally places more weight on the samples being closer to the mean observed values in the training set - shown as a blue horizontal line in the figure. This preference for predicting the mean by the bootstrap method shifts the intervals upwards when the prediction is significantly below the mean and downwards when the prediction is above the mean. This effect is the reason that the bootstrap prediction finds sample number 25 within it's predicted interval, while the zeroth order approximation does not.

3.4.8 Spectroscopy

The last example looked at here uses spectroscopic data [Hetrick et al., 2017, García-Muñoz and Hernandez Torres, 2020]. The raw data was pre-processed with the methods described by Hetrick et al. [2017], where only wavelengths from 1626 to 1993 nm are used with Standard Normal Variate transformation applied (transforming the data to have mean zero and variance one), and a three latent variable PLS model. Spectroscopy was chosen to evaluate how our method performed in high dimensional settings. Additionally, this case study provided a good chance to evaluate our method in comparison to the zeroth order approximation method, where it has already been shown to perform well [Faber, 2002]. Note again that the data for this case study has not been released for confidentiality reasons.

Two PLS models were fit from this pre-processed data, one using 50 training data points and one using 1000 training data points, these are shown in Figure 3.7. As can be seen for this data set the zeroth order approximation and the prediction from the bootstrapped PLS method are indistinguishable. This is a boon for our method as it has been found on multiple occasions that

for spectroscopy data sets the zeroth order approximation method is very effective in finding accurate predictions [Faber, 2002], suggesting that even in cases where established methods perform well, our method is not out performed.

3.5 Conclusions

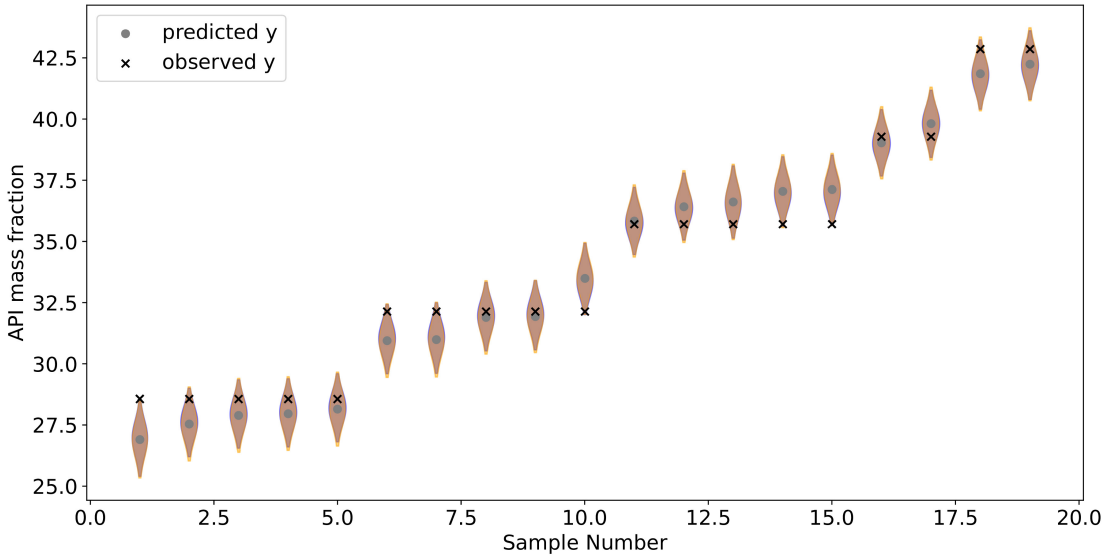
This chapter applies the techniques described by Fushiki et al. [2005], which allows us to make principled probabilistic predictions from PLS. The probabilistic bootstrapping approach considers the uncertainty arising from the interaction between the parameter uncertainty and the portion of the input variables typically ignored by PLS. In addition to the theoretical justifications for this method, the probabilistic bootstrapping approach finds meaningful for prediction intervals for real world data, and runs efficiently enough to provide useful Design Spaces.

Although prediction and Design Space identification are the two topics chosen to demonstrate the power of our method here, they are by no means the only uses that our method could be put to. The method could be adapted to identify regions where it is likely that the Design Space could be – similar to finding experimental spaces as described by Facco et al. [2015] and Bano et al. [2017]. This differs from finding a Design Space, as some points in the input space have moderate probability for producing a desired output as a result of some of the uncertain parameters predicting likely to be within the output range, while others do predict it to be very unlikely. Meanwhile other points in the Design Space could be estimated equally likely to produce an outcome in the desired range, but as a result of all of the uncertain parameters predicting a moderate probability being within the Design Space. Of these two cases the former is more interesting to explore than the latter, as knowledge of the true parameter values may significantly improve the probability of finding a point to be within the Design Space, which is not true for the latter.

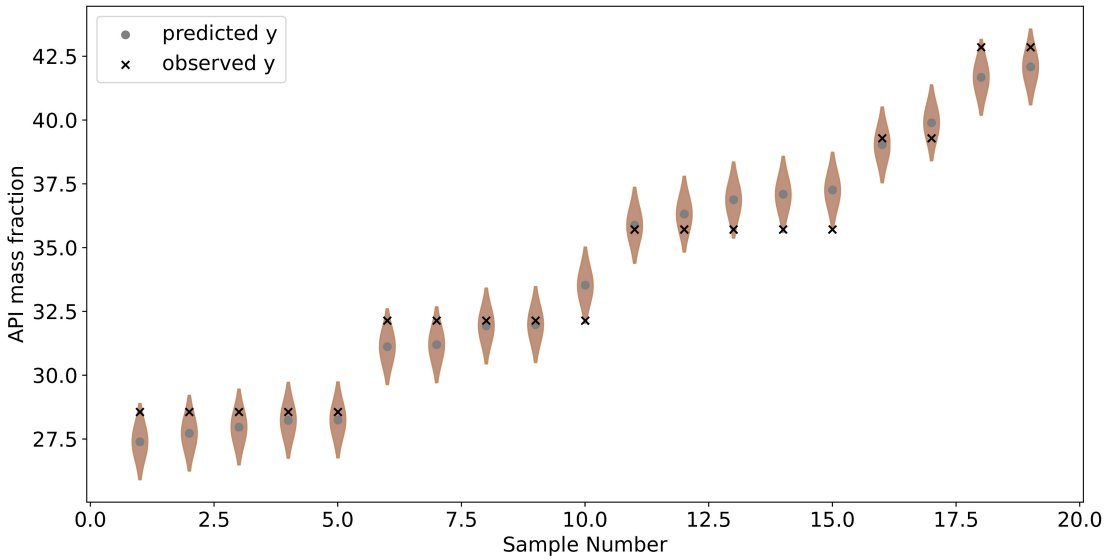
Another direct application of the method presented in this chapter is in pharmaceutical manufacturing, specifically in the implementation of a Real Time Release Testing solution using a

PLS model. The method presented here allows the uncertainty of the model to be accounted for in the decision making process (e.g. to accept or to reject product), without making (invalid) assumptions regarding the trustworthiness of the predictions. The consideration of uncertainty can be of particular interest in applications near a physical boundary, like a decay curve in continuous manufacturing, or the estimation of drug content for a low dose product.

One short fall of our method is that it assumes that the system can be well approximated by a linear prediction and that the difference between the predicted and the true values of the output are explainable by noise. However, many relationships in Chemical Manufacturing and Chemometrics cannot be well modeled by a linear relationship between input and output variables. The current work is unable to distinguish the models uncertainty due to noise and uncertainty arising from mismatch between PLS and the true system being investigated. An open avenue for PLS research is to consider methods to incorporate the uncertainty arising from possible these model mismatched.



(a) 50 training data points



(b) 1000 training data points

Figure 3.7: The prediction from bootstrap prediction and predictions using a zeroth-order approximation. The blue shapes indicate the probability distribution found by the bootstrap prediction, whereas the red shows the prediction from the zeroth-order approximation method. The points are the prediction from PLS before uncertainty is included and the crosses indicate the observed values for the API fraction. The two predictions are indistinguishable for the 1000 point training data set, and nearly identical for the 50 point training data set.

Chapter 4

Weighted-Sums of Gaussian Process Latent Variable Models

This chapter is currently under review for publication and is available as a preprint [Odgers et al., 2024]. Ruby Sedgwick helped with running various experiments as well as editing portions of the text.

4.1 Introduction

We develop a Bayesian non-parametric approach for handling additive models, focusing on *signal separation*. Signal separation considers applications where a single observation is composed of several components and typically aims to either identify what are the underlying signals [Jayaram and Thickstun, 2020, Chatterjee et al., 2021] or estimate the fraction of each signal present [Wytock and Kolter, 2014, Yip et al., 2022].

We motivate our approach by considering spectroscopy, which seeks to identify the abundances of different chemical species $\{c = 1, \ldots, C\}$ in a particular sample based on the observation of many wavelengths. Spectroscopic signals often naturally give rise to signal separation problems, due to Beer-Lambert's law which which states that light absorption at a particular wavelength

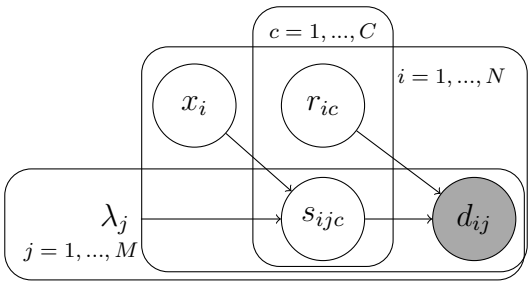


Figure 4.1: Bayesian diagram for Weighted-Sum of Gaussian Process latent variable models (WS-GPLVM) with unobserved mixture weights r_{ic}.

due to a particular chemical species is proportional to the concentration of that chemical species and the path length through the medium. Adding complexity to the problem, the signal of each chemical, referred to in this paper as the *pure component signals*, depends not only on the wavelength but also on other factors, for example, temperature Wülfert et al. [1998], which are often unobserved. Our key idea, and the starting point for this paper, is to use a latent variable $x_i \in \mathbb{R}^A$ to model these unobserved factors:

$$d_{ij} = \sum_{c=1}^{C} r_{ic}\, s_c(x_i, \lambda_j) + \epsilon_{ij}, \tag{4.1}$$

where $\epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$ and the pure component signals $s_c(\cdot, \cdot)$ are unknown. This is often combined with the additional constrain that the weights sum-to-one:

$$\sum_{c=1}^{C} r_{ic} = 1 \; \forall \; i \quad \text{and} \quad r_{ic} \in [0, 1] \; \forall \; i, c. \tag{4.2}$$

Similar setups appear often in spectroscopy [Tauler, 1995, Kriesten et al., 2008, García-Muñoz and Hernández Torres, 2020], but the formulation is more general and also used in other applications such as hyper-spectral unmixing [Altmann et al., 2013, Borsoi et al., 2019, 2021] and image/audio processing Gandelsman et al. [2019], Jayaram and Thickstun [2020], Mariani et al. [2023].

Our motivating example places constraints that no existing methods can solve. Namely, we are interested in settings where:

- there is no parametric description of the signals, so we cannot rely on physical models

[Alsmeyer et al., 2004, Dirks and Poole, 2019],

- there are relatively few observations of which only a small number are labeled, so we cannot rely on deep learning [Neri et al., 2021, Hong et al., 2021]
- we may not have data on the pure signals s_c, that is we may lack data with all r_{ic} equal 0 except one $\hat{c}$ with $r_{i\hat{c}} = 1$, so we cannot follow Borsoi et al. [2019] and Jayaram and Thickstun [2020].

The challenges presented by these constraints necessitate appropriate regularization. Gaussian Processes (GPs) are an attractive choice for this due to their ability to automatically regularize through the optimization of the marginal likelihood [Rasmussen and Williams, 2006]. As the conditions driving variability of the signals are unobserved, we use Gaussian Process Latent Variable Models (GPLVMs) [Lawrence, 2005, Titsias and Lawrence, 2010, Damianou et al., 2016, Dai et al., 2017, de Souza et al., 2021, Ramchandran et al., 2021, Lalchand et al., 2022] to estimate what the conditions driving variation were. To the best of our knowledge, the setting where data arises from a weighted sum of GPLVMs has not been studied previously.

Related Works. Multi-output GPs often assume data is generated as the sum of unobserved latent functions, equivalent to what we refer to as pure component signals, with the goal of capturing covariance between different outputs [Teh et al., 2005, Alvarez and Lawrence, 2011, 2008, Álvarez et al., 2012]. While these methods can capture relationships between latent functions and observed data, none of them allow for variation in latent functions across different observations. By introducing latent variables as inputs to the latent functions our method captures variations in these latent functions which are not possible otherwise.

Gaussian Process Factor Analysis (GPFA) [Yu et al., 2008, Luttinen and Ilin, 2009, Nadew et al., 2024] considers sums of GPs with the goal of extracting latent signals. Our method differs from this in two ways. Firstly, the GPFA model does not include the latent variable x_i, and secondly, their setting considered all weights to be unobserved whereas our setting has a mixture of both observed and unobserved weights.

Specifically for *signal separation* settings, Altmann et al. [2013] use a GPLVM for hyper-spectral

imaging where the latent variables correspond to the unknown mixture fractional weights to capture a specific, physically motivated form of non-linearity. In contrast, our approach incorporates latent variables which may be independent of the mixture fractional weights allowing us to capture a wider range of potentially unknown non-linearities in the pure component signals.

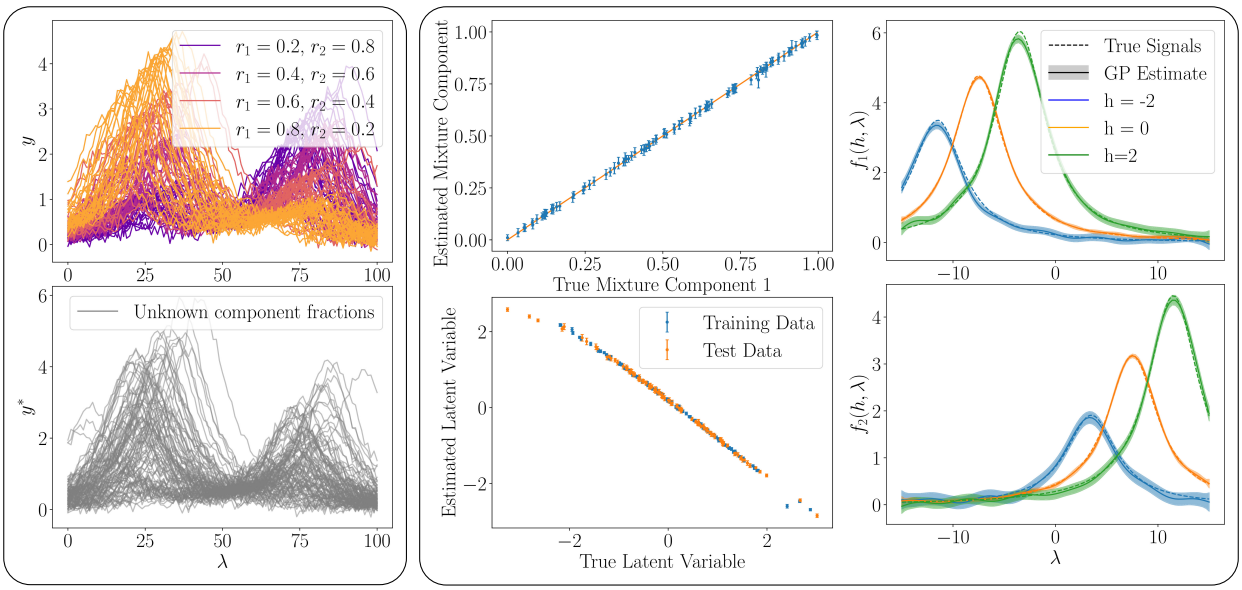
Contributions. We propose a model to make predictions for data arising from Equation (4.1) and (4.2) and derive an Evidence Lower Bound (ELBO) for this model to approximate the posterior distribution over the weights, pure component spectra and latent variables, as described in Section 4.2. The optimization of this ELBO is challenging due to the model tending to rely on the non-linear variations of the GPLVM, so in Section 4.3 we provide a training procedure which encourages the model to appropriately use both the latent variables, x_i, and any unobserved weights, r_i. We demonstrate our approach on real datasets in Section 4.4.

4.2 Weighted-Sum of Gaussian Process Latent Variable Models

We consider a dataset generated according to Equation 4.1 and assume each data point $i = 1, \ldots, N$ is observed at a fixed grid of $j = 1, \ldots, M$ wavelengths. We introduce the notation $s_{ijc} = s_c(x_i, \lambda_j)$ for the pure component signals. Throughout this derivation, we often need to aggregate over the C pure components and weights. We denote this with the vectors $r_i = (r_{i1}, \ldots, r_{iC})^T \in \mathbb{R}^C$ and $s_{ij} = (s_{ij1}, \ldots, s_{ijC})^T \in \mathbb{R}^C$.

Spectroscopy applications typically have a set of training observations $D = \{d_{ij}\} \in \mathbb{R}^{N \times M}$ with known weights $R = \{r_{ic}\} \in \mathbb{R}^{N \times C}$, and a set of test observations $D^* = \{d^*_{ij}\} \in \mathbb{R}^{N^* \times M}$ with unknown weights. We aim to learn the posterior for the unknown weights of the test observations, $R^* = \{r^*_{ic}\} \in \mathbb{R}^{N^* \times C}$, the latent variables for the training and test data, denoted by $X = \{x_i\} \in \mathbb{R}^{N \times A}$ and $X^* = \{x^*_i\} \in \mathbb{R}^{N^* \times A}$ respectively, and the pure component signals, $S = \{s_{ijc}\} \in \mathbb{R}^{N \times M \times C}$ and $S^* = \{s^*_{ijc}\} \in \mathbb{R}^{N^* \times M \times C}$.

With this data, Equation (4.1) is non-identifiable. Even for data points with known weights,



(*a*) Observed data (*b*) Variational Posterior Estimates

Figure 4.2: Illustrative example of WS-GPLVM: by inputting a set of data (*a*) of training data *(top)*, with known mixture fractions, and test data *(bottom)*, with unknown mixture fractions, WS-GPLVM produces variational posterior estimates of (*b*) the fractional composition of the test data *(top left)*, the latent variables of both the training and the test data *(bottom left)*, and a Gaussian Process estimate, shown with a mean and 95% confidence interval, for the pure component signals *(right)*.

r_i, Equation (4.1) defines a $C-1$ dimensional hyperplane of pure component signals, s_{ij}, which have equal likelihood. As there is no shared variable between each of the N data points, more observations do not help. One solution is to put a prior on the pure component signals that creates dependence between $s_c(x_i, \lambda_j)$ and $s_c(x_{i'}, \lambda_{j'})$. In spectroscopy the degree of dependence between the data points will vary: for data points with similar latent conditions, *e.g.* data collected at similar temperatures and humidity, the model should share lots of information, *i.e.* $s_c(x_i, \lambda_j) \approx s_c(x_{i'}, \lambda_j)$ when $x_i \approx x_{i'}$. Meanwhile, data points collected under very different conditions may share very little information. Additionally, spectroscopy datasets are often small and can exhibit a wide range of possible variations. GPLVMs [Lawrence, 2005, Titsias and Lawrence, 2010] are an attractive prior in this context as they can learn the degree of dependence between $s_c(x_i, \lambda_j)$ and $s_c(x_{i'}, \lambda_{j'})$, while still being both flexible and data efficient enough to work on realistic datasets.

Formally, we place a GP prior over each of the functions $s_c(\cdot, \cdot)$, $c = 1, \ldots, C$, such that the

vector containing all of the pure component signal values of that component, is distributed by

$$(s_{11c}, \ldots, s_{NMc}) \sim N(0, K^c_{\bullet\bullet}), \tag{4.3}$$

where the kernel matrix is given by

$$[K^c_{\bullet\bullet}]_{(ij)(i'j')} = k_c((x_i, \lambda_j), (x_{i'}, \lambda_{j'})). \tag{4.4}$$

This prior is not the only plausible one to achieve the dependence between data points and we discuss another choice of GP prior in Section 4.2.1.

We use a Bayesian treatment of the latent variable, along with an ARD kernel to achieve automatic determination of the effective dimensionality of the latent space, as demonstrated by Titsias and Lawrence [2010]. The prior for the latent variable x_i is a unit normal prior on the latent space:

$$x_i \sim N(0, I_A). \tag{4.5}$$

The prior for $r_i = \{r_{ic}\}_{c=1}^C$ depends on the problem setting. For our applications, we primarily use a Dirichlet prior to force the weights to sum-to-one and be positive:

$$r_i \sim Dir(1_c). \tag{4.6}$$

We discuss other options for priors for r_i in Section 4.2.1.

GPs are generally trained by maximizing the marginal likelihood [Rasmussen and Williams, 2006]. However, the Bayesian treatment of the latent variables, H, means the exact marginal likelihood is no longer tractable. We therefore follow the variational inference approach of Titsias and Lawrence [2010] to approximate a lower bound of the posterior distribution over the quantities of interest, S, S^*, X, X^* and R^*.

More precisely, for each component c, we introduce a set of inducing points $u_c \in \mathbb{R}^L$ drawn

from the Gaussian Process at a set of locations $V_c \in \mathbb{R}^{L\times A}$ so that

$$u_c \sim N(0, K^c_{V_cV_c}), \tag{4.7}$$

where the elements of the kernel matrix $K^c_{V_cV_c}$ are given by the kernel functions between the inducing point input locations. With these inducing points, the conditional distribution for each pure component signal is

$$\tilde{s}_c|u_c \sim N(\mu^s_c, K^s_c), \tag{4.8}$$

where

$$\mu^s_{ijc} = K^c_{(x_i\lambda_j)V_c}(K^c_{V_cV_c})^{-1}u_c \tag{4.9}$$

$$K^s_{ijc} = K^c_{(x_i\lambda_j)(x_i\lambda_j)} - K^c_{(x_i\lambda_j)V_c}(K^c_{V_cV_c})^{-1}K^{c\ T}_{(x_i\lambda_j)V_c}. \tag{4.10}$$

We are interested in the posterior distribution

$$p(\mathcal{S}, \mathcal{X}, R^*, U|\mathcal{D}, R) = \frac{p(\mathcal{D}|\mathcal{R}, \mathcal{S})p(\mathcal{S}|\mathcal{X}, U)p(\mathcal{R})p(\mathcal{X})p(U)}{p(\mathcal{D}, R)}, \tag{4.11}$$

where for conciseness, we used the following notations: $\mathcal{S} = \{S, S^*\}$, $\mathcal{X} = \{H, X^*\}$, $\mathcal{D} = \{D, D^*\}$, $\mathcal{R} = \{R, R^*\}$, and $U = \{u_c\} \in \mathbb{R}^{L\times C}$ the matrix containing the inducing points for all the components.

We aim to approximate this posterior distribution with a variational distribution of the form

$$q(\mathcal{S}, \mathcal{X}, R^*, U) = p(\mathcal{S}|\mathcal{X}, U)q(\mathcal{X})q(U)q(R^*) \tag{4.12}$$

which, similarly to the Bayesian GPLVM [Titsias and Lawrence, 2010], allows a simplification and an analytic computation of the ELBO given in Equation 4.15. We use a mean field assumption for both $q(\mathcal{X})$ and $q(R^*)$ such that the distributions factorize over the observations:

$$q(\mathcal{X}) = \prod_{i=1}^{N} N(x_i|\mu^x_i, \Sigma^x_i) \prod_{i=1}^{N^*} N(X^*_i|\mu^{X^*}_i, \Sigma^{X^*}_i) \tag{4.13}$$

with $\Sigma_i^x \in \mathbb{R}^{A\times A}$ and $\Sigma_i^{X^*} \in \mathbb{R}^{A\times A}$ being diagonal matrices and

$$q(R^*) = \prod_{i=1}^{N^*} q(r_i^*|\alpha_i), \tag{4.14}$$

where $q(r_i^*|\alpha_i)$ is a Dirichlet distribution with parameter $\alpha_i \in \mathbb{R}^C$. We make no assumption on the form of $q(U)$, although it is shown in Appendix D.2.1 to have an optimal distribution which is normal.

The optimal variational parameters, $\{\mu_i^h, \Sigma_i^h\}_{i=1}^N$, $\{\mu_i^{X^*}, \Sigma_i^{X^*}, \alpha_i\}_{i=1}^{N^*}$ and $\{r_i^*\}_{i=1}^C$, the inducing point locations, $\{V_c\}_{c=1}^C$, the Gaussian Process kernel hyper-parameters, and σ^2, can be jointly found by maximizing the Evidence Lower Bound (ELBO):

$$\begin{aligned}\mathcal{L} = \langle \log\left(p(\mathcal{D}|\mathcal{S},\mathcal{R})\right)\rangle_{q(\mathcal{S},\mathcal{X},R^*,U)} - KL\left(q(U)||p(U)\right)) - KL\left(q(\mathcal{X})||p(\mathcal{X})\right))\\ - KL\left(q(R^*)||p(R^*)\right) + \log\left(p(R)\right), \end{aligned}\tag{4.15}$$

where we use the notation $\langle\, \cdot\, \rangle_{q(\cdot)}$ to denote the expectation with respect to $q(\cdot)$.

The challenging term is the expected log-likelihood term, which is given by:

$$\begin{aligned}\langle \log\left(p(\mathcal{D}|\mathcal{S},\mathcal{R})\right)\rangle_{q(\mathcal{S},\mathcal{X},R^*,U)} = \sum_{i=1}^{N}\sum_{j=1}^{M}\left[2d_{ij}r_i^T\langle s_{ij}\rangle_{q(\mathcal{S},\mathcal{X},U)} - \langle s_{ij}^T r_i r_i^T s_{ij}\rangle_{q(\mathcal{S},\mathcal{X},U)}\right]\\ + \sum_{i=1}^{N^*}\sum_{j=1}^{M}\left[d_{ij}^*\langle {r_i^*}^T\rangle_{q(r_i^*)}\langle s_{ij}^*\rangle_{q(s_{ij}^*,X^*,U)} - \langle {s_{ij}^*}^T\langle r_i^*{r_i^*}^T\rangle_{q(r_i^*)}s_{ij}^*\rangle_{q(s_{ij}^*,X^*,U)}\right] + B,\end{aligned}\tag{4.16}$$

where B contains all the terms which do not depend on the variational parameters.

The derivation can be continued by using the results that

$$\langle {r_i^*}^T\rangle_{q(r_i^*)} = \mu_i^r \qquad \langle r_i^*{r_i^*}^T\rangle_{q(r_i^*)} = \mu_i^r{\mu_i^r}^T + \Sigma_i^r,$$

where μ_i^r and Σ_i^r are the mean and covariance of $q(r_i^*|\alpha_i)$. Similarly,

$$\langle s_{ij}\rangle_{q(S|X,U)} = \mu_{ij}^s$$

$$\langle s_{ij}^T r_i^* r_i^{*\,T} s_{ij} \rangle_{q(S,R^*|X,U)} = \mu_{ij}^{f\,T} \left(\mu_i^r \mu_i^r + \Sigma_i^r\right) \mu_{ij}^{f\,T} + Tr\left(K_{ij}^s \left(\mu_i^r \mu_i^{r\,T} + \Sigma_i^r\right)\right),$$

where $\mu_{ij}^s \in \mathbb{R}^C$ and $K_{ij}^s \in \mathbb{R}^{C\times C}$ are the mean and covariance for the Gaussian Process conditional on U for measurement i at location j. As s_{ij} is a vector of C concatenated independent Gaussian Processes, the expressions for μ_{ij}^f and K_{ij}^s depend on all of the values of u_c together. To write these expressions we use the vectorized form of U, referred to as u in this paper, which has the prior distribution:

$$u \sim N(0, K_{VV}) \tag{4.17}$$

where K_{VV} is a $LC \times LC$ block diagonal matrix:

$$K_{VV} = \text{block}(K_{V_1V_1}^1, \ldots, K_{V_CV_C}^C)$$

As s_{ij} and u are jointly normally distributed with

$$\mu_{ij}^s = K_{(x_i,\lambda_j)V} K_{VV}^{-1} u \tag{4.18}$$

$$K_{ij}^s = K_{(x_i,\lambda_j)(x_i,\lambda_j)} - K_{(x_i,\lambda_j)V} K_{VV} K_{(x_i,\lambda_j)V}^T, \tag{4.19}$$

where the covariance between s_{ij} and u is a $C \times LC$ block diagonal matrix with the expression

$$K_{(x_i,\lambda_j)V} = \text{block}\left(K_{(x_i,\lambda_j)V_1}^1, \ldots K_{(x_i,\lambda_j)V_C}^C\right)$$

and the covariance of s_{ij} is the $C \times C$ matrix

$$K_{(x_i,\lambda_j),(x_i,\lambda_j)} = \text{block}\left(k_1\Big((x_i,\lambda_j),(x_i,\lambda_j)\Big), \ldots, k_c\Big((x_i,\lambda_j),(x_i,\lambda_j)\Big)\right).$$

Plugging in the optimal form of $q(U)$ (see Appendix D.2.1) it can be shown that the ELBO is:

$$\begin{aligned}\mathcal{L} = \frac{1}{2\sigma^4}\xi_1^T\left(\frac{1}{\sigma^2}\xi_2 + K_{VV}\right)^{-1}\xi_1 + \frac{1}{2}\log(\det(K_{VV})) - \frac{1}{2}\log\left(\det\left(\frac{1}{\sigma^2}\xi_2 + K_{VV}\right)\right) \\ - \frac{1}{2\sigma^2}\xi_0 + \frac{1}{2\sigma^2}Tr\left(K_{VV}^{-1}\xi_2\right) - \frac{NM}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(\sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij}^2 + \sum_{i=1}^{N^*}\sum_{j=1}^{M} d_{ij}^{*\,2}\right) \\ + \log(p(R)) - KL\left(q(R^*)||p(R^*)\right) - KL\left(q(\mathcal{X})||p(\mathcal{X})\right) \quad (4.20)\end{aligned}$$

where the quantities $\xi_0 \in \mathbb{R}$, $\xi_1 \in \mathbb{R}^{C\times LC}$ and $\xi_2 \in \mathbb{R}^{LC\times LC}$ are analogous to the quantities often referred to as ψ_0, ψ_1 and ψ_2 in the Bayesian GPLVM literature, but in our setting also depend on the weights r_i. These are given by:

$$\xi_0 = \sum_{i=1}^{N}\sum_{j=1}^{M}\left\langle Tr\left(r_i r_i^T K_{(x_i\lambda_j)(x_i\lambda_j)}\right)\right\rangle_{q(x_i)} + \sum_{i=1}^{N^*}\sum_{j=1}^{M}\left\langle Tr\left(\left(\mu_i^r \mu_i^{r\,T} + \Sigma_i^r\right)K_{(x_i^*\lambda_j)(x_i^*\lambda_j)}\right)\right\rangle_{q(x_i^*)}$$

$$\xi_1 = \sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij} r_i^T\left\langle K_{(x_i\lambda_j)V}\right\rangle_{q(h)} + \sum_{i=1}^{N^*}\sum_{j=1}^{M} d_{ij}^* \mu_i^{r^*\,T}\left\langle K_{(x_i^*\lambda_j)V}\right\rangle_{q(x_i^*)}$$

$$\xi_2 = \sum_{i=1}^{N}\sum_{j=1}^{M}\left\langle K_{(x_i\lambda_j)V}^T r_i r_i^T K_{(x_i\lambda_j)V}\right\rangle_{q(x_i)} + \sum_{i=1}^{N^*}\sum_{j=1}^{M}\left\langle K_{(x_i^*\lambda_j)V}^T\left(\mu_i^{r^*}\mu_i^{r^*\,T} + \Sigma_i^{r^*}\right)K_{(x_i^*\lambda_j)V}\right\rangle_{q(x_i^*)}.$$

4.2.1 Extensions

Different distributions for the weights. The choice of the form of the distributions $q(R^*)$ and $p(R^*)$ is flexible. The only requirements are that (i) the first and second moments for a vector set of test weights r_i^* are well defined and (ii) that the KL divergence between $q(R^*)$ and $p(R^*)$ can be calculated. A simple approach, followed in this paper, is to select a variational distribution from the same family as the prior distribution. So far we have used the Dirichlet distribution but our method is also applicable to other settings, for example, where $r_{ic} \in \{0, 1\}$. In this case, where only one signal is present in each observation, we suggest using a categorical distribution for the a prior and variational distribution. This is similar to the method presented in Lázaro-Gredilla et al. [2012], but extended to include latent variables. One of the examples in Section 4.4 demonstrates the MO-GPLVM in this setting.

Independent wavelengths. In Equation (4.3) we assume that the pure component signals are correlated across wavelengths, which is desirable in spectroscopy when the pure component signals are smooth. However, for other applications there may not be covariance between the M measurement locations, or the nature of the covariance may be unknown *a-priori*. In this case, each of the measurement locations can be treated as independent, so we place an independent GP prior for each location $j = 1, \dots M$, that share the same kernel, so that Equation (4.3) becomes:

$$(s_{1jc}, \dots, s_{Mjc}) \overset{iid}{\sim} N(0, K^c_{XX}) \text{ where } [K^c_{XX}]_{ii'} = k_c(x_i, x_{i'}).$$

We call this version of our model WS-GPLVM-ind. The derivation of the ELBO for this specific case is provided in Appendix D.2.2.

4.3 Implementation

Code to reproduce the results in this paper is available at and additional details for the simulations is given in Appendix D.4.

Initialization.

Maximizing the ELBO is a highly non-convex optimization problem meaning the initialization is critical. The means of GPLVM latent variables are commonly initialized using principal component analysis (PCA) [Lawrence, 2005]. We adapt this by using PCA on the reconstruction error of the training set if it is assumed the pure signals do not vary between observations. The least squares estimate for a static signal is then

$$\hat{S} = D^T R (R^T R)^{-1} \tag{4.21}$$

leading to reconstruction error for the training dataset

$$E = D - R\hat{S}^T. \tag{4.22}$$

We perform PCA on E to get the initializations for the latent variables of the training data $\{\mu_i^x\}_{i=1}^N$. The initial variance Σ_i^x is initialized I_A for all i. The variational parameters of the test data are initialized to their prior distribution, i.e. for every i, $q(x_i) = N(0, \mathbb{I}_A)$ and $q(r_i) = Dir(r|1_C)$. The other GP hyper-parameters, $\{\beta_c,\ \gamma_c,\ \sigma_{s_c}^2\}_{c=1}^C$, are initialized with constants (Appendix D.5).

Optimization.

Optimizing all model parameters together resulted in the model not using the weights, instead relying only on the flexibility of the GPLVM. To overcome this we developed a three-step approach to encourage our model to use the weights appropriately.

1. We infer the WS-GPLVM hyper-parameters and training data latent variational parameters, μ_i^x and Σ_i^x. This entails finding the ELBO without the contribution from the test data, initializing the latent variables as described above and optimizing the parameters using gradient descent methods until convergence.

2. We then fix the positions of the training data points and the kernel parameters. We set the noise variance σ^2 to a large value and fix it. We initialise the test data's variational parameters, α_i^r, $\mu_i^{X^*}$ and $\Sigma_i^{X^*}$, to be equal to the prior and optimize the ELBO to convergence with the fixed high value of variance for the noise. Once the noise test variational parameters have been optimized we reduced σ^2 and re optimized the objective. We repeat this process until we achieve the noise levels found buy optimizing the only on the training data in step 1. At the end of this step we have a good approximate position of the training and test variational parameters.

3. Finally, all variational and WS-GPLVM hyper-parameters are jointly optimized. This joint optimization gives a the final predictions.

While all of these steps contribute to a successful procedure, the most important step is the second one in which the noise is annealed from a high to a low value. If the noise is not initially

raised to a high level, the optimization objective is very non-smooth. As a result the optimization falls into a local maximum where the highly flexible GP model fits all of the variation in the data without relying on the linear weights. This is a highly non-desirable outcome as the weights are the parameters which we will typically care the most about. The introduction of this noise annealing step was crucial to developing a method which was successful.

In addition to this method we do at least 5 random restarts, selecting the one with the highest ELBO. Appendix D.5 contains details of the initializations and random restarts.

One additional optimization trick was used for when the weights are categorical. The model is initially optimized as if the weights are continuos with a Dirichlet prior and variational distribution, then once that optimization has converged those values for the latent variable distributions are used to optimize a model with a categorical weight prior and variational distribution.

GP Kernel

The experiments in Section 4.4 use a single ARD kernel for all components. The choice of the ARD kernel was motivated for automatic latent dimensionality selection [Titsias and Lawrence, 2010], as well as an analytic computation of $\xi_i,\ i = 0, 1, 2$. The expression for this is given by

$$\begin{aligned} k_c((x_i, \lambda_j), (x_{i'}, \lambda_{j'})) &= k((x_i, \lambda_j), (x_{i'}, \lambda_{j'})) \\ &= \sigma_s^2 exp\left(-\frac{1}{2}\left(x_i - x_{i'}\right)^T \beta^{-1}\left(x_i - x_{i'}\right)\right) exp\left(-\frac{1}{2}\left(\lambda_j - \lambda_{j'}\right)^T \gamma^{-1}\left(\lambda_j - \lambda_{j'}\right)\right) \end{aligned} \quad (4.23)$$

for all values of c. This choice of kernel was made for convenience as a standard default for GPLVM models, however in Section 4.5 we discuss further research which could be undertaken into alternative kernels.

Data pre-processing

For the real datasets, there are often interferences which are not caused by the mixture model. Two common interferences are (i) an offset of the entire signal by a constant and (ii) a multiplicative factor scaling the entire data. To mitigate these effects, prior work has suggested pre-processing procedures [Rakthanmanon et al., 2013, Dau et al., 2019, Rato and Reis, 2019, Gerretzen et al., 2015, Kappatou et al., 2023].

We incorporate a pre-processing step called Standard Normal Variate (SNV) [Barnes et al., 1989] or z-normalisation [Rakthanmanon et al., 2013] before fitting any models. SNV is a non-parametric pre-processing step where for each raw observation $d'_i \in \mathbb{R}^M = (d'_{i1}, ..., d'_{iM})^T$ the observation is transformed as follows

$$y_i = \frac{d'_i - \bar{d}'_i}{\sqrt{\frac{1}{M}\sum_{j=1}^{M}(d'_{ij} - d'_i)^2}} \quad \text{where} \quad \bar{d}'_i = \frac{1}{M}\sum_{j=1}^{M} d'_{ij} \ . \tag{4.24}$$

The effect of this transform is the observations each have have mean zero and variance one across their wavelengths.

One drawback of this is that it means that the pure components for the spectra d_i will be be affected by the mixture of components, due to the fact that each pure component signal has different means and varaiances which will be combined with the mixture fractions to produce the value for d_i. This drawback is discussed more in the examples. The decision to include SNV was taken as it was found empirically it was found that the model was able to compensate for the challenges introduced by of signals varying with SNV pre-processing better than the variability left by not using SNV.

Inducing Points

We define our inducing points $v_k^x \in \mathbb{R}^A$ and $v_{k'}^\lambda \in \mathbb{R}$ for the latent and observed space respectively. Our inducing points for each component are then:

$$V_c = V = \{v_k^x, v_{k'}^\lambda\}_{k=1\ldots L_h, k'=1\ldots L_\lambda} \; \forall \; c. \tag{4.25}$$

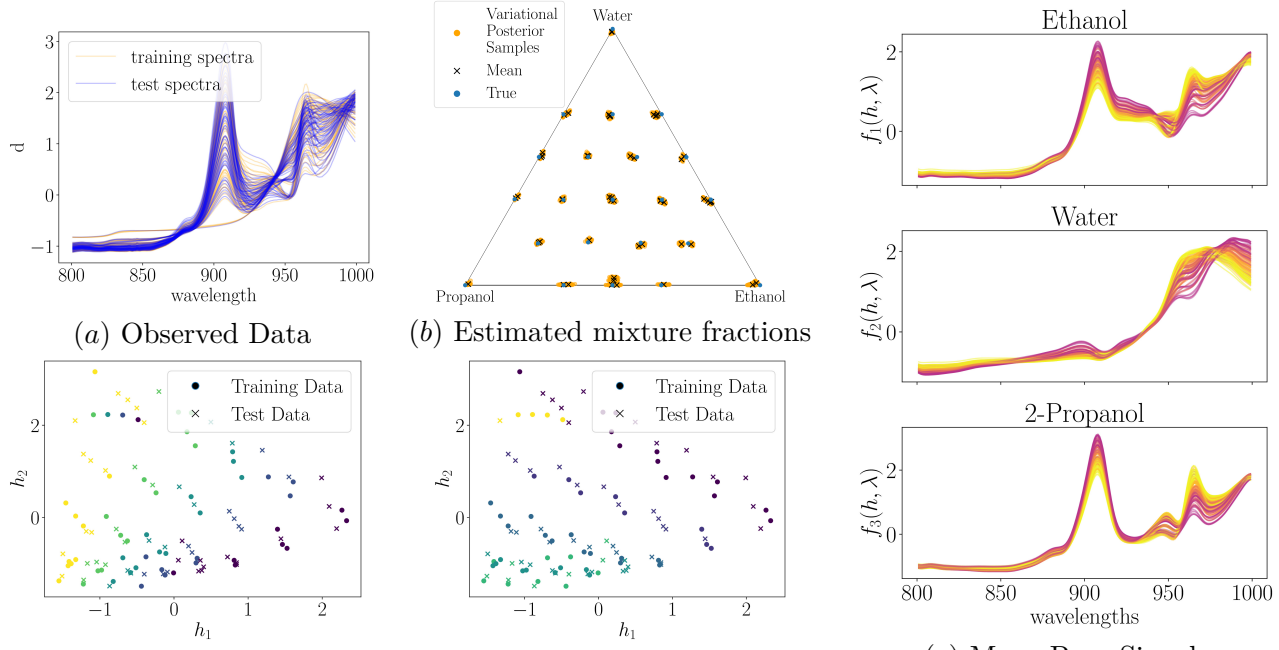
Hyperparameter and Inducing Point Initialization

Table D.3 gives full details of the hyperparameter initialisations for the WS-GPLVM and WS-GPLVM-ind for the different experiments. As we do not standardize the input (i.e. the wavelengths) the input lengthscale initialization is dependent on the range of the input. The latent variables are initialized to standard normal $x_i \sim \mathcal{N}(0, \mathbb{I}_A)$ prior, and the weights are initialized to the Dirichlet prior $r_i \sim Dir(1_c)$ for the regression settings. For the classification case, the weights are initialized to the prior, which is a uniform prior over the unit vectors.

4.4 Experiments

4.4.1 Toy Example

Our toy example simulates two chemical species with distinct peaks which shift and scale depending on a single, one-dimensional latent variable. Figure 4.2(a) illustrates the 100 training and 100 test observations with distinct peaks: the observations shift in both wavelength and change in height with changes to a single latent variable. We fit this data with a WS-GPLVM model with five latent dimensions, but the model correctly identified only one was relevant for predictions. From the dataset illustrated in Figure 4.2(a), we approximated the posterior of the pure signals ($s_c(\cdot, \cdot)$), latent variables (x) and unobserved mixture components (r^*). The Figure 4.2(b) results demonstrate that our WS-GPLVM framework can effectively learn the values of all the unobserved variables when our assumptions hold.



(*a*) Observed Data (*b*) Estimated mixture fractions

(*c*) h colored by temperature (*d*) x coloured by water fraction (*e*) Mean Pure Signals

Figure 4.3: *(a)* Shows the training and test data after SNV was applied. Each orange line represents a training data point and each blue line is a test data point. *(b)* Shows estimated mass fractions. The blue points show the true values, the black crosses show the mean of our method's variational posterior, the orange points are 100 samples drawn from the variational posterior. *(c)* and *(d)* show the latent variables coloured by temperature (purple for $30°C$, yellow for $70°C$) which we expected to cause variation in the pure signals, and water fraction (purple for no water, yellow for pure water), where the variation is induced by the normalizing procedure. *(e)* Mean estimated pure component signals for the training and test spectra, evaluated at the mean of the learned latent variable and coloured by the temperature of the measurement (red for $30°C$, yellow for $70°C$).

4.4.2 Examples on Datasets from the Literature

For the real-world examples, we compare to an inverse linear model of coregionalization model(ILMC), an approach based on Classical Least Squares (CLS/CLS-GP) and Partial Least Squares (PLS) described in Section 2.2. The CLS-GP model corresponds to WS-GPLVM without the inclusion of the latent variables h. When the covariance across inputs is removed this model corresponds to CLS.

Both ILMC and PLS are inverse models, *i.e,* rather than treating the observed spectra (Y) as an output and performing Bayesian inference, they treat the observed spectra as an input and attempt to directly predict the weights using $r_i^* = g(y_i^*) + \eta_i$, where $g(\cdot)$ is a function to be learned from training data and η_i is noise. PLS is a linear method which combines

dimensionality reduction and prediction. It is a standard tool for making predictions from spectra. The ILMC uses a MOGP to estimate $g(\cdot)$, for this, we first reduce high dimensional inputs (Y) using PCA, as GPs do not perform well with high dimensional inputs [Chen et al., 2007b, Binois and Wycoff, 2022]. The dimensionality of the PCA embedding was selected using the ELBO.

All baselines were run with five random restarts for each data split, selecting the one with the highest ELBO for ILMC and CLS/CLS-GP or cross-validation score for PLS. See Appendix D.4.1 for full details.

For regression, we analyze the models using Negative Log Predictive Density (NLPD) and the Root Mean Squared Error (RMSE) from the mean prediction to the true value of the test data. For classification, we compare predictive accuracy, Log Predictive Probability (LPP) and Area Under the Receiver Operator Curve (ROC AUC). Probabilistic predictions are not widely used in PLS [Odgers et al., 2023], so we do not report NLPD or LPP for PLS.

Near Infra Red Spectroscopy with Varying Temperatures

This dataset, from the chemometrics literature, explores the effects of changes in temperature on spectroscopic data [Wülfert et al., 1998]. The spectroscopic data, shown in Figure 4.3, consists of a mixture of three components: water, ethanol and 2-propanol. These components were mixed in 22 different ratios, including pure components for each of them. Measurements were taken at 5 temperatures between $30°C$ and $70°C$. The temperature is not given to any of the models for making predictions. We pre-processed the data as described in Section 4.3 and randomly split it into 10 training and test datasets, each with 50% of the observations used for training and 50% for testing, shown in Figure 4.3(a).

Figures 4.3(b) and (e) illustrate the performance of the proposed WS-GPLVM method in estimating the mixture fraction weights (b) and the pure component signals (e). The color of the estimated pure component signals indicates the temperature of the measurement: these variations are in agreement with the expected signals in Wülfert et al. [1998].

Figure 4.3(c) and (d) show the positions of the points in the two most significant latent variables found by the ARD kernel, colored by the temperature of the sample (in Figure 4.3(c)) or by the water fraction of the sample (in Figure 4.3(d)). Latent variable dependence on the temperature was expected, but the latent variable depending on the water fraction was a surprise because water fraction is also available in the fractional weights r_{ic}. We found this was due pre-processing that adjusted each spectrum to have a mean of zero and a variance of one, affecting samples differently based on their water mixture fractions. While this isn't directly modeled, the latent variables deal with this without interfering with the estimates for the mixture fractions.

For our comparison methods, the number of latent dimensions for PLS was selected based on a five-fold cross-validation of the training data. We used discrete Bayesian model selection to decide the number of PCA dimensions for the ILMC - for each training set eight GPs were fitted to PCA embeddings of dimensions $\{1, \ldots, 8\}$ and the one with the highest ELBO selected. Figure 4.4 shows the performance of WS-GPLVM and the baselines. WS-GPLVM had better mean accuracy with an MSE of 2.67×10^{-4}. The only comparable accuracy was PLS with an MSE of 3.04×10^{-4}. The WS-GPLVM has a much better NLPD than the ILMC, but worse than CLS-GP. This indicates the WS-GPLVM may be overconfident, which is a known problem with variational inference [Blei et al., 2017].

Oil Flow Dataset

This mixture dataset consists of simulated spectroscopic data of a mixture of oil, gas and water flowing through a pipe in different configurations, measured at two wavelengths, three horizontal and three vertical locations [Bishop and James, 1993]. This dataset has been used in the GPLVM community [Lawrence, 2005, Titsias, 2009, Urtasun and Darrell, 2007, Lalchand et al., 2022] to demonstrate clustering of the classes of flow. Using WS-GPLVM, we can directly estimate the mixture fraction.

This example has limited information on the spatial correlation of the measurements, so we use the WS-GPVLM-ind. model and consider all measurement locations independent. Additionally, the relationship between the mass fraction and the set of twelve measurements is known to be

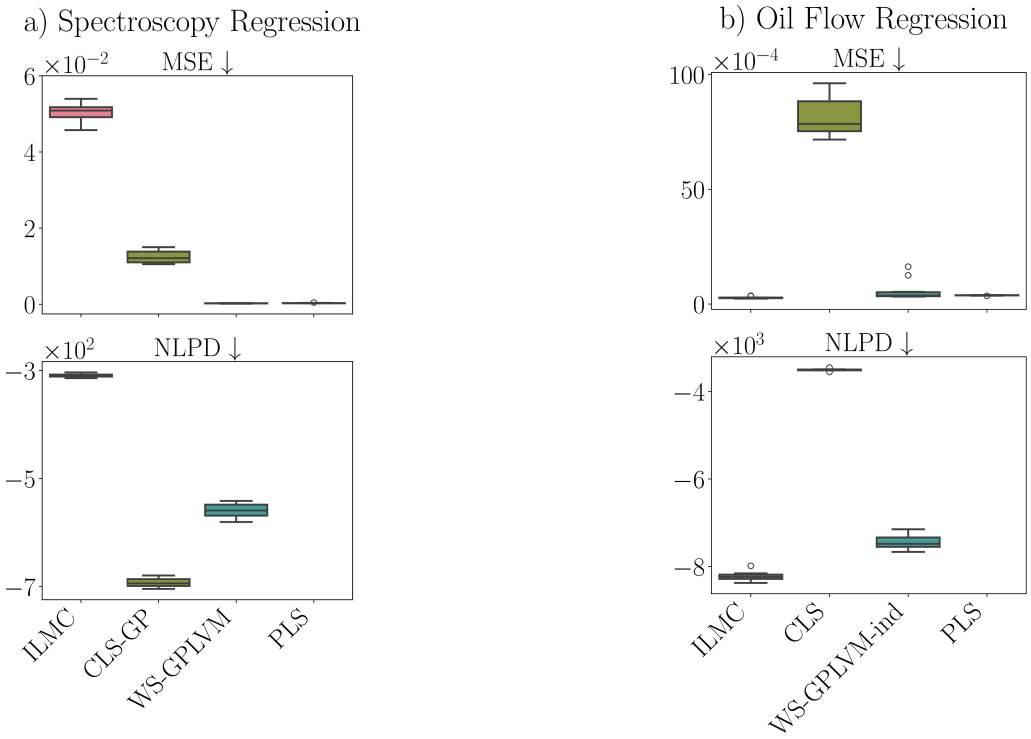


Figure 4.4: Mean squared error (MSE) and negative log predictive density (NLPD) for 10 train-test splits of the a) near infra-red spectroscopy and b) oil flow datasets.

non-linear in a more pronounced way than the previous example after applying SNV. This challenges the latent variable's ability to account for the changes in the pure component signals as the fractional composition changes.

For this example, we don't perform PCA first for the ILMC as the original dimensionality is relatively low. We use the CLS baseline without covariance across inputs as there is no clear covariance structure to use. The aim here is to predict the fractional weights of oil, gas and water. Figure 4.4 outlines the results.

In this experiment, the inverse models (PLS and ILMC) perform the best, likely due to the non-linear relationships between the weights and the spectra. However, the WS-GPLVM-ind very nearly matches the performance of the inverse models in terms of accuracy and outperforms CLS by the greatest margin of any of our experiments. This experiment demonstrates the ability of the latent variable h to greatly increase the performance of signal separation models when model misspecification could be a concern.

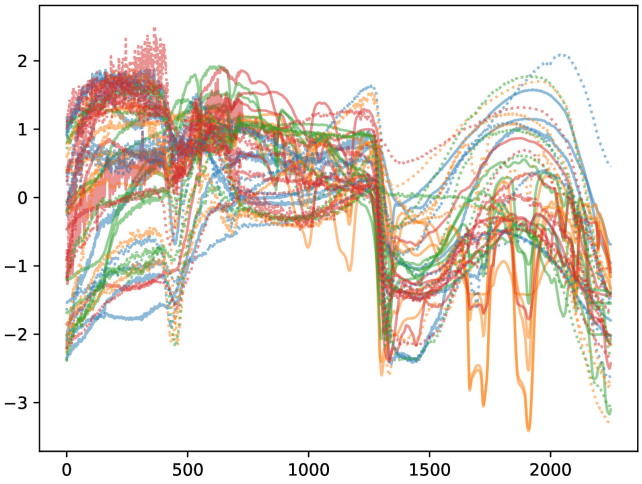


Figure 4.5: Training and test data for the remote sensing rock classification example.

Remote Sensing Rock Classification

To demonstrate how the WS-GPLVM can be used for classification, i.e. $r_{ic} \in \{0, 1\}$, we include a four class spectroscopic classification example from the UCR time series database [Baldridge et al., 2009, Dau et al., 2019]. This dataset has 70 data points, on which we did a 10-fold cross-validation to compare the different methods. For this example, we use a unit vector prior as described in Section 4.2.1. Five-fold cross-validation was used to select the number of PLS dimensions, and the ELBO was used to select the number of PCA dimensions for the ILMC, out of $\{1..8\}$. A softmax likelihood was used to produce a valid categorical prediction.

The challenge here was that the training procedure needed to be adapted to work with a small classification dataset. Each classification example does not provide information about any other class so, based on the training data, there is no reason for different classes coming from the same latent point to be similar, which is not ideal to optimize the test set. We therefore skipped the initial pre-training of the model on just the training dataset, as described in Section 4.3.

Figure 4.6 shows the results. For this dataset, WS-GPLVM and PLS are comparable, with mean accuracies of 81.4% and 82.9% and standard deviations of 21.3% and 17.9% respectively. The explicit estimate for the pure component signals in each class could be a compelling reason to choose WS-GPLVM over PLS. Note that while the ILMC method obtains a comparable to

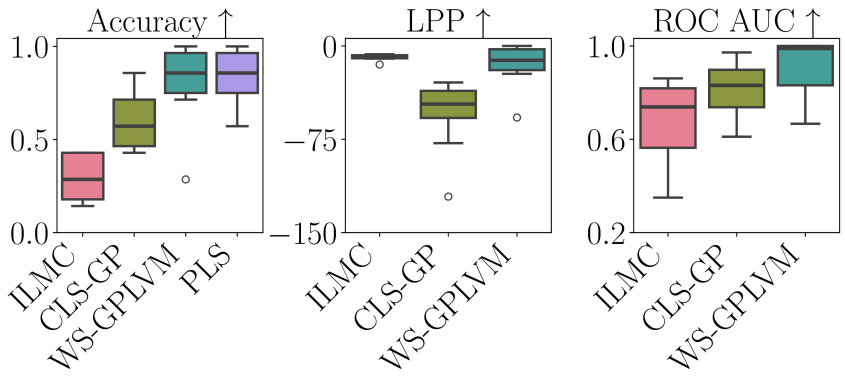


Figure 4.6: Accuracy, log predictive probability (LPP) and area under receiver operator curve (ROC AUC) for remote sensing rock classification for a 10-fold cross-validation.

WS-GPLVM, it has a very low accuracy; this is due to the ILMC predicting roughly equal weights for all classes with a high level of uncertainty, therefore not making any meaningful classification.

4.5 Discussion

This chapter was successful in demonstrating that using Weighted-Sums of GPLVMs is capable of making accurate and interpretable predictions for weights under varying of conditions. This framework is highly applicable to spectroscopic data where the summation in Equation (4.1) is a good approximation, but our approach is also effective in cases where the linear relationship between the mixture fractions and the observations does not hold exactly. We show that both regression and classification fit well within our model's framework and that our model complements existing GPLVM techniques.

An interesting extension to this work would be to investigate the effect of kernel selection on this method. Specifically, an interesting alternative would be to use an additive kernel between the wavelengths and the latent variables. This could allow information which is shared for similar wavelengths even across very different conditions, which is not currently possible with the product kernel we used. Further work on kernel selection would be interesting to investigate further.

However, the approach proposed here is challenging to work with and is difficult to recommend practically. The training procedure suggested, along with the time taken for the training, makes it difficult to recommend for most tasks. Additionally, the uncertainty estimates given by the model underestimate the true uncertainty of the model—likely due to the known shortcomings of the mean field approach to Variational Inference which is known to underestimate uncertainties Blei et al. [2017]. The contribution of the next chapter is to addresses these issues, providing a method which is both quicker, simpler and applicable to a wider range of settings.

Chapter 5

Variational Methods for Practical Probabilistic Variable Signal Spectroscopy

5.1 Introduction

In spectroscopic analysis, the Weighted-Sum Gaussian Process Latent Variable Model (WS-GPLVM) introduced in Chapter 4 demonstrated the potential to handle variable pure component signals while maintaining interpretability with the underlying physics. However, the original formulation presented three key practical limitations that this chapter addresses. First, the model required a complex training procedure involving noise annealing to ensure appropriate use of both the linear weights and non-linear latent variables. This created challenges for reproducibility and made the model difficult to apply in practice. Second, the method was restricted to datasets where all component weights were observed in the training data - a significant limitation given that spectroscopic data often contains contributions from unobserved components like humidity or light scattering effects. Finally, the uncertainty estimates from the model were found to be poorly calibrated due to limitations in the mean-field variational approximation used. fThis chapter introduces three methodological advances to address these

limitations:

- A reformulation of the Evidence Lower Bound (ELBO) using collapsed variational inference that eliminates the need for complex training procedures while improving model stability and accuracy
- An augmented model structure that handles an arbitrary number of unobserved components in both training and test data, significantly expanding the range of applications
- A structured variational inference approach that provides better-calibrated uncertainty estimates while maintaining computational tractability

We demonstrate these improvements through both synthetic examples and real-world spectroscopic datasets. The synthetic examples illustrate how the model can now automatically determine the number of unobserved components and identify whether variations are shared or independent across components. The real-world applications show significant improvements in both prediction accuracy and uncertainty quantification compared to the original WS-GPLVM formulation.

5.2 Method

5.2.1 Unobserved Components

In Chapter 4 we considered the case in which the training data contains pure component weights for all of the relevant signals present in the observed spectra. However, previous work has shown that including additional pure component signals can improve prediction of the chemical species [Shi et al., 2019, García-Muñoz and Hernández Torres, 2020]. This is because even in settings where the complete chemical composition of the sample is known, additional signals, such as humidity in the air or scattering, can result in non-chemical signals which need to be accounted for to make accurate predictions. Therefore in this chapter we consider additional signals which are not observed even in the training dataset.

We approach this by splitting the training pure component weight vector $r_i \in \mathbb{R}^C$ to be the concatenation of a set of observed pure component weights $\hat{r}_i \in \mathbb{R}^{\hat{C}}$ and a set of unobserved weights $\hat{r}_i \in R^{\hat{C}}$, such that $r_i^T = (\hat{r}_i^T, \bar{r}_i^T)$. We similarly split the vector of pure component signals into a set of signals associated to the observed and unobserved weights respectively, so $s_{ij}^T = (\hat{s}_{ic}^T, \bar{s}_{ic}^T)$. We use the terms *observed* and *unobserved* throughout this chapter to differentiate between components where the weights are observed in training, *i.e.* unobserved refers to the components $\{1, \ldots, \hat{C}\}$ and unobserved refers to $\{(\hat{C}+1), \ldots, C\}$.

This is distinct from the difference between training and test data. Training and test data consist of separate observations d_i and d_i^*, where in the test data none of the pure component weights are observed. Observed and unobserved components indicate different groups of species causing variation in the signal. The observed components are the pure component weights are known in the training data, but still need to be inferred in the the test data. We also highlight once again that despite the use of training and test, all of the data is used when optimizing the ELBO.

Given this, we assume that the data is generated according to

$$d_{ij} = \sum_{c=1}^{\hat{C}} \hat{r}_{ic}\hat{s}_{ijc} + \sum_{c=1}^{\bar{C}} \bar{r}_{ic}\bar{s}_{ijc} + \epsilon_{ij}, \tag{5.1}$$

where $\epsilon_{ij} \sim N(0, \sigma)$. We will use the hat $(\hat{\cdot})$ and bar $(\bar{\cdot})$ throughout the chapter when it is necessary to differentiate between observed and unobserved components. The absence of either of these indicates the combination of both observed and unobserved components.

Explicitly, we assume that we have observed a set of training spectra at a set of M wavelengths, ($D = \{d_i \in \mathbb{R} | i = 1, \ldots, N\}$ where $d_i = \{d_{ij} | j = 1, \ldots, M\}$) for which some portion of the weights are observed, $\hat{R} = \{\hat{r}_i \in \mathbb{R}^{\hat{C}} | i = 1, \ldots, N\}$. We also assume that we will observe at the same M wavelengths another set of signals ($D^* = \{d_{ij}^* \in \mathbb{R} | i = 1, \ldots, N^*\}$ where $d_i^* = \{d_{ij}^* \in \mathbb{R} | j = 1, \ldots, M\}$). Using this we seek to find the posterior distribution for the unobserved weights in the training data set ($\bar{R} = \{\bar{r}_i \in \mathbb{R}^{\bar{C}} | i = 1, \ldots, N\}$) both observed and unobserved component weights for the test data ($R^* = \{r_i^* = (\hat{r}_i^*, \bar{r}_i^*) \in \mathbb{R}^C | i = 1, \ldots, N^*\}$)

where $C = \hat{C} + \bar{C}$. In order to do this, we also need to find the posterior for the other variables needed to define the pure component spectra, *e.g.* for WS-GPLVM introduced in the last chapter, this would be $\{S, S^*, H, H^*, U\}$. In this chapter, we call this set of variables needed to define the pure component signals $\mathcal{Z}$.

In the previous Chapter we assumed that the weights were constrained to the simplex, in this chapter we do not make that assumption. Instead we begin by assuming that the weights are unconstrained, then discuss how to include a sum-to-one constraint for the observed components, if it is beneficial to the model.

5.2.2 Reformulating the ELBO

One of the contributions of this chapter is to simplify the required training procedure described in Section 4.3. As previously stated, the GPLVM model is highly flexible and therefore allows the observed spectra to be matched for an arbitrary set of weights. The motivation for including the complex training procedure in Section 4.3 was to ensure that the parameters for the variational distribution of the weights (α_i) were used, rather than using the flexible GP to account for the variation in the spectra. In this Section we describe a change in the distributions of the prior for the weights ($p(r_i)$) such that the variational distribution of the weights ($q(r_i)$) can be found analytically and therefore the variational parameters for the weights do not need to be optimized numerically. As the numerical optimization was the motivation for the complex training procedure this change allows a simpler training procedure.

As we have added additional unobserved variables in the form of the unobserved components for the training data, the variational posterior needs to be changed. In this section assume that the variational posterior is given by

$$q(\bar{R}, R^*, \mathcal{Z}) = q(\mathcal{Z}) \prod_{i=1}^{N} q(\bar{r}_i) \prod_{i=1}^{N^*} q(r_i^*). \tag{5.2}$$

We introduce a different variational posterior in Section 5.2.4 we discuss an alternative variational family. We refer to the variational family introduced here as the mean field variational

approach.

Deriving the optimal variational distribution for the weights does not require specifying the prior for the pure component spectra. We therefore use the variable $\mathcal{Z}$ introduced above when deriving the ELBO to indicate that the optimal weights derived here are valid for many model choices on the pure component spectra. For example, we could also have used alternatives such as VAEs [Kingma and Welling, 2013], although we do not in this thesis due to the limited data available in pharmaceutical manufacturing which motivates this work. The only requirement is that the variable $\mathcal{Z}$ shares the factorization given in Equation 5.2 and that expectations of the signal of the pure component and the outer product of the pure component signal are available, as discussed in more detail in Section 5.2.3. We denote the variational posterior for this set of variables by $q(\mathcal{Z})$ and the prior by $p(\mathcal{Z})$.

As in the previous chapter, the prior distribution for the spectra and weights factorises. Putting this in the notation used for this chapter we have

$$p(r, r^*, \mathcal{Z}) = \prod_{i=1}^{N} \left[p(r_i)\right] \prod_{i=1}^{N^*} \left[p(r_i^*)\right] p(\mathcal{Z}) \tag{5.3}$$

With this the ELBO can be written out as

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{N} \Bigg[\underbrace{\sum_{j=1}^{M} \left[\langle \log \left(p\left(d_{ij} | r_i, s_{ij}\right)\right) \rangle_{q(\bar{r}_i, \mathcal{Z})} - KL\left(q(\bar{r}_i) || p(\bar{r}_i)\right)\right]}_{\mathcal{F}_i} + \log\left(p(\hat{r}_i)\right) \Bigg] \\ + \sum_{i=1}^{N} \Bigg[\underbrace{\sum_{j=1}^{M} \left[\langle \log \left(p\left(d_{ij}^* | r_i^*, s_{ij}^*\right)\right) \rangle_{q(r_i^*, \mathcal{Z})} \right] - KL\left(q(r_i^*) || p(r_i^*)\right)}_{\mathcal{F}_i^*} \Bigg] - KL\left(q(\mathcal{Z}) || p(\mathcal{Z})\right).\end{aligned} \tag{5.4}$$

By writing the variational ELBO in this way it is clear that for any set distribution $q(\mathcal{Z})$ the ELBO is optimized by optimizing each of the terms $\mathcal{F}_i$ and $\mathcal{F}_i^*$ individually.

Of these two terms the term $\mathcal{F}_i^*$ is more straightforward to optimize so we address it first. Using

the likelihood introduced in Equation (5.1), $\mathcal{F}_i^*$ has the form

$$\begin{aligned}\mathcal{F}_i^* &= \int -\frac{1}{2\sigma^2}\sum_{j=1}^{M}\left(d_{ij}^* - {s_{ij}^*}^T r_i^*\right)^2 q(\mathcal{Z}, r_i^*)\, d\mathcal{Z}dr_i^* - KL\left(q(r_i^*)||p(r_i^*)\right) - \frac{M}{2}\log\left(2\pi\sigma^2\right) \\ &= \int \left(\underbrace{\left(\frac{1}{\sigma^2}\sum_{j=1}^{M} d_{ij}^* \langle {s_{ij}^*}^T\rangle_{q(\mathcal{Z})}\right)}_{b_i^*} r_i^* - \frac{1}{2}{r_i^*}^T \underbrace{\left(\frac{1}{\sigma^2}\sum_{j=1}^{M}\langle s_{ij}^* {s_{ij}^*}^T\rangle_{q(\mathcal{Z})}\right)}_{A_i^*} r_i^* \right) q(r_i^*)dr_i^* \\ &\quad - \frac{1}{2\sigma^2}\sum_{j=1}^{M} {d_{ij}^*}^2 - KL\left(q(r_i^*)||p(r_i^*)\right) - \frac{M}{2}\log\left(2\pi\sigma^2\right).\end{aligned} \tag{5.5}$$

Given this, we need to use a prior distribution $p(r_i)$, such that the variational posterior $q(r_i)$ can be analytically optimized. We wish to apply the approach introduced by King and Lawrence [2006] to find the optimal distribution for $q(r_i)$ analytically then substitute this back into expression for the ELBO, resulting in an objective which is making optimal use of the pure component weights as desired. The simplest distribution for which there is a closed form solution to this is $p(r_i^*) = N(0, \Sigma_0^*)$. We present a method to reintroduce the sum-to-one constraint later in this section.

Using this distribution for $p(r_i^*)$, the optimal variational posterior is

$$q(r_i^*) = N\left(r_i^* \,\middle|\, \left(A_i^* + {\Sigma_0^{r^*}}^{-1}\right)^{-1} b_i^*, \left(A_i^* + {\Sigma_0^{r^*}}^{-1}\right)^{-1}\right), \tag{5.6}$$

which can be substituted into Equation (5.5) to give a lower bound which no longer requires a numerical optimization of $q(r_i^*)$. Making this substitution gives

$$\mathcal{F}_i^* = \frac{1}{2}{b_i^*}^T\left(A_i^* + {\Sigma_0^{r^*}}^{-1}\right)^{-1} b_i^* + \log\left(\det(A_i^* + {\Sigma_0^{r^*}}^{-1})^{-1}\right) - \frac{M}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{M} {d_{ij}^*}^2. \tag{5.7}$$

For the training data we need to differentiate between the observed and unobserved components.

Noting this we can write

$$\begin{aligned}\mathcal{F}_i &= \int -\frac{1}{2\sigma^2}\sum_{j=1}^{M}\left(d_{ij} - \hat{s}_{ij}^T\hat{r}_i - \bar{s}_{ij}^T\bar{r}_i\right)^2 q(\mathcal{Z},\bar{r}_i)\, d\mathcal{Z}d\bar{r}_i - KL\left(q(\bar{r}_i)||p(\bar{r}_i)\right) - \frac{M}{2}\log\left(2\pi\sigma^2\right)\\ &= \int \left[\underbrace{\frac{1}{\sigma^2}\left(\sum_{j=1}^{M} d_{ij}\langle\bar{s}_{ij}^T\rangle_{q(\mathcal{Z})} - \hat{r}_i^T\left(\sum_{i=1}^{M}\langle\hat{s}_{ij}\bar{s}_{ij}^T\rangle_{q(\mathcal{Z})}\right)\right)}_{b_i}\bar{r}_i - \frac{1}{2}\bar{r}_i^T\underbrace{\left(\frac{1}{\sigma^2}\sum_{j=1}^{M}\langle\bar{s}_{ij}\bar{s}_{ij}^T\rangle_{q(\mathcal{Z})}\right)}_{A_i}\bar{r}_i\right] q(\bar{r}_i)d\bar{r}_i\\ &\quad - KL\left(q(\bar{r}_i)||p(\bar{r}_i)\right) - \frac{M}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{j=1}^{M} d_{ij}^2\\ &\quad + \frac{1}{\sigma^2}\left(\sum_{j=1}^{M} d_{ij}\langle\hat{s}_{ij}^T\rangle_{q(\mathcal{Z})}\right)\hat{r}_i - \frac{1}{2\sigma^2}\hat{r}_i^T\left(\sum_{j=1}^{M}\langle\hat{s}_{ij}\hat{s}_{ij}^T\rangle_{q(\mathcal{Z})}\right)\hat{r}_i.\end{aligned} \tag{5.8}$$

Note that this has a similar quadratic form structure as Equation (5.5). Therefore, provided we assume that the unobserved components have prior distribution $p(\bar{r}_i) = N(\bar{r}_i|0,\Sigma_0^{\bar{r}})$, we can write the optimal distribution for the variational posterior in terms of A_i, b_i. This is given by

$$q(\bar{r}_i) = N(\bar{r}_i|(A_i + \Sigma_0^{\bar{r}\,-1})^{-1}b_i, (A_i + \Sigma_0^{\bar{r}\,-1})^{-1}). \tag{5.9}$$

With this distribution the optimal value is

$$\begin{aligned}\mathcal{F}_i^* &= \frac{1}{2}b_i^T\left(A_i^* + \Sigma_0^{r^*\,-1}\right)^{-1}b_i + \log\left(\det(A_i + \Sigma_0^{r^*\,-1})^{-1}\right) - \frac{M}{2}\log(\sigma^2)\\ &\quad - \frac{1}{2\sigma^2}\sum_{j=1}^{M} d_{ij}^2 + \frac{1}{\sigma^2}\left(\sum_{j=1}^{M} d_{ij}\langle\hat{s}_{ij}^T\rangle_{q(\mathcal{Z})}\right)\hat{r}_i - \frac{1}{2\sigma^2}\hat{r}_i^T\left(\sum_{j=1}^{M}\langle\hat{s}_{ij}\hat{s}_{ij}^T\rangle_{q(\mathcal{Z})}\right)\hat{r}_i.\end{aligned} \tag{5.10}$$

Sum-to-One Constraints

The Dirichlet distribution used as a prior in the previous chapter ensured that the pure component weights summed to one and were strictly positive. Unfortunately it is challenging to enforcing both the sum-to-one and strictly positive constraints while analytically optimising the variational posterior. Here, we focus on introducing a prior that encodes the sum-to-one constraint. This prior choice naturally leads to a variational distribution on the weights that

also encodes the sum-to-one constraint. While we do not enforce the weights to be strictly positive in the variational posterior, we observe in the experiment section that the variational posterior assigns a small probability almost all data points in most real data sets.

One important feature to note is that the sum-to-one constraint is generally only applied to the set of observed components. This is because the sum-to-one constraint generally arises from a known set of chemical concentrations making up a complete drug product, which would be the observed components in our scenario. In our setting the sum-to-one constraint should not apply to additional unobserved components as these components are there to represent interfering signals, for example humidity in the air, which are not included in the sum-to-one constraint.

The approach which we take is to define a new variable with one fewer dimension than the original weights. We relate this new variable to the weights via a linear transformation such that every point in the lower dimensional space satisfies the sum-to-one constraint when transformed to the weight space. As the linear transformation of a normal distribution is still normally distributed we maintain an analytic solution for the variational posterior of the weights.

In our use case the sum-to-one constraint is only applicable to $\hat{r}_i$, as it arises when a sample has been made up by a practitioner with known chemical components. As $\hat{r}_i$ are known for the training data there is no change to the term $\mathcal{F}_i$. For the test data ($\hat{R}^* = \{\hat{r}_i \in \mathbb{R}^{\hat{C}} | i = 1, \ldots, N^*\}$) resulting in a change to Equation (5.7).

Specifically we assume that the random variable $\hat{r}_i^* \in \mathbb{R}^{\hat{C}}$ is given by a linear transformation of a random variable $\hat{\rho}_i^* \in \mathbb{R}^{(\hat{C}-1)}$ by

$$\hat{r}_i^* = \hat{W}_{\hat{C}} \hat{\rho}_i^* + \hat{m} \tag{5.11}$$

where $\hat{W}_{\hat{C}} \in \mathbb{R}^{\hat{C} \times (\hat{C}-1)}$ and $\hat{m} \in \mathbb{R}^{\hat{C}}$ are fixed quantities. To ensure that the sum-to-one constraint is satisfied, we enforce $\sum_{c=1}^{\hat{C}} \hat{m}_c = 1$ and for each row l of $\hat{W}_{\hat{C}}$, we have $\sum_{c=1}^{\hat{C}} \hat{w}_{cl} = 0$. The value for $\hat{m}$ could be any point that satisfies $\sum_c \hat{m}_c = 1$, but we will use $\hat{m} = (\frac{1}{\hat{C}}, \ldots, \frac{1}{\hat{C}})^T$. We want $\hat{W}_{\hat{C}}$ to be isotropic with respect to changes in the value in ρ_i^* so we do not distort the prior $p(\rho_i^*)$. We enforce this constraint by requiring that each row of $\hat{W}_{\hat{C}}$ has the same L2

norm.

A matrix $\hat{W}_{\hat{C}}$ which satisfies these properties can be defined iteratively as follows. First, note that if we only have $\hat{C} = 2$ then the constraints are satisfied by

$$\hat{W}_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}. \tag{5.12}$$

Then we note that if we have a value for $\hat{W}_{C-1}$ we can find a value for $\hat{W}_C$ which satisfies the constraints using

$$\hat{W}_C = \left(\begin{array}{ccc:c} & & & -\frac{1}{C-1} \\ & \sqrt{1-\frac{1}{(C-1)^2}}\,\hat{W}_{C-1} & & \vdots \\ & & & -\frac{1}{C-1} \\ \hdashline 0 & \dots & 0 & 1 \end{array}\right). \tag{5.13}$$

This allows a matrix where all of the rows columns sum to one and the L2 norm of the rows is one. Finally we note that this procedure produces a matrix whose $(C-1)$ non-zero eigenvalues are equal to $\frac{C+1}{C}$. We therefore scaled by a factor of $\sqrt{\frac{C}{C+1}}$ to set all non-zero eigenvalues equal to C.

In order to derive the ELBO this transformation for $\hat{r}_i^*$, needs to be combined with the weights $\bar{r}_i^*$, which are still assumed to be unconstrained. We achieve this by defining the new variables

$$W = \begin{pmatrix} \hat{W}_{\hat{C}} & 0 \\ 0 & \mathbb{I}_{\bar{C}} \end{pmatrix} \qquad \rho_i = \begin{pmatrix} \hat{\rho}_i \\ \bar{r}_i^* \end{pmatrix} \quad \text{and} \quad m = \begin{pmatrix} \hat{m} \\ 0_{\bar{C}} \end{pmatrix}. \tag{5.14}$$

With these variables pure component weights are given by

$$r_i = W\rho_i + m. \tag{5.15}$$

With this transformation, the new random variable is ρ_i^*, for which we use the prior $p(\rho_i^*) =$

$N(\rho_i^*|0, \Sigma_0^{\rho^*})$. The ELBO for the test data is given by

$$\begin{aligned}\mathcal{F}_i^* &= \int \left(b_i^{*\,T} (W\rho_i^* + m) - \frac{1}{2} (W\rho_i^* + m)^T A_i^* (W\rho_i^* + m) \right) q(\rho_i^*) d\rho_i^* \\ &\quad - \frac{1}{2\sigma^2} \sum_{j=1}^{M} d_{ij}^{*\,2} - KL\left(q(\rho_i^*)||p(\rho_i^*)\right) - \frac{M}{2} \log\left(2\pi\sigma^2\right) \\ &= \int \left(\left(b_i^{*\,T} - m^T A_i^* \right) W\rho_i^* - \frac{1}{2} \rho_i^{*\,T} W^T A_i^* W \rho_i^* \right) q(\rho_i^*) d\rho_i^* \\ &\quad - \frac{1}{2\sigma^2} \sum_{j=1}^{M} d_{ij}^2 - \frac{M}{2} \log\left(2\pi\sigma^2\right) + b_i^{*\,T} m - \frac{1}{2} m^T A_i^* m - KL(q(\rho_i^*)||p(\rho_i^*))\end{aligned} \tag{5.16}$$

and the optimal distribution for $q(\rho_i^*)$ is given by

$$q(\rho_i) = N(r_i|(W^T A_i W + \Sigma_0^{\rho\,-1})^{-1} W^T (b_i^* - A_i^* m), (W^T A_i W + \Sigma_0^{\rho\,-1})), \tag{5.17}$$

for which

$$\begin{aligned}\mathcal{F}_i^* =& \frac{1}{2} (b_i^{*\,T} - m^T A_i^*) W \left(W^T A_i^* W + \Sigma_0^{\rho\,-1} \right)^{-1} W^T (b_i^* - A_i^* m) \\ &+ \log\left(\det(W^T A_i^* W + \Sigma_0^{\rho\,-1})^{-1} \right) - \frac{M}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{M} d_{ij}^{*\,2} + b_i^{*\,T} m - \frac{1}{2} m_i^T A_i^* m_i.\end{aligned} \tag{5.18}$$

Variational distributions can easily be transformed back to make predictions for r_i^* by noting that a linear transformation of a multivariate normal random variable still follows a multivariate normal distribution.

5.2.3 Pure Component Spectra

In this section we describe how to take the expectations with respect to the variational posterior and calculate the KL divergence for the variables needed to define the component spectra. We do not differentiate between observed and unobserved components in this section. This is because, while we need several different expectations to calculate the ELBO in Equation (5.4) ($\langle \hat{s}_{ij} \rangle_{q(\mathcal{Z})}$, $\langle \bar{s}_{ij} \rangle_{q(\mathcal{Z})}$, $\langle \hat{s}_{ij} \hat{s}_{ij}^T \rangle_{q(\mathcal{Z})}$, $\langle \hat{s}_{ij} \bar{s}_{ij}^T \rangle_{q(\mathcal{Z})}$ and $\langle \bar{s}_{ij} \bar{s}_{ij}^T \rangle_{q(\mathcal{Z})}$), all these quantities can be related

to the expectations $\langle s_{ij} \rangle_{\mathcal{Z}}$ and $\langle s_{ij} s_{ij}^T \rangle_{\mathcal{Z}}$. This is done via the relations

$$\langle s_{ij} \rangle_{q(\mathcal{Z})} = \begin{pmatrix} \langle \hat{s}_{ij} \rangle_{q(\hat{s}_{ij})} \\ \langle \bar{s}_{ij} \rangle_{q(\bar{s}_{ij})} \end{pmatrix} \tag{5.19}$$

and

$$\langle s_{ij} s_{ij}^T \rangle_{q(\mathcal{Z})} = \begin{pmatrix} \langle \hat{s}_{ij} \hat{s}_{ij}^T \rangle_{q(\hat{s}_{ij})} & \langle \hat{s}_{ij} \bar{s}_{ij}^T \rangle_{q(\hat{s}_{ij}, \bar{s}_{ij})} \\ \langle \bar{s}_{ij} \hat{s}_{ij}^T \rangle_{q(\hat{s}_{ij}, \bar{s}_{ij})} & \langle \bar{s}_{ij} \bar{s}_{ij}^T \rangle_{q(\bar{s}_{ij})} \end{pmatrix}. \tag{5.20}$$

Therefore, to simplify notation in this section, we do not differentiate between observed and unobserved components. In order to calculate the ELBOs the expectations calculated here must be split into the relevant terms. Note that this differentiation between observed and unobserved components is not required for the test data as both are treated equivalently in that case.

Static signals

The simplest possible prior for the pure component spectra, the term $p(\mathcal{Z})$ in Equation (5.4), assumes that the signal does not vary for each data point. This assumes that $s_{ijc} = s_{ijc}^* = s_{jc}$ for all values of i. In this we have $\mathcal{Z} = \{s_{jc} | j = 1, \ldots, M, c = 1, \ldots, C\}$. For this we use a prior $p(\mathcal{Z}) = \prod_{j=1}^{M} N(s_{jc} | 0, \mathbb{I}_c)$ and the variational posterior is assumed to be of the form $q(\mathcal{Z}) = \prod_{j=1}^{M} N(s_{jc} | \mu_{jc}^s, \Sigma_{jc}^s)$. We refer to this approach as the static signal approach in our comparison methods. With these quantities, the expectations $\langle s_{ij} \rangle_{q(\mathcal{Z})} = \mu_j^s$ for all j and $\langle s_{ij} s_{ij}^T \rangle_{q(\mathcal{Z})} = \mu_j^s \mu_j^{s\,T} + \Sigma_j^s$ for all j.

The KL divergences are straightforward to calculate for the variational family and prior which we have chosen.

Variable signals with WS-GPLVMs

In order to allow variable signals we follow the same approach as in Chapter 4 where we use a GPLVM. The full model is both motivated and described in detail in Chapter 4, but we restate

it briefly here and describe how to take the relevant expectations.

For this we have a model in which we assume that the unobserved factors driving variation in the pure component spectra are modelled by some latent variable $h_i \in \mathbb{R}^A$. In this chapter we treat each wavelength as being independent so we consider the relationship between the latent variable and the pure component signals to be modelled by MC independent mappings $s_{jc}(\cdot) : \mathbb{R}^A \to \mathbb{R}$ such that

$$s_{ijc} = s_{jc}(h_i). \tag{5.21}$$

We place a spherical normal prior on the latent space so $p(h_i) = N(h_i|0, \mathbb{I}_A)$ and a GP prior on the mapping so $p(s_{ijc}) = N(s_{ijc}|0, K_{HH})$, where K_{HH} is the kernel matrix for the kernel in Equation (5.36).

In order to reach a tractable variational bound we need to introduce a set of L inducing points at locations $V = \{v_{jc} \in \mathbb{R}^{L\times A} | j = 1, \ldots, M,\ c = 1, \ldots, C\}$, with values $U = \{u_{jc} \in \mathbb{R}^L | j = 1, \ldots, M,\ c = 1, \ldots, C\}$. The prior of the inducing points is $p(U) = \prod_{j=1}^{M} \prod_{c=1}^{C} N(u_{jc}|0, K_{VV})$. We then condition the observations on the inducing points $p(s_{ijc}|u_{ijc}) = N(s_{ijc}|\mu^s_{ijc}, K^s_{ijc})$, where μ^s_{ijc} and K^s_{ijc} are defined in Equations (4.9) and (4.10). These combined variables give the quantity $\mathcal{Z} = \{U, H, H^*, S, S^*\}$.

These variables require defining a variational posterior for $q(\mathcal{Z}) = q(S, U, H, H^*)$. We assume that

$$q(S, S^*, H, H^*, U) = q(S, S^*|U, H, H^*)q(H, H^*)q(U). \tag{5.22}$$

As in Chapter 4, we assume that the variational family for the latent variables are $q(H, H^*) = \prod_{i=1}^{N} N(h_i|\mu^h_i, \Sigma^h_i) \prod_{i=1}^{N^*} N(h^*_i|\mu^{h^*}_i, \Sigma^{h^*}_i)$. In addition we assume that the conditional distributions are the same as that of the prior so $q(S, S^*|U, H, H^*) = \prod_{j=1}^{M} \prod_{c=1}^{C} p(s_{ijc}, s^*_{ijc}|h_i, h^*_i, u_{jc})$ which has a closed form expression given in Equation (4.9) and (4.10).

A difference to Chapter 4 is that here we need to maintain an explicit representation of the distribution of $q(U)$. This is very reminiscent of the switch from the collapsed bound to the uncollapsed bound for GPs [Hensman et al., 2013]. We chose the variational family $q(U) =$

$\prod_{j=1}^{M} N(u_j|\mu^u, \Sigma^u)$, where $u_j = (u_{j1}^T, \ldots, u_{jC}^T)^T \in \mathbb{R}^{LC}$, $\mu^u \in \mathbb{R}^{LC}$, and $\Sigma^u \in \mathbb{R}^{LC \times LC}$.

The two expectations that we are interested in are given by

$$\langle s_{ij} \rangle_{q(\mathcal{Z})} = \langle K_{h_i V} \rangle_{q(h_i)} K_{VV} \mu_j^u \tag{5.23}$$

and

$$\langle s_{ij} s_{ij}^T \rangle_{q(\mathcal{Z})} = \langle K_{h_i h_i} \rangle_{q(h_i)} - \left\langle K_{h_i V} K_{VV}^{-1} \left(\mu_j^u \mu_j^u + \Sigma_j^u - K_{VV} \right) K_{VV}^{-1} K_{h_i V}^T \right\rangle_{q(h_i)}. \tag{5.24}$$

The values for $\langle K_{h_i h_i} \rangle_{q(h_i)}$ and $\langle K_{h_i V} \rangle_{q(h_i)}$ are the same as those in the regular GPLVM. The final expectation can be calculated by noting that

$$K_{h_i V} = \begin{pmatrix} k_1(h_{i1}, V)^T & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & k_C(h_{iC}, V)^T \end{pmatrix}, \tag{5.25}$$

where $k_c(h_{ic}, V_c) \in \mathbb{R}^L$ is the kernel between a location h_{ic} and the inducing points V_c. Setting

$$\Psi = \begin{pmatrix} \psi_{11} & \ldots & \psi_{1C} \\ \vdots & \ddots & \vdots \\ \psi_{C1} & \ldots & \psi_{CC} \end{pmatrix} = \left(K_{VV}^{-1} \left(\mu_j^u \mu_j^u + \Sigma_j^u \right) K_{VV}^{-1} \right) \tag{5.26}$$

the expectation can be calculated as

$$\left\langle K_{h_i V} \Psi K_{h_i V}^T \right\rangle_{q(h_i)} = \begin{pmatrix} \langle k_1(h_i, V)^T \psi_{11} k_1(h_i, V) \rangle_{q(h_i)} & \ldots & \langle k_1(h_i, V)^T \psi_{1C} k_C(h_i, V) \rangle_{q(h_i)} \\ \vdots & \ddots & \vdots \\ \langle k_C(h_i, V)^T \psi_{C1} k_1(h_{i1}, V) \rangle_{q(h_i)} & \ldots & \langle k_C(h_i, V)^T \psi_{CC} k_C(h_i, V) \rangle_{q(h_i)} \end{pmatrix}. \tag{5.27}$$

For an RBF kernel all of these quantities have closed form expressions.

The other relevant expression for calculating the ELBO is the KL divergence. This can be

calculated with

$$KL(q(S,S^*,H,H^*,U)||p(S,S^*,H,H^*,U))) = \sum_{i=1}^{N} KL(q(h_i)||p(h_i)) + \sum_{i=1}^{N^*} KL(q(h_i^*)||p(h_i^*)) + \sum_{j=1}^{M} KL(q(u_j)||p(u_j)). \quad (5.28)$$

With this and the variational families we have selected for each of these quantities, all of these have closed form expressions.

5.2.4 Structured Variational Inference for WS-GPLVMs

In Section 5.2.2 we introduced a mean field approach for the variational posterior which is the approach we have worked with up to this point. That approach allowed our method can be applied to by including unobserved components and adapting the variational family to allow the variational distribution of the weights to be found optimally. In this section we introduce a method to address a different challenge of the previous chapter, namely the method severely underestimated the uncertainty of the model, resulting in very low log-predictive likelihoods. In this section we change our approach, which we refer to as the Structured approach, to attempt to address this issue.

Structured Stochastic Variational Inference [Hoffman and Blei, 2015] is an approach to avoid making the mean field assumption which is typically used in variational inference. The goal of this method is to include dependencies within the variational family which has two benefits: firstly this allows a better approximation of the posterior which should address the underestimation of the uncertainty which was present in the previous chapter, and secondly this avoids some of the non-convexity present in the ELBO [Wainwright and Jordan, 2008]. In this work we do not take advantage of the ability of the model to perform minibatching, so we refer to the variational family introduced in this section as Structured Variational Inference (SVI).

The key structure which allows this method to work is that the probabilistic family has a hierarchical structure in which a set of *local variables* are conditionally independent given a

set of *global variables*. For our setting the global variables are the inducing points (U) and the local variables are the latent variables, weights and signals ($H, R, S, H^* R^*, S^*$). The joint distribution factorizes as

$$p(U, R, R^*, S, S^*, H, H^*) = p(U)p(S, S^*|H, H^*, U) \prod_{i=1}^{N} p(h_i)p(r_i) \prod_{i=1}^{N^*} p(h_i^*)p(r_i^*). \tag{5.29}$$

In the regular mean field case which we have discussed so far the variational posterior cuts the dependence between the distribution of the local variable on the global variable. Structured Stochastic Variational Inference offers a method to avoid this by employing a variational posterior of the form

$$q(U, \bar{R}, R^*, \bar{S}, S^*, H, H^*) = q(U)p(S, S^*|H, H^*, U) \prod_{i=1}^{N} q(h_i, r_i|U) \prod_{i=1}^{N^*} q(h_i^*, r_i^*|U). \tag{5.30}$$

To work with a variational posterior of this form it is necessary to rewrite the ELBO as:

$$\mathcal{L} = \int \Bigg[\sum_{i=1}^{N} \underbrace{\Bigg[\sum_{j=1}^{M} \Big[\langle \log(p(d_{ij}|s_{ij}, r_i) \rangle_{q(s_{ij}, \bar{r}_i, h_i|U)} \Big] - KL\left(q(\bar{r}_i|U)||p(\bar{r}_i)\right) - KL(q(h_i|U)||p(h_i)) \Bigg]}_{\mathcal{G}_i(U)}$$

$$+ \sum_{i=1}^{N^*} \underbrace{\Bigg[\sum_{j=1}^{M} \Big[\langle \log(p(d_{ij}^*|s_{ij}^*, r_i^*)) \rangle_{q(s_{ij}, r_i^*, h_i^*|U))} \Big] - KL\left(q(r_i^*|U)||p(r_i^*)\right) - KL(q(h_i^*|U)||p(h_i^*)) \Bigg]}_{\mathcal{G}_i^*(U)} \Bigg] q(U) dU$$

$$- KL(q(U)||p(U)) + \sum_{i=1}^{N} log(p(\hat{r})). \tag{5.31}$$

Two important considerations are that $\mathcal{G}_i(U)$ and $\mathcal{G}_i^*(U)$ have the form of an ELBO, and will be referred to as the local ELBO. The local ELBO is defined by combining a specific value of U along with the local variational parameters: μ_i^h, Σ_i^h, $\mu_i^{\bar{r}}$ and $\Sigma_i^{\bar{r}}$ for the training data and $\mu_i^{h^*}$, $\Sigma_i^{h^*}$, $\mu_i^{r^*}$ and $\Sigma_i^{r^*}$ for the test data. Crucially, the values of the variational parameters are independent for all values of U.

The objectives G_i and G_i^* are very similar to their counterparts in the mean field case, but with two small differences. The first difference is that the expectation is conditioned on U.

This means that the KL term for the inducing points is dropped from the objective, and the expectations for the pure component signals become

$$\langle s_{ij} \rangle_{q(s_{ij},h_i|u_j)} = \langle K_{h_iV} \rangle_{q(h_i)} K_{VV} u_j \tag{5.32}$$

and

$$\langle s_{ij} s_{ij}^T \rangle_{q(s_{ij},h_i|u_j)} = \langle K_{h_ih_i} \rangle_{q(h_i)} - \left\langle K_{h_iV} K_{VV}^{-1} u_j u_j^T K_{VV}^{-1} K_{h_iV}^T \right\rangle_{q(h_i)}. \tag{5.33}$$

The second difference is that there is an additional KL term for the latent variable which needs to be considered in the optimisation.

Equation (5.31) is optimised by independently maximizing the local variational parameters for each value of the global variable. We can approximate the ELBO using Monte Carlo integration over the inducing variables, so Equation 5.31 is approximated using a set of points $\{U_p \in \mathbb{R}^{L \times M} | p = 1, \ldots, P\}$, where $U_p \sim q(U)$. This approximation gives

$$\mathcal{L} \approx \frac{1}{P} \sum_{p=1}^{P} \left[\sum_{i=1}^{N} \mathcal{G}_i(U_p) + \sum_{i=1}^{N^*} \mathcal{G}_i^*(U_p) \right] - KL(q(U)||p(U)) + \sum_{i=1}^{N} log(p(\hat{r})). \tag{5.34}$$

In order to approximate the ELBO we need to maximize

$$\max_{\mu_{ip}^h, \Sigma_{ip}^h, \mu_{ip}^r, \Sigma_{ip}^r} \quad \mathcal{G}_i(U_p) \tag{5.35}$$

for each value of i and each value of p. The method for generating U_p is described in Section 5.3.4.

5.3 Implementation Details

5.3.1 Learning Dependent or Independent Latent Variables

One assumption which we made in the previous chapter was that there was a single latent variable which controlled all of the pure component spectra. This assumption was motivated

by considering the variations in which all of the signals equally, such as change in spectrometer or change in temperature. However, some variations can cause variation in only one component such as a change in the granule size of a particular chemical component [Räsänen and Sandler, 2007].

Ideally, we would wish for the model to determine for itself which components a particular variation is affecting. This problem has been studied in the GPLVM literature in the form of Manifold Relevance Determination (MRD) [Damianou et al., 2012]. The key insight of MRD is to learn separate sets of ARD (Automatic Relevance Determination) kernel parameters for each observed data modality, while sharing a common latent space. The WS-GPLVM has a different kernel for each of the kernel functions, each of which are of the form:

$$k_c(h, h') = \sigma_{s\,c}^2 \exp\left(-\frac{1}{2}\sum_{a=1}^{A}\frac{1}{\beta_{ca}}(h_a - h'_a)^2\right) \tag{5.36}$$

where $\frac{1}{\beta_{ca}}$ represents the relevance weight for dimension a for the c-th pure component spectra, and σ_c^2 is the output scale for that modality. By learning these weights through optimization of the marginal likelihood, the model automatically discovers which latent dimensions are relevant for explaining the variations in each observation space. When $\frac{1}{\beta_{ca}}$ approaches zero, it effectively "switches off" the a-th latent dimension's influence on the c-th pure component spectra. Therefore, by allowing the ARD kernel parameters to be learnt independently it allows the model to learn which components a particular variation effects.

5.3.2 Selecting the Number of Unobserved Components

An additional hyper-parameter to be picked with this model compared to the one presented in the previous chapter is the maximum number of unobserved components in the samples. By allowing each GP kernel to have independent hyper-parameters the model can "switch off" that component by setting the variance to be zero.

5.3.3 Optimization

The training procedure for this method is much more straightforward than that of the method described in the previous chapter, avoiding the need for a three step approach to the training. Instead we optimize our model by jointly initialising the training and test data's latent variables and optimizing all kernel parameters and variational parameters jointly.

A difference to the previous chapters approach was the method for finding the initialization for the latent variables. We began the same least squares estimate of the pure component signals as was used in the previous chapter

$$\hat{S} = D^T R(R^T R)^{-1}. \tag{5.37}$$

However, we take a different approach to finding the observed signal, namely we use these estimates for the spectra to find an initial estimate for the values of R and R^*. These estimates are given by

$$\hat{R} = D\hat{S}^T \qquad \text{and} \qquad \hat{R}^* = D^*\hat{S}^T \tag{5.38}$$

and these values are the ones used to estimate the unexplained portion of the observations using

$$E = D - \hat{R}\hat{S}^T \qquad \text{and} \qquad E^* = D^* - \hat{R}^*\hat{S}^T. \tag{5.39}$$

The latent variables found by applying PCA to these two unexplained portions of the signal.

For initializing the distributions of the inducing points we take different approaches in the Mean Field and Structured approaches. For the structured approach we initialize the inducing points variational distribution to be the same as the prior for the inducing points. For the Mean Field approach the covariance is the same as the prior, however the mean $\mu_j^u \in \mathbb{R}^{LC}$ is initialised with $N(0, 0.01\mathbb{I}_{LC})$. This additional noise in the mean field case is required to break the symmetry between multiple unobserved components - if the mean is initialized as zero then the gradient is zero and training using gradient descent fails.

The model is initially optimised with Adam, then using LBFGS when the model is close to

an optimum until the increase in the ELBO between successive iterations falls below a small constant, typically set to one.

5.3.4 Sampling the Inducing Points for Structured Variational Inference

In order to use Structured variational inference, as described in 5.2.4 we need to generate samples from $U_p \in \mathbb{R}^{M\times L\times C}$, which is described in this section. To do this it is easier to work with the vectorized form of the inducing points from a single wavelength, denoted as $u_{jp} \in \mathbb{R}^{LC}$, for $j = 1, \ldots, M$. The relation between u_p and the distribution s_{ij} are given in Equation (4.9). We work with the vectorized form of the inducing points to allow covariance between inducing point outputs for different pure component spectra.

The original work describing Stochastic Structured Variational Inference proposes draws the global parameters, an optimizes the local parameters, then redraws the global parameters for the next iteration. We approximate the ELBO by using a Monte Carlo approach, as stated in Equation (5.34). However, rather than resampling at each step of the optimization, we instead fix a set of points to approximate the variational posterior and only update these as the parameters of the variational distribution are updated.

We do this by fixing an independent set of P points for each of the M wavelengths spread over an LC dimensional unit hyper cube. We then transform them such that the points to approximate the distribution $N(0, \mathbb{I}_{LC})$ using the inverse cumulative density function transform. We call these fixed points $t_{jp} \in \mathbb{R}^{LC}$. These samples are related to samples of the vectorized inducing points via

$$u_{jp} = \mu_j^u + L_j t_{jp}, \tag{5.40}$$

where $L_j \in \mathbb{R}^{LC\times LC}$ is the Choleskey decomposition of Σ_j^u.

Each of the P samples for u_{jp} need their own set of latent variables $\{\mu_{ip}^h, \Sigma_{ip}^h | i = 1, \ldots, N,\ p = 1, \ldots, P\}$. If we redrew the samples u_{jp} we would need to optimize the positions of these

inducing points from scratch. By only updating the samples u_{jp} when the parameters μ_j^u and Σ_j^u are updated, we can update the variational parameters of the latent variables and the inducing points together using gradient descent.

To generate samples from the inducing points by sampling PM samples from from the LC dimensional hypercube from a Soubol sequence [Sobol', 1967] before assigning P of these points to each of the M wavelengths. This ensures that we both cover the unit hypercube well and that there will not be repeated bias shared between all wavelengths, as would happen if a single set of points $\{s_p|P = 1, \ldots, P\}$ was used across all wavelengths. These points in the unit hypercube are then transformed to be samples from $N(0, \mathbb{I}_{LC})$ using the inverse cumulative density function.

5.4 Related Works

The previous work on spectroscopy with variable signals is given in Chapter 4, so is not discussed in detail here. The importance of unobserved components in spectroscopy was highlighted by the EIOT method developed in Shi et al. [2019] and García-Muñoz and Hernández Torres [2020]. EIOT includes spectroscopic signals from non-chemical sources (e.g., humidity) which have their own weights and signals. These weights could be either observed or unobserved. It was found empirically that including unobserved spectral signals improved the predictive ability of their model. In this model the unobserved signals were estimated using the loadings of a PCA model based on the reconstruction error of the static signal model. Given the links between this and a generative model for the low rank normal distribution found by Principle Component Analysis [Tipping and Bishop, 1999, Bishop, 1998], our method remains close to the approach used in EIOT, with the obvious difference being that the pure component spectra of the unobserved components varying from sample to sample.

Making the distribution for the latent variables conditional on the inducing points is very closely related to the Structured Stochastic Variational Inference approach by Hoffman and Blei [2015]. This approach in turn is a version of working with *collapsed* bounds. Collapsed methods arise

outside of Variational Inference, also appearing in collapsed Gibbs sampling (see Liu [2001] for example), and were adapted for Variational Inference by Teh et al. [2006]. Collapsed bounds turn out to be closely related to the approach from King and Lawrence [2006], which shares the key idea integrating out some set of the variables of interest to reduce computational complexity and improve the upper bound. The relationship of King and Lawrence [2006] to collapsed bounds is clearly explained in Hensman et al. [2012].

Maintaining an explicit representation for the parameters of $q(U)$ in GPs was subsequently introduced by Hensman et al. [2013] in order to scale GPs to large datasets and is reffered to as the uncollapsed bound, to differentiate it from the collapsed bound of Titsias [2009]. The use of collapsed bounds for GPLVMs has been studied by Lalchand et al. [2022], allowing for both big data and non-Gaussian likelihoods.

5.5 Experiments

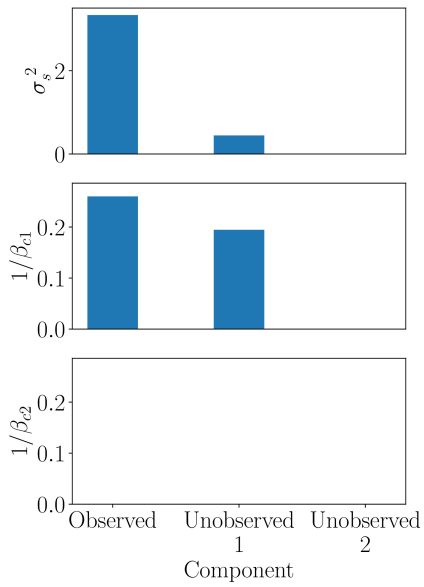
5.5.1 Toy Example with Unobserved Components

The toy example is very similar to that described in Section 4.4. The data is simulated according to

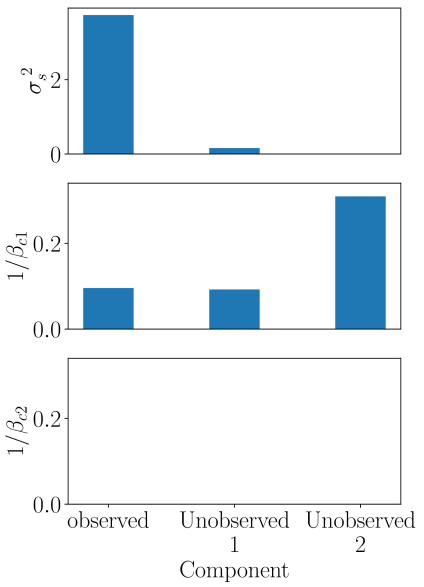
$$d_{ij} = \underbrace{(2h_{i1} + 15)g(2h_{i1} - 7.5 + \lambda_j)}_{s_{ij1}} r_{i1} + \underbrace{(2h_{i2} + 10)g(2h_{i2} + 7.5 + \lambda_j)}_{s_{ij2}} r_{i2} + \epsilon_{ij}, \tag{5.41}$$

where $g(\cdot)$ is the pdf of the Cauchy distributions with centre 0 and scale of 1, $\epsilon_{ij} \sim N(0, \sigma^2)$.

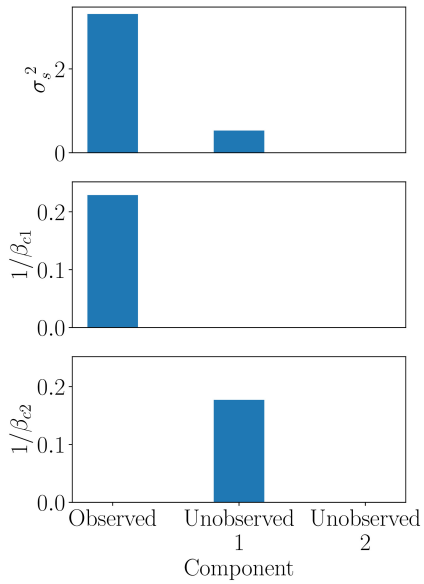
The first change in the setup to that described in Section 4.4 is that one of the component weights is not observed in the training data, so $\hat{r}_i = r_{i1}$ and $\bar{r}_i = r_{i2}$. This is equivalent to working with samples where only one chemical concentration is known but mixed other chemicals of unknown concentration. Note that because in this simple example we are not assuming that all of the chemical components are observed we cannot use the sum-to-one constraint, instead we use the unconstrained variational family for which the optimal distribution is given



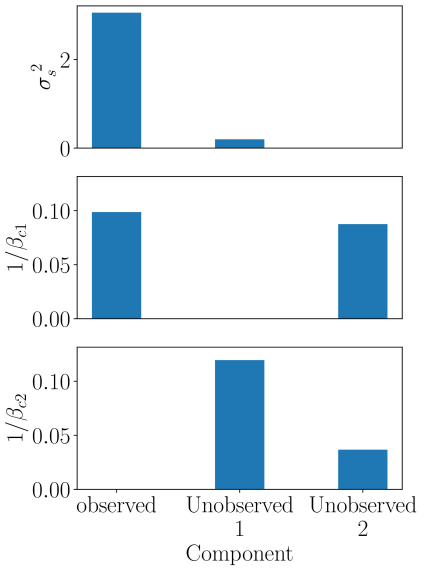
(a) Mean Field method, dependent latent variables.



(b) Structured method, dependent latent variables.

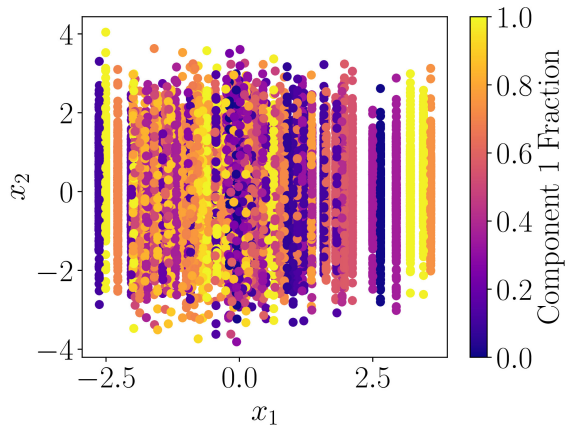


(c) Mean Field method, independent latent variables

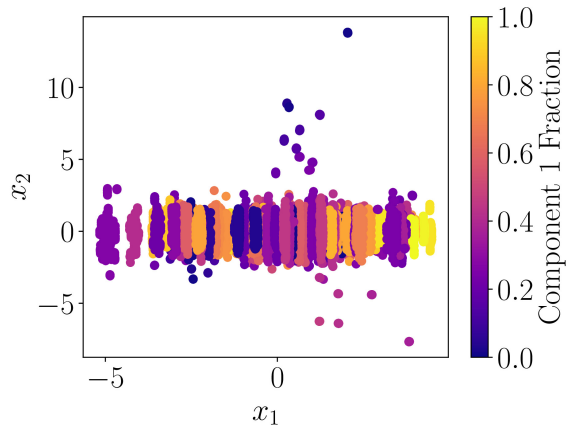


(d) Structured method, independent latent variables

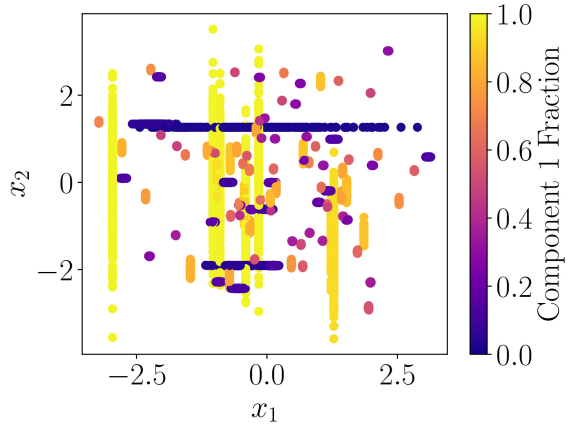
Figure 5.1: Gaussian process parameters for the toy example with unobserved components. The top row of each plot is the scale parameter of the GP kernel. The bottom two rows of each plot shows the reciprocal of the length-scale; a larger value indicates more variation in that dimension. Both the mean field and structured approaches are able to: (*i*) identify that only one unobserved component is required to fit the data, and (*ii*) identify when the two significant components share a single significant latent variable or two distinct latent variables. An interesting distinction in the optimisation between the mean field variational families (*left column*) and the structured families (*right column*) is that the the structured families learn shorter length scales for variation. This does not affect predictions, as for both the scale of that component is very close zero. However, it indicates that the structured family has a markedly different loss landscape than the mean field family which drives the variance of the learned function to zero much faster.



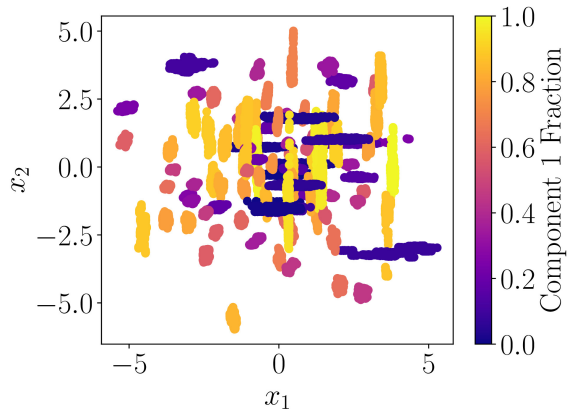
(a) Mean Field method, dependent latent variables.



(b) Structured method, dependent latent variables.

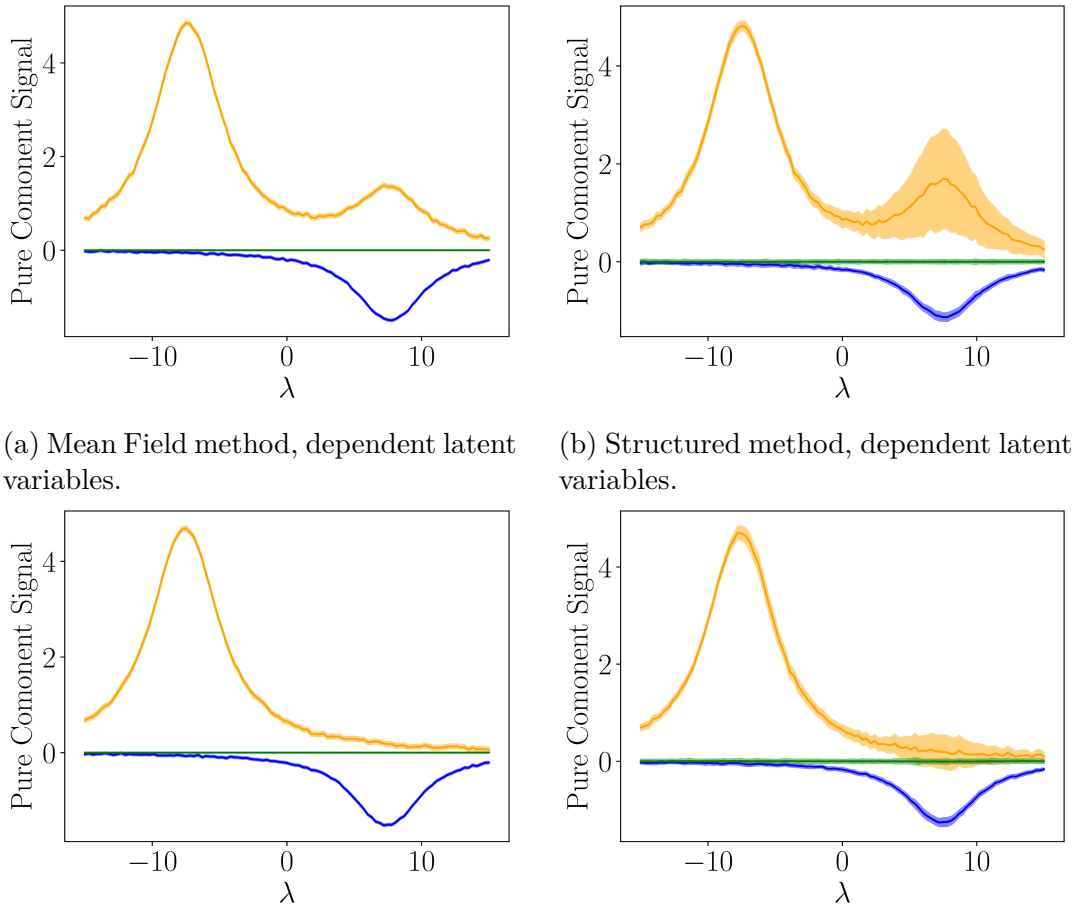


(c) Mean Field method, independent latent variables



(d) Structured method, independent latent variables

Figure 5.2: Samples from the variational posterior for the latent space of the test data $h_i^* \sim q(h_i^*)$. The latent spaces for the toy example introduced in this chapter for mean field (*left column*) vs. structured (*right column*) variational families, and variation caused by a single (*top row*) vs. independent (*bottom row*) latent variables. For models with a single latent variable, the distribution of the points resemble the prior along one dimension, indicating that the model has correctly identified that the model has correctly identified that only one latent variable is required for all data points. Meanwhile, for two independent latent variables the distribution of the point in in the latent space depends on the pure component weights for that sample. If a particular data point had very little of one fraction it is impossible to estimate the latent variable associated with that component can't be estimated accurately. This results in a distribution of points in the latent space which could have high uncertainty in either of the latent dimensions depending on the relative abundances of the components. This can be seen clearly in the bottom row of figures, where points with low component one (dark coloured points) have a high variance in x_1 and low variance in x_2, while points with high component one (light points) have low variance in x_i and high variance in x_2. The structured variational families appear to have a generally higher uncertainty than their mean field counterparts.



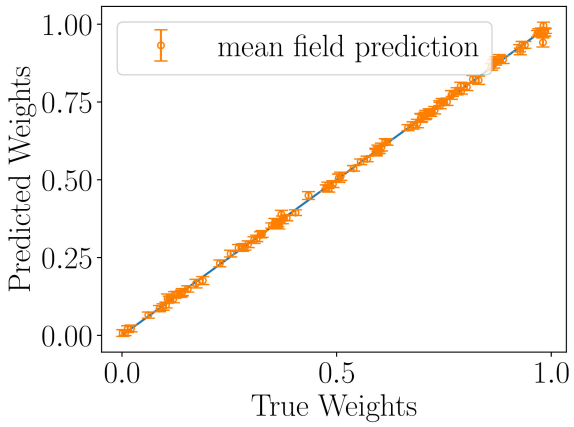
(a) Mean Field method, dependent latent variables.

(b) Structured method, dependent latent variables.

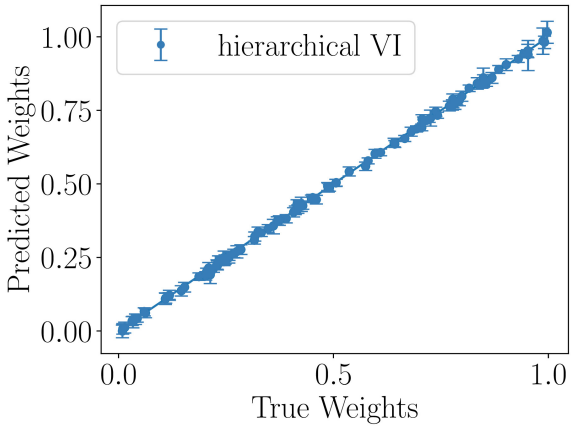
(c) Mean Field method, independent latent variables

(d) Structured method, independent latent variables

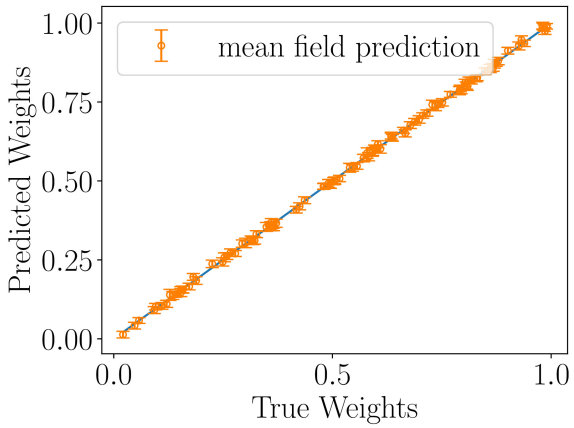
Figure 5.3: Pure component spectra mean and 95% credible interval estimated by the uncollapsed WS-GPLVMs. The estimates are taken at the origin of the latent space ($x = (0, 0)^T$) The left column shows the mean field estimates, while the right column shows the structured estimates. The top row shows the case where a single latent variable controlling both of the components, while the lower row shows the estimates for the two components independent. The signal associated with the observed component is shown as orange, while WS-GPLVM's two components associated with unobserved components are shown in blue and green. In the case where there is a single latent variable which controls both peaks the model incorrectly learns that pure component one has a second peak at the location of pure component two. This does not affect the predictions as the model correctly learns that there is also a peak for the unobserved component there, allowing the signals to cancel one another out, but it does indicate that the pure component spectra estimated by the model must be treated with scepticism. As in the latent space, the structured model does a better job of estimating the uncertainty and has appropriately learned to be sceptical about its own estimate for the pure component signal around the location of the peak with the unobserved signal.



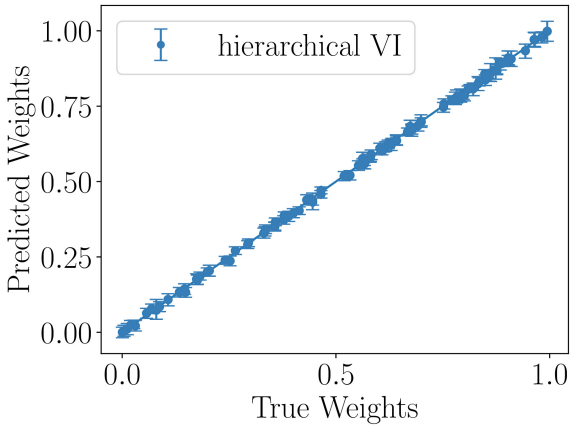
(a) Mean Field method, dependent latent variables.



(b) Structured method, dependent latent variables.



(c) Mean Field method, independent latent variables



(d) Structured method, independent latent variables

Figure 5.4: Plots of predictions for the pure component weights of the toy example introduced in this chapter for mean field *(left)* and structured *(right)*, and shared *(top)* and independent *(bottom)* latent variables. Both variational families are able to make good predictions for both conditions. The structured variational family has marginally larger error bars, however for this simple example it is not significant. This minor difference between the predictions of the structured and mean field families indicates that each spectra simply contains more information than is needed to produce a good estimate for the pure component weights, in a more complex example in which several wavelengths needed to be considered together in order to make good predictions we expect that there would be a more significant difference in the predictions between the two variational families.

in Equation (5.9). The training data for this example consists of 100 data points $\{\{d_{ij}|j = 1,\ldots,M\}, r_{i1}|i = 1,\ldots,N\}$ and the test data consists of $\{\{d^*_{ij}|j = 1,\ldots,M\}|i = 1,\ldots,N\}$. As in Chapter 4 the training data is generated from this model with 25 data points (r_{i1}, r_{i2}) associated to each of the following four mixture weights: $\{(0.2, 0.8), (0.4, 0.6), (0.6, 0.4), (0.8, 0.2)\}$ and 100 additional test data points with weights drawn from the Dirichlet distribution $(r^*_{i1}, r^*_{i2}) \sim Dir(1_2)$.

We use two choices of distribution for the two latent variables to illustrate the model learning whether a single latent variable effects both pure component spectra or just a single one, as discussed in Setion 5.3.1. We do this by simulating this example with two settings: one *dependent* scenario where a single latent variable drives the effects in both components, *i.e.* $h_{i1} = h_{i2}$ and $h_{i1} \sim N(0, 1)$, and an *independent* scenario where $(h_{i1}, h_{i2}) \sim N(0, \mathbb{I}_2)$ are distinct and uncorrelated.

The model which we fit includes one observed component and two unobserved components in the mixture. This allows us to illustrate the performance of the procedure when assuming too many unobserved components.

Both the mean-field model, whose variational family is given in Equation 5.2, and structured models, whose variational family is in Equation 5.30, where the weights are conditional on the inducing points, were fit to this data with two latent variables, two unobserved components and two latent variables and 8 inducing points. The structured variational inference samples $P = 32$ when drawing our points $s_{ijp} \sim q(s_{ij})$.

The results from these experiments, for both mean field and structured families (left and right columns) and both dependent and independent latent variables (top and bottom rows) are shown in Figures 5.1-5.4. We do not compare the approach from this chapter to that of the previous chapter as in this example we are interested in demonstrating that the model is capable of identifying dependent or independent latent variables and comparing the predictions of the mean-field and structured variational families.

Figure 5.1 shows the hyper parameters learnt for this toy example. The top graph in each of

these cases is the scale of the kernel (σ_s^2). The two plots below show the inverse of the length scale $1/\beta_{ia}$ for the two latent dimensions. The reciprocal is plotted for the length scale so the highest values correspond to dimensions which cause variation, while small values correspond to dimensions of the latent space along which variation does not affect that component.

For both the mean field and the structured model the third component has a scale of zero meaning that this component was switched off and differences in the length scale can be ignored. The scale of the second component being significantly lower than the first component also is not important in these results, as the second component has unobserved weights, hence the weight of the second component will be related to the prior placed on the unobserved weights - if the variance on $p(\bar{r}_i)$ was set to a lower value the scale of the GP would increase correspondingly and vice versa. In both variational families and both dependent and independent latent variable configurations, the model correctly identifies that there are only two components present in the mixture, resulting in the scale (σ_s^2) of the third component being very close to zero in all cases. Both variational families are able to correctly identify when each of the two components is dependent or independent of each other. This is shown by the single significant latent dimension being the same for both components in the dependent case, while they are different in the independent case.

Figure 5.2 shows the distribution of the test data's latent variables. It does this by showing samples from the variational posteriors for the test set, $h_i^* \sim q(h_i^*)$. For the mean field variational family uncertainty in each of the N^* test points is represented by 256 samples from $N(\mu_i^{h^*}, \Sigma_i^{h^*})$. For the structured VI case these are represented by 16 samples from each of the 32 mixture components $N(\mu_{ip}^{h^*}, \Sigma_{ip}^{h^*})$. The plots are coloured by the fraction of the pure component weights made up of the component with observed weights (r_{i1}).

The dependent latent variables both variational families exhibit the expected behaviour that there is a large uncertainty along the dimension which are not significant for causing variation in the pure component signal, characterised by a large variation in the vertical direction and a small variance in the horizontal direction.

For the independent latent variable there is significant also variation in direction in the latent

space which is not significant for that data point. An additional complexity is because the direction which is not significant depends on whether there is a large or a small fraction of the observed pure component weight present in that sample, illustrated in the figure by the colour of the data point. When there is a large fraction of the component with observed weights, the latent variable associated with variation in that pure component is very well specified. However if there was no presence of the component for which the weights are observed the distribution of points in the direction of variation of that component should be identical to the prior. Note that the weights for this test data points have been drawn from a Dirichlet distribution, so when component one is high component two is low and visa versa, and that the same behaviour of uncertainty in the irrelevant direction applies for component two, the plots behave as expected. Points with a large presence of the observed component and a low presence of the unobserved component (yellow points) have a low uncertainty in the direction associated with variation in the observed pure component signal and high uncertainty in the direction associated with variation in the other direction. This is illustrated by the yellow samples having a wide spread in the vertical direction. Conversely, points with low presence of component one (dark points) have high uncertainty in first latent dimension, but low uncertainty in the second latent dimension resulting in a large horizontal spread in the figure. While not huge, the structured variational inference approach results in a larger uncertainty in the latent space than the mean field approach, although this difference is expressed more clearly in other Figures.

Figure 5.3 shows the estimated pure component spectra at the origin of the latent space ($x = (0, 0)^T$). Both variational families correctly place a peak around -7.5 for component one. In the dependent latent variable case the model incorrectly places a peak around $\lambda = 7.5$, which corresponds to the location of the second peak. However, the incorrect inclusion of this second peak does not affect the predictions from the model, as illustrated in Figure 5.4d. This peak aligns with the second component, for which the weights are unobserved in this example, has a peak in the same location around $\lambda = 7.5$, allowing variation in the observed signal to be explained without requiring variation in the first component. The second component's signal $\bar{s}_{ij}$ can be either positive or negative, this simply requires the corresponding weights to be

multiplied by a negative value for $\bar{r}_i$ in order to recreate the spectra. Additionally, all of these models correctly find that the second unobserved pure component spectra, shown in green, is zero for all values.

Figure 5.4 also illustrates the increased uncertainty given by the structured family. The effect of the increased uncertainty is only minor in Figure 5.4d, which shows only a very minor increase in uncertainty. This may be due to the example being very simple, allowing the weights to be ascertained with a very low uncertainty provided that only some of the wavelengths are known with high certainty, but the more accurate representation of the uncertainty in the pure component signals could be of interest for a setting where the main goal of a signal separation is accurately unmixing the pure component signals.

5.5.2 Real Spectroscopy Datasets

Data

We examine two real world spectroscopy datasets here. First, we examine the spectroscopy dataset which was used in Section 4.4 and was introduced by Wülfert et al. [1998]. We referred to this as the Spectroscopy regression dataset in the previous chapter, in this chapter we have multiple datasets which we use for similar regression tasks so we refer to is as the Wulfert dataset instead. This dataset consists of a mixture of three components, water, ethanol and propan-2-ol, which are measured at differing temperatures to cause variation in the pure component spectra. This dataset makes use of the sum-to-one constraint introduced in this chapter, but does not require the inclusion of additional unobserved components. Instead we use this dataset to illustrate the significant computational advantages presented by our new method in terms of speed, accuracy and ease of use. The SNV transformation is used on this dataset prior to fitting any models, as described in Section 4.3. This example does not require any unobserved components, but is directly comparable to the method presented in the previous chapter and was therefore included as to show a comparison.

The second dataset which we use is one from pharmacutical tablet manufacturing introduced by

Dyrby et al. [2002]. This mixture contains two components: Active Pharmacutical Ingredient (API) and excipient. The weights in this experiment are also fractional and so require the sum to one constraint. This dataset was preprocessed by removing wavelengths which contain large amounts of noise. After this SNV preprocessing was applied. This example has solid rather than liquid samples being measured, which results in more complex interaction between the light and sample in spectroscopy. This more complex interaction requires introducing additional unobserved components to account for these interactions.

Experiments

For these two datasets we compare models with two dataset sizes. The first experiment uses a random split of half the data being used for training and the other half used for testing. The second experiment used a 10-fold split of the data, so 90% of the data had known weights. The purpose of this test was to investigate the two somewhat contradictory goals of using a physically motivated GP prior. One of the advantages of GPs is that they are highly flexible models, so given enough data they should be able to model most possible variations in the pure component spectra. On the other hand, GPs with RBF/ARD kernels encourage smoothness which we believe is a good prior for the variation of pure component spectra. The use of a larger number of training data points allows us to see the extent to which the flexibility of the GP model is important, while the use of a smaller number of training samples reveals the extent to which it is the smooth prior placed on the pure component variation which is of value.

These datasets were fit with both the Mean field and Structured variational families. We perform 10 random restarts of the optimisation procedure for each of these datasets. For this chapter the only values affected by the random restart is the noise added to the mean field estimate for μ_j^u, the positions of the inducing points V and the draws for the SVI samples of U. The random restart with the highest ELBO was reported for the results, although in the discussion section we suggest that further research may be needed into how to select between fits.

We report the values of the MSEP between the mean of the predicted distribution and the true

value of the test data points. We also report the fraction of test data points which lie within the 95% credible intervals for each point, referring to this as the coverage in the results. Along with this, we report the mean length of the credible intervals. A good result from these metrics would be a short interval and a high coverage.

Comparison Methods

We compare the variable signal spectroscopy approach to the static signal approach, where the pure component signals are assumed to be static across all wavelengths. The only hyper parameter to be selected for this model is the number of unobserved components which is set to one for the Dalby dataset and zero for the solution dataset. We report all of the same metrics as the variable spectra models.

We also compare to a PLS model for both datasets. We select the number of latent variables for the PLS model with a 10 fold cross validation on the training data and picking the lowest RMSE value. For the PLS model we only report the MSEP.

Results

	Dalby 0.1 Test Fraction		
Method	**MSE**	**Coverage**	**CI Length**
Static	$2.63(0.80)e-5$	$1.29(0.45)e-1$	$2.10(0.04)e-3$
MF-WS-GPLVM	$1.81(0.32)e-5$	$1.19(0.51)e-1$	$1.59(0.39)e-3$
Structured-WS-GPLVM	$1.95(0.75)e-5$	$7.39(1.04)e-1$	$7.36(2.67)e-3$
PLS	$9.51(1.82)e-6$	-	-

Table 5.1: Results for the Dalby dataset with 90/10 training/test split.

	Dalby 0.5 Test Fraction		
Method	**MSE**	**Coverage**	**CI Length**
Static	$3.01(0.93)e-5$	$9.61(2.55)e-2$	$1.76(0.33)e-3$
MF-WS-GPLVM	$1.54(0.27)e-1$	$3.42(1.92)e-2$	$2.73(3.97)e-2$
Structured-WS-GPLVM	$1.52(0.19)e-1$	$7.87(4.04)e-2$	$3.48(1.23)e-1$
PLS	$1.01(0.12)e-5$	-	-

Table 5.2: Results for the Dalby dataset with 50/50 training/test split.

	Wulfert 0.1 Test Fraction		
Method	**MSE**	**Coverage**	**CI Length**
Static	$-1.44(0.36)e-2$	$3.48(0.86)e-1$	$1.18(0.01)e-1$
MF-WS-GPLVM	$7.02(5.09)e-5$	$2.61(1.05)e-1$	$4.08(0.19)e-3$
Structured-WS-GPLVM	$5.48(2.45)e-5$	$3.73(1.32)e-1$	$4.65(2.40)e-3$
PLS	$3.44(1.32)e-4$	-	-

Table 5.3: Results for the Wulfert dataset with 90/10 training/test split.

	Wulfert 0.5 Test Fraction		
Method	**MSE**	**Coverage**	**CI Length**
Static	$1.99(0.31)e-2$	$2.35(0.22)e-1$	$1.00(0.05)e-1$
MF-WS-GPLVM	$7.97(2.07)e-5$	$2.56(0.41)e-1$	$4.26(0.19)e-3$
Structured-WS-GPLVM	$6.65(2.56)e-5$	$3.88(0.83)e-1$	$5.23(2.57)e-3$
PLS	$3.55(0.07)e-4$	-	-

Table 5.4: Results for the Wulfert dataset with 50/50 training/test split.

Method	Time (s)
MF-WS-GPLVM (max ELBO)	14.0
Structured-WS-GPLVM	106.0
WS-GPLVM-ind.	96.4

Table 5.5: Time taken to optimise model. Note that the WS-GPLVM-ind. results are used as this corresponds to the model used in these chapters. The performance of the WS-GPLVM-ind. on the Wulfert dataset given in Table D.1, and is significantly below the results presented here.

The results for the solution dataset are show in Tables 5.3 and 5.4. The mean squared error for both the mean field and structured methods presented in this chapter are significantly lower than for PLS or the static signal model. The 0.5 test fraction results are also comparable to the spectroscopy regression results from the previous chapter in Table D.1 and significantly improves those results. The coverage for all of the methods are significantly lower than the desired 95%. The coverage of the structured variational family is higher for only a small increase in the credibility interval length, suggesting that the structured family we have selected improves the uncertainty estimates from our model, but further improvements would be required to make these uncertainty estimates reliable.

The Dalby data set results are shown in Table 5.1 and 5.2. With a low fraction of the data being used for training data the results for the the static signal were significantly better than those of the variable signal approach. However, for the 90/10 training test split the variable signal shows an improved MSE compared to the static signals. The weakness of the variable signal

models for datasets with fewer training data points can be explained by noting that because of the increased flexibility of the model more data is required to correctly shrink the posterior to be sufficiently close to the posterior that the model predictions are accurate. For both training data sizes the PLS model perform better, suggesting that for this dataset the inductive biases from the use of PLS as a linear inverse model are more appropriate than the inductive biases from the physics based models. In Section 5.6 we discuss a possible cause for this and what further research could be carried out to address this.

5.5.3 Limits of the use of ELBO as an objective

There is no obvious mathematical reason for the PLS model to be more accurate than the Dalby dataset. One strong possibility though could be that the ELBO is not a good choice of objective for a model which is designed to make predictions about one latent random variable, in our case (r_i^*), based on an observed random variable, in our case d_i^*. The quantity that we would ideally like to estimate is the conditional density $p(r_i^*|d_i^*)$, however the ELBO which we calculate minimizes the KL divergence to the joint distribution including the pure component signals, inducing points and latent variables. As the Variational Inference objective does not prioritize the predictive distribution it is possible to achieve high a high ELBO with poor predictive performance.

The problem can be illustrated by considering the term

$$\mathcal{F}_i^* = \langle \log\left(p(d_i^*|r_i^*, S_i^*)\right)\rangle_{q(r_i^*, S_i^*)} - KL(q(r_i^*)||p(r_i^*)) \tag{5.42}$$

which is the contribution of a test data point to the ELBO. The first term in this expression penalises the model for reconstruction error of the observed spectra, while the second term penalises the distance of the variational posterior of the weights to the prior. The issue is that the reconstruction of the spectra is not tied to the prediction of the weights at all. Instead this term would be optimised if the distribution for the signals $q(S_i^*)$ was such that the observation d_i^* could be reconstructed perfectly with only very minimal use of r_i^*, thereby allowing the penalty

term $KL(q(r_i^*)||p(r_i^*))$ to be minimized. In fact, part of the motivation for reformulating the ELBO such that the optimal distribution for $q(r_i^*)$ was to avoid this problem of the pure component spectra describing too much of the variation.

Ideally this behaviour would be avoided by the counteracting KL divergence between the variational posterior for the pure component signals and the prior, $KL(q(S, S^*)||p(S, S^*))$. Unfortunately, this is not guaranteed to happen for all combination of priors on the pure component signals and weights, and we demonstrate that our model is susceptible to this.

To demonstrate this we ran another experiment on the Dalby dataset introduced earlier in the chapter. We use the mean field variational family, one unobserved component, and two latent dimensions using the 90/10 training test splits. We fit two models with this setup, one with 8 inducing points in the latent space and one with 16. By increasing the number of inducing points we allow the non-linear variations of the pure component spectra to become more expressive.

As with the results reported earlier in this chapter we refit each of the 10 datasets with 10 random restarts. Figure 5.5 shows the ELBO plotted against the MSE. The blue crosses show all of the runs. Additionally, for each dataset the runs with the best MSEP are shown with green arrows, the best NLL with green arrows, and the best ELBO - the run which would be selected by our criteria - are shown with red dots.

The model with 8 inducing points has limited capacity to describe variations in the pure component spectra, requiring it to make use of the weights for a good reconstruction for the observed spectra. However, the model with 16 inducing points has a much higher capacity to represent a complex distribution for the pure component spectra, making it susceptible to the problem described above. The results for these models match this. For 8 inducing points, achieving a high ELBO implies good estimates for $q(r_i^*)$ and hence a low MSEP. However, with 16 inducing points a new possible optima for the variational posterior appears with much higher MSEP. In fact every dataset results in models with low MSEP, but the criterion of the ELBO only selects models which fall into the optima with a high MSEP. These results are exactly consistent with the issue described above, in which a trade between a more expressive set of

pure components in exchange for less dependence on the distribution of the weights is benefical for the ELBO, but bad for the fit.

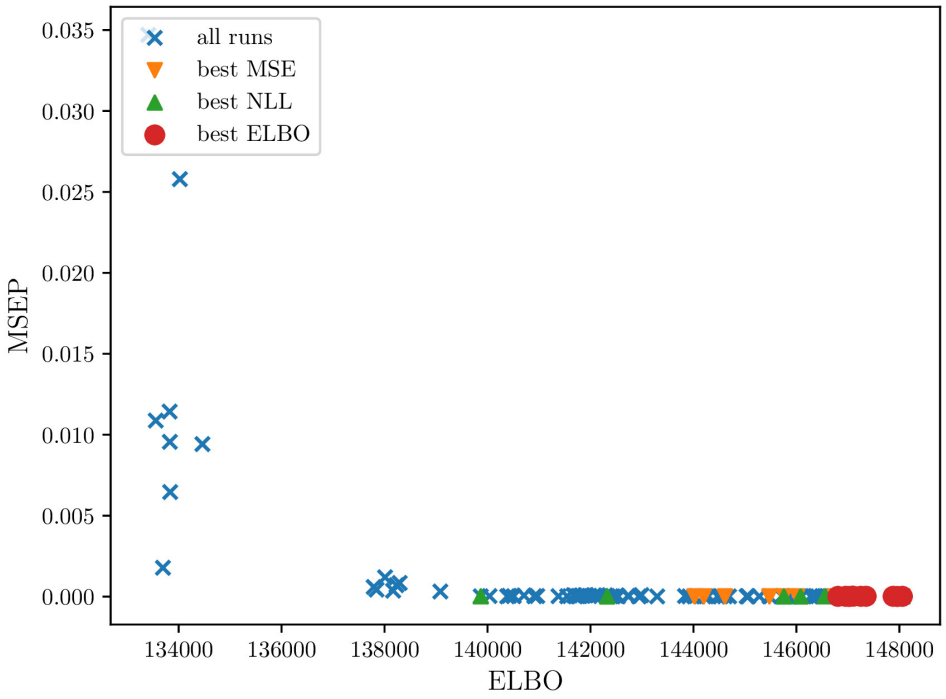
This issue of overly powerful decoders removing the dependence on latent variables is well known in the Variational Auto Encoder (VAE) literature Kingma and Welling [2013], Bowman et al. [2015], and several alternatives to the objective and model have been examined Chen et al. [2016], Higgins et al. [2017], Zhao et al. [2017]. There are several methods of decomposing the ELBO meaningfully Hoffman and Johnson [2016], and further research should consider if it is possible to adapt the objective in a way which enforces appropriate use of the weights for predictions and results in better estimates for the weights.

5.6 Discussion

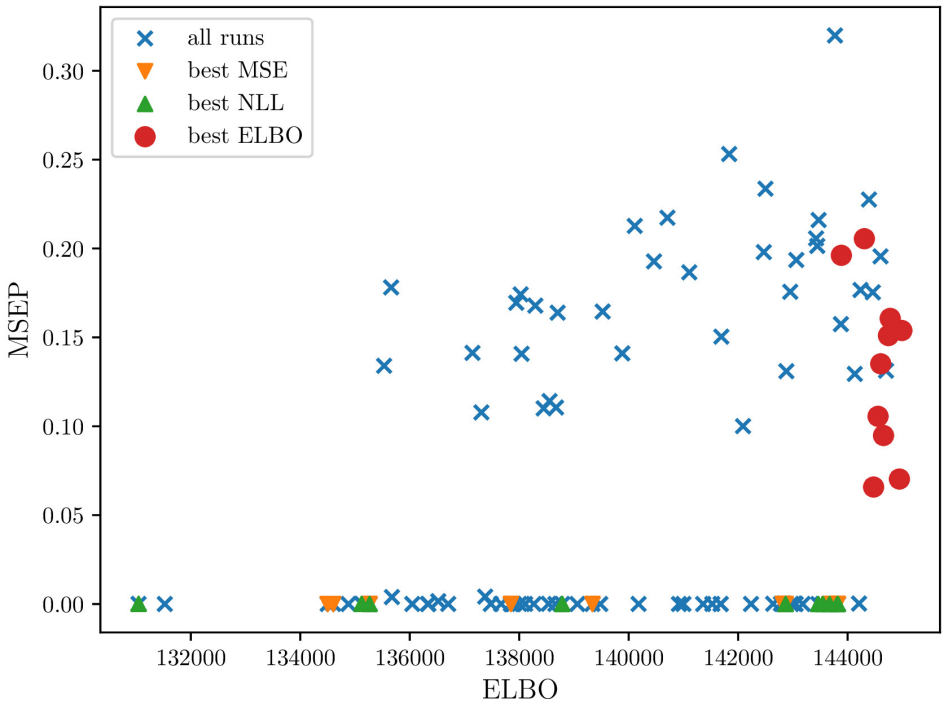
The improvements suggested in this chapter significantly improved the mean field approach of using GPLVMs in the previous chapter. The toy example illustrated that in principle both variational families are able to identify the number of unobserved mixture components present in samples, and whether variations in these components are dependent or independent. This removes some of the obvious issues limiting the approach taken in the previous chapter.

Additionally, the use of the analytic optimum for the weights significantly simplified, sped up and improved the accuracy of the approach compared to the method in the previous chapter. It also allowed the inclusion of new datasets which required the addition of unobserved components to make accurate predictions. Additionally, pushing the analytic optimum from the distribution of the inducing points to the distribution of the weights significantly improved the stability of the estimates, resulting in a model which is attractive for practical use in circumstances where the pure component spectra are known to vary significantly.

The inclusion of the Structured Variational family showed some amount of improvement in coverage compared to the mean field approach. However, the coverage was still below, and sometimes far below, the desired 95% credible interval. One avenue that could be used as further research would be to continue to examine more flexible variational posteriors in order



(a) 8 Inducing points,



(b) 16 inducing points

Figure 5.5: Plots of ELBOs against MSEP for the Dalby dataset with differing numbers of inducing points (8,16) for a 90/10 split on the training/test data fractions. For the lower number of inducing points the increasing the ELBO predicts a decrease in the MSEP (Figure 5.5a), meaning for this setting selecting the model with the highest ELBO is a good criterion for model selection. However, in the model with a larger number of inducing points there appear to be multiple local optima and selecting the maximum ELBO always results in selecting a model with a high MSEP (Figure 5.5b).

to improve approximations to the true posterior. However, for the approach suggested here at least, there is a significant computational burden for the structured approach. As the mean field approach proposed in this chapter already achieved a high level of stability and a very similar level while not achieving the high coverage which would imply accurate uncertainty estimates further work to estimate the uncertainties from this model is required.

There are several other interesting avenues of research which remain in this line of work. One of these is simply to apply the same technique to similar problems in different fields, for example hyper-spectral imaging [Bhargava et al., 2024]. Although not presented here the approach of Lázaro-Gredilla et al. [2012] would make it possible to work on the kind of classification problems common in that field, and the use of the uncollapsed bound for the GP would allow the approach to scale up to much larger datasets with SVI [Hoffman et al., 2013, Hensman et al., 2013, 2015]. The inclusion of SVI does add complexity to the model as the optimisation of the latent variables needs to be carried out at each step, but the techniques from Lalchand et al. [2022] would make this problem tractable.

Additional research should also focus on the issue described in this chapter of the ELBO not being a good objective to enforce accurate use of the linear weights. Exploration of this area may explain the remaining performance gap for some datasets to inverse models such as PLS.

Chapter 6

Conclusion

6.1 Summary of Thesis Achievements

This thesis has explored the challenges of analysing spectroscopic data in pharmaceutical manufacturing settings, with a particular focus on two key areas: (i) providing uncertainty quantification for partial least squares regression and (ii) developing new probabilistic models for variable spectroscopic signals.

In Chapter 3, we introduced a novel bootstrap-based approach for uncertainty quantification in partial least squares regression. This method provides a principled way to estimate prediction uncertainties without making strong assumptions about the parametric form of the uncertainty. We demonstrated the effectiveness of this approach across multiple industrial case studies, including Design Space identification for pharmaceutical manufacturing processes and spectroscopic analysis. The method shows particular strength in cases where existing linear approximations break down, while maintaining comparable performance to established methods in standard settings.

In Chapter 4, we developed the Weighted-Sum Gaussian Process Latent Variable Model (WS-GPLVM), a novel probabilistic approach for analysing spectroscopic data with varying signals. This model combines the physical understanding of Beer-Lambert's law with the flexibility of

Gaussian Process Latent Variable Models, allowing us to capture variations in pure component spectra while maintaining interpretability. We demonstrated the model's effectiveness on both simulated data and real-world spectroscopic datasets, showing improved performance over existing approaches in many settings.

Finally, in Chapter 5, we introduced several key improvements to the WS-GPLVM framework, addressing practical limitations and extending its applicability. These improvements included a simplified training procedure through collapsed variational bounds, the ability to handle unobserved components in the training data, and improved uncertainty quantification through structured variational inference. These advances make the model more practical for real-world applications while maintaining its strong theoretical foundations.

6.2 Applications

The methods developed in this thesis have immediate applications in pharmaceutical manufacturing, but their potential uses extend beyond this domain:

- **Pharmaceutical Quality Control:** The bootstrap PLS method provides a robust way to estimate uncertainties in spectroscopic measurements used for real-time release testing. This is particularly valuable when dealing with low-dose products or when monitoring decay curves in continuous manufacturing.
- **Process Monitoring:** The WS-GPLVM framework enables more accurate analysis of spectroscopic data when pure component signals vary due to environmental conditions. This is crucial for maintaining product quality in manufacturing settings where temperature, humidity, or other factors may affect measurements.
- **Mixture Analysis:** Beyond pharmaceuticals, the methods can be applied to any setting where complex mixtures need to be analyzed and components may vary between measurements. This includes applications in environmental monitoring, food safety, and materials characterization.

- **General Signal Separation:** The theoretical framework developed for WS-GPLVM could be adapted for other signal separation problems where the underlying signals vary according to some unobserved factors, such as in audio processing or hyperspectral imaging.

6.3 Future Work

Several promising directions for future research emerge from this work:

First, while we have demonstrated the effectiveness of our methods on real datasets, deploying them in active pharmaceutical manufacturing settings would provide valuable insights into their practical limitations and areas for improvement. This would help bridge the gap between theoretical development and practical implementation.

Another important direction is the integration of physical constraints and domain knowledge into our models. While the current WS-GPLVM framework incorporates some physical understanding through its structure, there may be additional constraints or relationships that could be incorporated to improve performance. This could include known relationships between temperature and spectral shifts, or chemical constraints on possible variations in pure component spectra.

The uncertainty quantification methods developed here could also be extended to handle more complex types of uncertainty. For example, developing methods to separate uncertainty due to model misspecification from uncertainty due to noise in the measurements would provide valuable insights for practitioners.

Finally, the successes of deep learning in many domains suggest investigating hybrid approaches that combine the interpretability and physical basis of our current models with the flexibility of neural networks. This could potentially lead to models that better capture complex non-linear relationships while maintaining the physical interpretability that makes our current approaches valuable in practice.

These future directions reflect the ongoing need to balance theoretical rigor with practical applicability in the development of analytical methods for pharmaceutical manufacturing and beyond.

Bibliography

F. Alsmeyer, H.-J. Koß, and W. Marquardt. Indirect spectral hard modeling for the analysis of reactive and interacting mixtures. *Applied Spectroscopy*, 58(8):975–985, 2004.

Y. Altmann, N. Dobigeon, S. McLaughlin, and J.-Y. Tourneret. Nonlinear spectral unmixing of hyperspectral images using Gaussian processes. *IEEE Transactions on Signal Processing*, 61(10):2442–2453, 2013. .

M. Alvarez and N. Lawrence. Sparse convolved Gaussian processes for multi-output regression. *Advances in Neural Information Processing Systems*, 21, 2008.

M. A. Alvarez and N. D. Lawrence. Computationally efficient convolved multiple output Gaussian processes. *Journal of Machine Learning Research*, 12:1459–1500, 2011.

M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012. .

A. M. Baldridge, S. J. Hook, C. Grove, and G. Rivera. The ASTER spectral library version 2.0. *Remote Sensing of Environment*, 113(4):711–715, 2009.

G. Bano, P. Facco, N. Meneghetti, F. Bezzo, and M. Barolo. Uncertainty back-propagation in PLS model inversion for design space determination in pharmaceutical product development. *Computers & Chemical Engineering*, 101:110–124, 2017.

G. Bano, P. Facco, F. Bezzo, and M. Barolo. Probabilistic design space determination in pharmaceutical product development: A Bayesian/latent variable approach. *AIChE Journal*, 64(7):2438–2449, 2018.

R. Barnes, M. S. Dhanoa, and S. J. Lister. Standard normal variate transformation and detrending of near-infrared diffuse reflectance spectra. *Applied Spectroscopy*, 43(5):772–777, 1989.

A. Bhargava, A. Sachdeva, K. Sharma, M. H. Alsharif, P. Uthansakul, and M. Uthansakul. Hyperspectral imaging and its applications: A review. *Heliyon*, 10(12), 2024.

M. Binois and N. Wycoff. A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization. *ACM Transactions on Evolutionary Learning and Optimization*, 2(2):1–26, 2022.

C. Bishop. Bayesian PCA. *Advances in Neural Information Processing Systems*, 11, 1998.

C. M. Bishop and G. D. James. Analysis of multiphase flows using dual-energy gamma densitometry and neural networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 327(2-3):580–593, 1993.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

R. A. Borsoi, T. Imbiriba, and J. C. M. Bermudez. Deep generative endmember modeling: An application to unsupervised spectral unmixing. *IEEE Transactions on Computational Imaging*, 6:374–384, 2019.

R. A. Borsoi, T. Imbiriba, J. C. M. Bermudez, C. Richard, J. Chanussot, L. Drumetz, J.-Y. Tourneret, A. Zare, and C. Jutten. Spectral variability in hyperspectral data unmixing: A comprehensive review. *IEEE Geoscience and Remote Sensing Magazine*, 9(4):223–270, 2021.

S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

R. Bro. Multiway calibration. Multilinear PLS. *Journal of Chemometrics*, 10(1):47–61, 1996.

A. J. Burnham, R. Viveros, and J. F. MacGregor. Frameworks for latent variable multivariate regression. *Journal of Chemometrics*, 10(1):31–45, 1996.

M. Chatterjee, J. Le Roux, N. Ahuja, and A. Cherian. Visual scene graphs for audio source separation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1204–1213, 2021.

H. Chen, B. R. Bakshi, and P. K. Goel. Bayesian latent variable regression via Gibbs sampling: Methodology and practical aspects. *Journal of Chemometrics*, 21(12):578–591, 2007a.

T. Chen, J. Morris, and E. Martin. Gaussian process regression for multivariate spectroscopic calibration. *Chemometrics and Intelligent Laboratory Systems*, 87(1):59–71, 2007b.

X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

Z. Dai, M. Álvarez, and N. Lawrence. Efficient modeling of latent information in supervised learning using Gaussian processes. *Advances in Neural Information Processing Systems*, 30, 2017.

A. Damianou, C. Ek, M. Titsias, and N. Lawrence. Manifold relevance determination. *arXiv preprint arXiv:1206.4610*, 2012.

A. C. Damianou, M. K. Titsias, and N. D. Lawrence. Variational inference for latent variables and uncertain inputs in Gaussian processes. *Journal of Machine Learning Research*, 17(42): 1–62, 2016.

H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

S. De Jong. SIMPLS: An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18(3):251–263, 1993.

D. de Souza, D. Mesquita, J. P. Gomes, and C. L. Mattos. Learning GPLVM with arbitrary kernels using the unscented transformation. In *International Conference on Artificial Intelligence and Statistics*, pages 451–459. PMLR, 2021.

E. del Castillo and M. S. Reis. Bayesian predictive optimization of multiple and profile response systems in the process industry: A review and extensions. *Chemometrics and Intelligent Laboratory Systems*, 206:104121, 2020.

M. C. Denham. Prediction intervals in partial least squares. *Journal of Chemometrics*, 11(1): 39–52, 1997.

M. Dirks and D. Poole. Incorporating domain knowledge about XRF spectra into neural networks. In *Workshop on Perception as Generative Reasoning, NeurIPS*, 2019.

M. Dyrby, S. B. Engelsen, L. Nørgaard, M. Bruhn, and L. Lundsberg-Nielsen. Chemometric quantitation of the active substance (containing C≡N) in a pharmaceutical tablet using near-infrared (NIR) transmittance and NIR FT-Raman spectra. *Applied Spectroscopy*, 56(5): 579–585, 2002.

B. Efron and G. Gong. A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistician*, 37(1):36–48, 1983.

B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. Chapman and Hall, New York, 1993.

S. el Bouhaddani, H.-W. Uh, C. Hayward, G. Jongbloed, and J. Houwing-Duistermaat. Probabilistic partial least squares model: Identifiability, estimation and application. *Journal of Multivariate Analysis*, 167:331–346, 2018.

L. Elden. Partial least-squares vs. Lanczos bidiagonalization—I: Analysis of a projection method for multiple regression. *Computational Statistics & Data Analysis*, 46(1):11–31, 2004.

K. Faber and B. R. Kowalski. Prediction error in least squares regression: Further critique on the deviation used in The Unscrambler. *Chemometrics and Intelligent Laboratory Systems*, 34(2):283–292, 1996.

K. Faber and B. R. Kowalski. Propagation of measurement errors for the validation of predictions obtained by principal component regression and partial least squares. *Journal of Chemometrics*, 11(3):181–238, 1997.

N. K. M. Faber. Uncertainty estimation for multivariate regression coefficients. *Chemometrics and Intelligent Laboratory Systems*, 64(2):169–179, 2002.

P. Facco, F. Dal Pastro, N. Meneghetti, F. Bezzo, and M. Barolo. Bracketing the design space within the knowledge space in pharmaceutical product development. *Industrial & Engineering Chemistry Research*, 54(18):5128–5138, 2015.

J. P. Folch, J. Odgers, S. Zhang, R. M. Lee, B. Shafei, D. Walz, C. Tsay, M. van der Wilk, and R. Misener. Practical path-based Bayesian optimization. *NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2023.

T. Fushiki. Bootstrap prediction and Bayesian prediction under misspecified models. *Bernoulli*, 11(4):747–758, 2005.

T. Fushiki, F. Komaki, and K. Aihara. Nonparametric bootstrap prediction. *Bernoulli*, 11(2): 293–307, 2005.

Y. Gandelsman, A. Shocher, and M. Irani. "Double-DIP": Unsupervised image decomposition via coupled deep-image-priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11026–11035, 2019.

S. García-Muñoz and E. Hernandez Torres. Supervised extended iterative optimization technology for the estimation of powder compositions in pharmaceutical applications: Method and lifecycle management. *Industrial & Engineering Chemistry Research*, 59(21):10072–10081, 2020.

S. García-Muñoz and E. Hernández Torres. Supervised extended iterative optimization technology for estimation of powder compositions in pharmaceutical applications: Method and lifecycle management. *Industrial & Engineering Chemistry Research*, 59(21):10072–10081, 2020.

J. Gerretzen, E. Szymańska, J. J. Jansen, J. Bart, H.-J. van Manen, E. R. van den Heuvel, and L. M. Buydens. Simple and effective way for data preprocessing selection based on design of experiments. *Analytical Chemistry*, 87(24):12096–12103, 2015.

M. G. Gustafsson. A probabilistic derivation of the partial least-squares algorithm. *Journal of Chemical Information and Computer Sciences*, 41(2):288–294, 2001.

I. S. Helland. On the structure of partial least squares regression. *Communications in Statistics—Simulation and Computation*, 17(2):581–607, 1988.

J. Hensman, M. Rattray, and N. Lawrence. Fast variational inference in the conjugate exponential family. *Advances in Neural Information Processing Systems*, 25, 2012.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

J. Hensman, A. Matthews, and Z. Ghahramani. Scalable variational Gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. PMLR, 2015.

E. M. Hetrick, Z. Shi, L. E. Barnes, A. W. Garrett, R. G. Rupard, T. T. Kramer, T. M. Cooper, D. P. Myers, and B. C. Castle. Development of near infrared spectroscopy-based process monitoring methodology for pharmaceutical continuous manufacturing using an offline calibration approach. *Analytical Chemistry*, 89(17):9175–9183, 2017.

I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017.

M. D. Hoffman and D. M. Blei. Structured stochastic variational inference. In *Artificial Intelligence and Statistics*, pages 361–369. PMLR, 2015.

M. D. Hoffman and M. J. Johnson. ELBO surgery: Yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, 2016.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

D. Hong, L. Gao, J. Yao, N. Yokoya, J. Chanussot, U. Heiden, and B. Zhang. Endmember-guided unmixing network (EGU-Net): A general deep learning framework for self-supervised

hyperspectral unmixing. *IEEE Transactions on Neural Networks and Learning Systems*, 33 (11):6518–6531, 2021.

A. Höskuldsson. PLS regression methods. *Journal of Chemometrics*, 2(3):211–228, 1988.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

ICH Expert Working Group. ICH harmonized tripartite guideline. Pharmaceutical development Q8 (R2). In *International Conference on Harmonization of Technical Requirements for Registration of Pharmaceuticals for Human Use*, 2009.

C. M. Jaeckle and J. F. MacGregor. Industrial applications of product design through the inversion of latent variable models. *Chemometrics and Intelligent Laboratory Systems*, 50(2): 199–210, 2000.

V. Jayaram and J. Thickstun. Source separation with deep generative priors. In *International Conference on Machine Learning*, pages 4724–4735. PMLR, 2020.

C. D. Kappatou, J. Odgers, S. García-Muñoz, and R. Misener. An optimization approach coupling preprocessing with model regression for enhanced chemometrics. *Industrial & Engineering Chemistry Research*, 62(15):6196–6213, 2023.

N. J. King and N. D. Lawrence. Fast variational inference for Gaussian process models through KL-correction. In *Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings 17*, pages 270–281. Springer, 2006.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

E. Kriesten, F. Alsmeyer, A. Bardow, and W. Marquardt. Fully automated indirect hard modeling of mixture spectra. *Chemometrics and Intelligent Laboratory Systems*, 91(2):181–193, 2008.

K. P. Kusumo, L. Gomoescu, R. Paulen, S. García Muñoz, C. C. Pantelides, N. Shah, and B. Chachuat. Bayesian approach to probabilistic design space characterization: A nested sampling strategy. *Industrial & Engineering Chemistry Research*, 59(6):2396–2408, 2019.

D. Laky, S. Xu, J. S. Rodriguez, S. Vaidyaraman, S. García Muñoz, and C. Laird. An optimization-based framework to define the probabilistic design space of pharmaceutical processes with model uncertainty. *Processes*, 7(2):96, 2019.

V. Lalchand, A. Ravuri, and N. D. Lawrence. Generalised GPLVM with stochastic variational inference. In *International Conference on Artificial Intelligence and Statistics*, pages 7841–7864. PMLR, 2022.

N. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

M. Lázaro-Gredilla, S. Van Vaerenbergh, and N. D. Lawrence. Overlapping mixtures of Gaussian processes for the data association problem. *Pattern Recognition*, 45(4):1386–1395, 2012. .

P. Lebrun, B. Boulanger, B. Debrus, P. Lambert, and P. Hubert. A Bayesian design space for analytical methods based on multivariate models and predictions. *Journal of Biopharmaceutical Statistics*, 23(6):1330–1351, 2013.

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 *2011 IEEE International Conference on Bioinformatics and Biomedicine*, pages 526–531. IEEE, 2011.

J. S. Liu. *Monte Carlo strategies in scientific computing*, volume 10. Springer, 2001.

J. Luttinen and A. Ilin. Variational Gaussian-process factor analysis for modeling spatio-temporal data. In *Advances in Neural Information Processing Systems*, volume 22, 2009.

L. Mackey. Deflation methods for sparse PCA. *Advances in Neural Information Processing Systems*, 21, 2008.

G. Mariani, I. Tallini, E. Postolache, M. Mancusi, L. Cosmo, and E. Rodolà. Multi-source diffusion models for simultaneous music generation and separation. *arXiv preprint arXiv:2302.02257*, 2023.

W. F. Massy. Principal components regression in exploratory statistical research. *Journal of the American Statistical Association*, 60(309):234–256, 1965.

Y. Y. Nadew, X. Fan, and C. J. Quinn. Conditionally-conjugate Gaussian process factor analysis for spike count data via data augmentation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 37188–37212. PMLR, 2024.

J. Neri, R. Badeau, and P. Depalle. Unsupervised blind source separation with variational auto-encoders. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 311–315. IEEE, 2021.

M. N. Nounou, B. R. Bakshi, P. K. Goel, and X. Shen. Process modeling by Bayesian latent variable regression. *AIChE Journal*, 48(8):1775–1793, 2002.

M. P. Ochoa, S. García-Muñoz, S. Stamatis, and I. E. Grossmann. Novel flexibility index formulations for the selection of the operating range within a design space. *Computers & Chemical Engineering*, 149:107284, 2021.

J. Odgers, C. Kappatou, R. Misener, S. García Muñoz, and S. Filippi. Probabilistic predictions for partial least squares using bootstrap. *AIChE Journal*, 69(7):e18071, 2023.

J. Odgers, C. Kappatou, R. Misener, and S. Filippi. Weighted-sum of Gaussian process latent variable models. *arXiv preprint arXiv:2024.xxxx*, 2024.

A. C. Olivieri, N. M. Faber, J. Ferré, R. Boqué, J. H. Kalivas, and H. Mark. Uncertainty estimation and figures of merit for multivariate calibration (IUPAC technical report). *Pure and Applied Chemistry*, 78(3):633–661, 2006.

D. Palací-López, P. Facco, M. Barolo, and A. Ferrer. New tools for the design and manufacturing

of new products based on latent variable model inversion. *Chemometrics and Intelligent Laboratory Systems*, 194:103848, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. J. Peterson. A Bayesian approach to the ICH Q8 definition of design space. *Journal of Biopharmaceutical Statistics*, 18(5):959–975, 2008.

A. Phatak, P. Reilly, and A. Penlidis. An approach to interval estimation in partial least squares regression. *Analytica Chimica Acta*, 277(2):495–501, 1993.

E. H. Puwakkatiya-Kankanamage, S. García-Muñoz, and L. T. Biegler. An optimization-based undeflated PLS (OUPLS) method to handle missing data in the training set. *Journal of Chemometrics*, 28(7):575–584, 2014.

T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. *ACM Transactions on Knowledge Discovery from Data*, 7(3):1–31, 2013.

S. Ramchandran, M. Koskinen, and H. Lähdesmäki. Latent Gaussian process with composite likelihoods and numerical quadrature. In *International Conference on Artificial Intelligence and Statistics*, pages 3718–3726. PMLR, 2021.

E. Räsänen and N. Sandler. Near infrared spectroscopy in the development of solid dosage forms. *Journal of Pharmacy and Pharmacology*, 59(2):147–159, 2007.

C. E. Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

C. E. Rasmussen and C. K. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

T. J. Rato and M. S. Reis. SS-DAC: A systematic framework for selecting the best modeling approach and pre-processing for spectroscopic data. *Computers & Chemical Engineering*, 128:437–449, 2019.

M. S. Reis and P. M. Saraiva. Prediction of profiles in the process industries. *Industrial & Engineering Chemistry Research*, 51(11):4254–4266, 2012.

M. S. Reis, A. C. Pereira, J. M. Leça, P. M. Rodrigues, and J. C. Marques. Multiresponse and multiobjective latent variable optimization of modern analytical instrumentation for the quantification of chemically related families of compounds: Case study—Solid-phase microextraction (SPME) applied to the quantification of analytes with impact on wine aroma. *Journal of Chemometrics*, 33(3):e3103, 2019.

R. Rosipal and L. J. Trejo. Kernel partial least squares regression in reproducing kernel Hilbert space. *Journal of Machine Learning Research*, 2:97–123, 2001.

Z. Shi, J. Hermiller, and S. García Muñoz. Estimation of mass-based composition in powder mixtures using extended iterative optimization technology (EIOT). *AIChE Journal*, 65(1): 87–98, 2019.

I. M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802, 1967.

R. Tauler. Multivariate curve resolution applied to second order data. *Chemometrics and Intelligent Laboratory Systems*, 30(1):133–146, 1995.

Y. Teh, D. Newman, and M. Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. *Advances in Neural Information Processing Systems*, 19, 2006.

Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric latent factor models. In *International Workshop on Artificial Intelligence and Statistics*, pages 333–340, 2005.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574. PMLR, 2009.

M. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 844–851. JMLR Workshop and Conference Proceedings, 2010.

J. Trygg and S. Wold. Orthogonal projections to latent structures (O-PLS). *Journal of Chemometrics*, 16(3):119–128, 2002.

R. Urtasun and T. Darrell. Discriminative Gaussian process latent variable model for classification. In *Proceedings of the 24th International Conference on Machine Learning*, pages 927–934, 2007.

C. Vemavarapu and S. I. Badawy. Role of drug substance material properties in the processibility and performance of wet granulated products. In *Handbook of Pharmaceutical Wet Granulation*, pages 387–419. Elsevier, 2019.

V. E. Vinzi, W. W. Chin, J. Henseler, and H. Wang. *Handbook of partial least squares*. Springer, 2010.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

H. Wold. Soft modelling by latent variables: The non-linear iterative partial least squares (NIPALS) approach. *Journal of Applied Probability*, 12(S1):117–142, 1975.

S. Wold, A. Ruhe, H. Wold, and W. Dunn, III. The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3):735–743, 1984.

S. Wold, M. Sjöström, and L. Eriksson. PLS-regression: A basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109–130, 2001.

F. Wülfert, W. T. Kok, and A. K. Smilde. Influence of temperature on vibrational spectra and consequences for the predictive ability of multivariate models. *Analytical Chemistry*, 70(9): 1761–1767, 1998.

M. Wytock and J. Z. Kolter. Contextually supervised source separation with application to energy disaggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

K. H. Yip, I. P. Waldmann, Q. Changeat, M. Morvan, A. F. Al-Refaie, B. Edwards, N. Nikolaou, A. Tsiaras, C. A. de Oliveira, P.-O. Lagage, G. Kargl, B. R. Brandl, A. Glasse, and J.-P. Beaulieu. ESA-Ariel data challenge NeurIPS 2022: Inferring physical properties of exoplanets from next-generation telescopes. *arXiv preprint arXiv:2206.14642*, 2022.

B. M. Yu, J. P. Cunningham, G. Santhanam, S. Ryu, K. V. Shenoy, and M. Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In *Advances in Neural Information Processing Systems*, volume 21, 2008.

L. Zhang and S. García-Muñoz. A comparison of different methods to estimate prediction uncertainty using partial least squares (PLS): A practitioner's perspective. *Chemometrics and Intelligent Laboratory Systems*, 97(2):152–158, 2009. .

S. Zhao, J. Song, and S. Ermon. InfoVAE: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.

J. Zheng, Z. Song, and Z. Ge. Probabilistic learning of partial least squares regression model: Theory and industrial applications. *Chemometrics and Intelligent Laboratory Systems*, 158: 80–90, 2016.

Appendix A

Table of notation for Chapter 3

Symbol	Meaning
x	Input variable
y	Response variable
$\{(x_i, y_i)\}_{i=1}^N$	Training data set
$\{(x_i^*, y_i^*)_{(b)}\}_{i=1}^N$	Bootstrapped training data set
P	Least squares transformation from t to x
Q	Least squares transformation from t to y
W	Concatenated, transposed values of w_l
$\bar{x}$	Sample mean of X
$\bar{y}$	Sample mean of Y
$P_{(b)}^*$	Bootstrapped calculation of P using data set b
$Q_{(b)}^*$	Bootstrapped calculation of Q using data set b
$W_{(b)}^*$	Bootstrapped calculation of W using data set b
$\bar{x}_{(b)}^*$	Bootstrapped sample mean of the input data in data set b
$\bar{y}_{(b)}^*$	Bootstrapped sample mean of the output data in data set b
t	Latent variable found from PLS
e	Variance in input variable not considered by PLS
f	Noise from the Prediction to the true value of PLS
w_l	Direction of maximum covariance between X_l and Y
p_l	Least squares prediction between the lth component of t and X_l
q_l	Least squares prediction between the lth component of t and Y
X	Matrix of input training data
Y	Matrix of output training data
X_l	Transformed input variables for the j_th component of PLS
$\hat{C}_{a,b}$	MLE of the covariance matrix of vectors a and b

Symbol	Meaning
$\hat{y}$	Prediction of response variable from PLS given an input x
n_x	Number of dimensions of the input variables
n_y	Number of dimensions of the output variables
n_L	Number of latent variables of PLS
N	Number of training data points
$s(x)$	Standard error of sample output variable
t_ν	Student t distribution with ν degrees of freedom
DS	Design Space
y_{des}	Desired range of response variables for a manufacturing process
α	Error acceptance rate for the Design Space
B	Number of bootstrap replications
$\mathcal{N}(y;\mu,\sigma)$	Probability density function for a normal distribution with mean μ and variance σ^2 evaluated at y
$\delta(\cdot)$	Dirac delta function

Appendix B

Table of Notation for Chapter 4

Notation	Definition
$\cdot^*$	Star indicates test samples of a variable (where components are unobserved) *Note: there is an exception for μ_i^r, Σ_i^r and α_i^r which refer to the test data, but the star is omitted for brevity*
N	Number of observations
M	Number of observations for each sample
C	The number of components that are contained in the samples
L	The number of inducing points for each component
A	The number of Latent dimensions in the MO-GPLVM model
d_{ij}	The observation for sample i at position j
r_{ic}	The component mixture c for sample i
s_{ijc}	The cth pure component spectra evaluated at wavelength j for sample i
x_i	The latent variable of the ith sample
λ_j	The jth location of measurement for all the samples
ϵ_{ij}	The Gaussian noise observed for sample i at position j
r_i	The $\mathbb{R}^C$ vector containing all of the mixture components for sample i
s_{ij}	The $\mathbb{R}^C$ vector containing all of the pure component values for sample i at wavelength j
D	The $\mathbb{R}^{N\times M}$ matrix the training spectra
D^*	The $\mathbb{R}^{N^*\times M}$ matrix containing the test spectra
$\mathcal{D}$	The set of both training and test observations combined
R	The $\mathbb{R}^{N\times C}$ matrix containing the training mixture components
R^*	The $\mathbb{R}^{N\times C}$ matrix containing the test mixture components
$\mathcal{R}$	The combined training and test weights
S	The $\mathbb{R}^{N\times M\times C}$ tensor containing all of the training pure component spectra
S^*	The $\mathbb{R}^{N\times M\times C}$ tensor containing all of the test pure component spectra

Notation	Definition
$\mathcal{S}$	The $\mathbb{R}^{N\times M\times C}$ The combined training and test pure component signals
$\hat{S}$	The $\mathbb{R}^{M\times C}$ matrix containing least squares estimate for the pure component signals are static based on the training set
X	The $\mathbb{R}^{N\times A}$ matrix the training spectra
X^*	The $\mathbb{R}^{N^*\times A}$ matrix containing the test spectra
$\mathcal{X}$	The set of both training and test latent variables combined
E	The $\mathbb{R}^{N\times M}$ matrix containing the reconstruction error of the training spectra D using static pure components $\hat{S}$
U	The $\mathbb{R}^{L\times C}$ matrix containing of all the inducing points
V	The $\mathbb{R}^{L\times(A+1)\times C}$ tensor containing of all the input locations of the inducing points
V_c	The $\mathbb{R}^{L\times(A+1)}$ matrix containing the input locations of the inducing points for component c
v_{lc}	The input of inducing point lc
v_{lc}^x	The latent dimensions of inducing point v_{lc}
v_{lc}^λ	The observed dimensions of inducing point v_{lc}
u	The $\mathbb{R}^{LC}$ vectorized form of U
u_c	The $\mathbb{R}^{L}$ vector of the output of the inducing points for c
α_i	The $\mathbb{R}^{c}$ vector of the variational parameters for the Dirichlet distribution of r_i
d_i'	The $\mathbb{R}^{M}$ vector of raw (unpreprocessed) observations of d_i
$\mathcal{L}$	Evidence Lower Bound (ELBO) of the MO-GPLVM model
$\mathcal{E}$	The U dependent terms of $\mathcal{L}$
$D_{KL}\left(\cdot\|\|\cdot\right)$	The KL divergence
$\langle g(\cdot)\rangle_{q(\cdot)}$	The expectation of expression $g(\cdot)$ over the distribution $q(\cdot)$
μ_i^x	Mean of the variational posterior for the position of the latent space for data point i
Σ_i^x	The covariance of the variational posterior for the position of the latent space for data point i
β_c	Length-scales of the ARD kernel for the latent space for component c
γ_c	Length-scales of the kernel for the observed space for component c
σ_s^2	Scale of the kernel
σ^2	Variance of the noise of the Gaussian Process
μ_i^r	The $\mathbb{R}^{C}$ mean estimate of component i
Σ_i^r	The $\mathbb{R}^{C\times C}$ covariance fore the estimate of r_i
μ_{ij}^s	The $\mathbb{R}^{C}$ GP mean of s_{ij} conditional on U
K_{ij}^s	The $\mathbb{R}^{C\times C}$ covariance fore the estimate of s_{ij} conditional on U
K_c^s	The $\mathbb{R}^{NM\times NM}$ covariance for the estimate of $(s_{11c},\ldots,s_{NMc})$ conditional on u_c
$K_{\bullet\bullet}^c$	The $\mathbb{R}^{NM\times NM}$ covariance where each element is given by $k^c((x_i,\lambda_j),(h_{i'},\lambda_{j'}))$
$K_{\bullet V_c}^c$	The $\mathbb{R}^{NM\times L}$ covariance where each element is given by $k^c((x_i,\lambda_j),v_{lc})$
$K_{V_cV_c}^c$	The $\mathbb{R}^{L\times L}$ covariance where each element is given by $k^c(v_{lc},v_{lc})$
K_{VV}	The $\mathbb{R}^{LC\times LC}$ kernel matrix giving the covariance of the inducing points. *Note: This has block diagonal structure of $L\times L$ matrices*
$K_{(x_i\lambda_j)V}$	The $\mathbb{R}^{C\times LC}$ kernel matrix giving the covariance between s_{ij} and the inducing points. *Note: This has block diagonal structure of* $1\times L$ matric
$K_{(x_i\lambda_j)(x_i\lambda_j)}$	The $\mathbb{R}^{C\times C}$ kernel matrix giving the covariance between s_{ij} and the inducing points. *Note: This has a diagonal structure.*
ξ_0	See Equation (D.18). Analogous to ψ_0 in the regular GPLVM model.
ξ_1	See Equation (D.19). Analogous to ψ_1 in the regular GPLVM model.
ξ_2	See Equation (D.20). Analogous to ψ_2 in the regular GPLVM model.

Appendix C

Table of Notation for Chapter 5

Notation	Definition
$\cdot^*$	Star indicates test samples of a variable (where components are unobserved)
$\hat{\cdot}$	Indicates that the quantity refers to the observed components, *i.e.* $c \in \{1, \ldots, \hat{C}\}$
$\bar{\cdot}$	Indicates that the quantity refers to the unobserved components, *i.e.* $c \in \{C+1, \ldots, C\}$
N	Number of observations
M	Number of observations for each sample
$\hat{C}$	The number of observed components that are contained in the samples
$\bar{C}$	The number of unobserved components that are contained in the samples
C	The total number of components present in the samples
L	The number of inducing points for each component
A	The number of Latent dimensions in the MO-GPLVM model
d_{ij}	The observation for sample i at position j
r_{ic}	The component mixture c for sample i
s_{ijc}	The cth pure component spectra evaluated at wavelength j for sample i
x_i	The latent variable of the ith sample
λ_j	The jth location of measurement for all the samples
ϵ_{ij}	The Gaussian noise observed for sample i at position j
r_i	The $\mathbb{R}^C$ vector containing all of the mixture components for sample i
s_{ij}	The $\mathbb{R}^C$ vector containing all of the pure component values for sample i at position j
D	The $\mathbb{R}^{N \times M}$ matrix the training spectra
D^*	The $\mathbb{R}^{N^* \times M}$ matrix containing the test spectra

Notation	Definition
R	The $\mathbb{R}^{N\times C}$ matrix containing the training mixture components
R^*	The $\mathbb{R}^{N\times C}$ matrix containing the test mixture components
$\mathcal{R}$	The combined training and test weights
S	The $\mathbb{R}^{N\times M\times C}$ tensor containing all of the training pure component spectra
S^*	The $\mathbb{R}^{N\times M\times C}$ tensor containing all of the test pure component spectra
$\mathcal{Z}$	The $\mathbb{R}^{N\times M\times C}$ The combined training and test pure component signals
E	The $\mathbb{R}^{N\times M}$ matrix containing the reconstruction error of the training spectra D using static pure components $\hat{F}$
U	The $\mathbb{R}^{L\times C}$ matrix containing of all the inducing points
V	The $\mathbb{R}^{L\times(A+1)\times C}$ tensor containing of all the input locations of the inducing points
V_c	The $\mathbb{R}^{L\times(A+1)}$ containing the input locations of the inducing points for component c
v_{lc}	The input of inducing point lc
v_{lc}^x	The latent dimensions of inducing point v_{lc}
v_{lc}^λ	The observed dimensions of inducing point v_{lc}
u	The $\mathbb{R}^{LC}$ vectorized form of U
u_c	The $\mathbb{R}^{L}$ vector of the output of the inducing points for c
d_i'	The $\mathbb{R}^{M}$ vector of raw (unpreprocessed) observations of d_i
$\mathcal{L}$	Evidence Lower Bound (ELBO) of the MO-GPLVM model
$\mathcal{F}$	The expected likelihood terms of the ELBO
$\mathcal{G}$	The U dependent terms of $\mathcal{L}$

Appendix D

Appendices for Chapter 4

D.1 Additional Results

Table D.1 andD.2 show the mean and standard deviation of the performance metrics for the two regression and one classification experiments. We include the WS-GPLVM-ind for the near infra-red spectroscopy regression and remote sensing rock classification examples for comparison. For both these examples, the WS-GPLVM-ind performs worse than the WS-GPLVM justifying the inclusion of the dependence on wavelengths for these examples. However, it still outperforms the ILMC and CLS-GP, indicating it can still reasonably predict the weights using only the latent variables.

	Spectroscopy		*Oil Flow*	
Method	**NLPD)**	**MSE** $_{[10^{-4}]}$	**NLPD (Oil Flow)**	**MSE** $_{[10^{-4}]}$
WS-GPLVM	-561(9.22)	**2.67(0.78)**	-	-
WS-GPLVM-ind.	-516(57.5)	58.4(45.8)	-7430(81.12)	5.97(4.58)
CLS / CLS-GP	**-693(7.91)**	124(16.2)	-3508(24.6)	81.5(8.2)
ILMC	-309(3.21)	505(25.5)	**-8228(112)**	**2.82(0.42)**
PLS	n/a	3.04(0.73)	n/a	3.81(0.13)

Table D.1: Results of the Weighted-Sum GPLVM (WS-GPLVM), Multi-output Gaussian Process (ILMC), and Partial Least Squares (PLS). Regression results show Negative Log Predictive Density (NLPD) and Mean Squared Error (MSE) of the prediction. Each example was run 10 times and the mean is quoted here, along with two times the standard error of the mean in brackets. For results that deviate significantly from one, powers of ten were removed from all of the results and where applicable these are shown in square brackets at the top of the column. The best mean for each column is in bold.

	Remote Sensing		
Method	**LPP**	**Acc.** $_{[10^{-2}]}$	**AUC**
WS-GPLVM	-14.2(17.1)	81.4(21.3)	**0.91(0.13)**
WS-GPLVM-ind.	-21.0(17.5)	75.7(17.9)	0.89(0.12)
CLS / CLS-GP	-53.7(27.8)	60.0(14.8)	0.81(0.11)
ILMC	**-9.27(2.35)**	30.0(12.5)	0.68(0.18)
PLS	n/a	**82.9(16.2)**	n/a

Table D.2: Results of the Weighted-Sum GPLVM (WS-GPLVM), Multi-output Gaussian Process (ILMC), and Partial Least Squares (PLS). Classification results report the Log Predicted Probability (LPP), the Accuracy (Acc.) and the Area Under the Receiver Operator Curve (AUC). Each example was run 10 times and the mean is quoted here, along with two times the standard error of the mean in brackets. For results that deviate significantly from one, powers of ten were removed from all of the results and where applicable these are shown in square brackets at the top of the column. The best mean for each column is in bold.

D.2 ELBO Derivations

D.2.1 Full ELBO derivation

We start from the ELBO for our variational distribution:

$$\mathcal{L} = \int \log\left(\frac{p(D|R,S)p(D^*|R^*,s^*)\cancel{p(S,S^*|H,H^*,U)}p(R)p(R^*)p(H)p(H^*)p(U)}{\cancel{p(S,S^*|H,H^*,U)}q(R^*)q(H)q(H^*)q(U)}\right) \\ q(F,S^*,H,H^*,U)dR^*dSdS^*dHdH^*dU. \tag{D.1}$$

By setting $q(S, S^*|H, H^*, U) = p(S, S^*|H, H^*, U) = p(S^*|H^*, U)$ the above simplifies to

$$\begin{aligned}\mathcal{L} = &\int \log\left(\frac{p(Y|R,F)p(R)p(H)p(d^*|R^*,s^*)p(R^*)p(H^*)p(U)}{q(R^*)q(H^*)q(H)q(U)}\right) p(S,S^*|H,H^*,U)q(H)q(H^*)\\ &q(R^*)q(U)dR^*dSdS^*dHdH^*dU\\ =&\underbrace{\int \log\left(\frac{p(D,D^*|R,R^*,S,S^*)p(U)}{q(U)}\right) q(F,s^*,H,H^*,U)dUdFdS^*dHdH^*dR^*}_{\mathcal{E}}\\ &- D_{KL}\left(q(H)||p(H)\right) - D_{KL}\left(q(H^*)||p(H^*)\right) - D_{KL}\left(q(R^*)||p(R^*)\right) + \log(p(R)).\end{aligned} \tag{D.2}$$

The challenging terms to deal with are those that depend on U, we therefore collect these into the new term $\mathcal{E}$ which we now focus on.

$$\begin{aligned}\mathcal{E} = \int \frac{-1}{2\sigma^2}&\left(\sum_{i=1}^{N}\sum_{j=1}^{M}(d_{ij} - r_i^T s_{ij})^2 + \sum_{i=1}^{N^*}\sum_{j=1}^{M}(d_{ij}^* - r_i^{*\,T} s_{ij}^*)^2\right)\\ &p(S,S^*|H,H^*,U)q(H)q(H^*)q(R^*)q(U)dUdSdS^*dHdH^*dR^*\\ &-\frac{NM}{2}\log(\sigma^2) - \frac{NM}{2}log(2\pi) - D_{KL}\left(q(U)||p(U)\right)\end{aligned} \tag{D.3}$$

We next take the expectation over R^*, for which we introduce

$$\langle r_i^* \rangle_{q(R^*)} = \mu_i^r \tag{D.4}$$

and

$$\begin{aligned}\langle r_i^* r_i^{*\,T}\rangle_{q(r^*)} =&\langle((r_i^* - \mu_i^r) + \mu_i^r)(r_i^* - \mu_i^r) + \mu_i^{r\,T}))\rangle_{q(r^*)}\\ =&\langle(r_i^* - \mu_i^r)(r_i^* - \mu_i^r)^T + 2\cancel{\mu_i^r(r_i - \mu_i^r)} + \mu_i^r\mu_i^{r\,T}\rangle_{q(r^*)}\\ =&\Sigma_i^r + \mu_i^r\mu_i^{r\,T}\end{aligned} \tag{D.5}$$

for the Dirichlet distribution $Dir(r_i|\alpha_i)$, which is our primary focus, these quantities are given by

$$\mu_i^r = (\mu_{i1}^r, \ldots, \mu_{iC}^r)^T, \quad \text{where} \quad \mu_{ic}^r = \frac{\alpha_{ic}}{\sum_{c=1}^{C}\alpha_{ic}} \tag{D.6}$$

$$\Sigma_i^r = \frac{1}{1 + \sum_{c=1}^{C} \alpha_{ic}} \left(\text{diag}(\mu_i^r) - \mu_{i\cdot}^r \mu_{i\cdot}^{r\,T} \right). \tag{D.7}$$

For the mixture model the classification distribution is the multinomial distribution with one observation, so

$$\mu_i^r = (\alpha_1, \ldots, \alpha_C), \quad \text{where} \quad \sum_{c=1}^{C} \alpha_{ic} = 1 \; \forall \; i \tag{D.8}$$

$$\Sigma_i^r = \text{diag}(\mu_i^r) - \mu_{i\cdot}^r \mu_{i\cdot}^{r\,T} \tag{D.9}$$

These allow us to write

$$\begin{aligned} \mathcal{E} = \int \frac{-1}{2\sigma^2} \Bigg(& \sum_{i=1}^{N} \sum_{j=1}^{M} d_{ij}^2 - 2 d_{ij} r_i^T s_{ij} + s_{ij}^T r_i r_i^T s_{ij} \\ & + \sum_{i=1}^{N^*} \sum_{j=1}^{M} d_{ij}^{*\,2} - 2 d_{ij}^* \mu_i^{r\,T} s_{ij}^* + Tr\left(\left(\mu_i^r \mu_i^{r\,T} + \Sigma_i^r \right) s_{ij}^* s_{ij}^{*\,T} \right) \Bigg) p(S, S^* | H, H^*, U) \\ & q(H) q(H^*) q(U) dS dS^* dH dH^* dU \\ & - \frac{NM}{2} \log(\sigma^2) - \frac{NM}{2} log(2\pi) - D_{KL}\left(q(U) || p(U) \right). \end{aligned} \tag{D.10}$$

The next step is to take the expectation over F. At this point it is should be noted that, while the prior factorizes over each component c, the likelihood factorizes over each observation ij which is the combination of C Gaussian Processes. For this derivation we have been working with the likelihood, it is therefore necessary to be able to write the vector s_{ij} as a linear transformation of of U which requires some new quantities that we introduce now. Firstly, we introduce $u \in \mathbb{R}^{LC} = \text{vec}(U)$ which is the vectorized form of the inducing point outputs U. To go with this we use the block diagonal matrices

$$K_{VV} \in R^{LC \times LC} = \begin{pmatrix} K_{V_1 V_1}^1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & K_{V_C V_C}^C \end{pmatrix} \quad \text{with elements} \quad [K_{V_c V_c}^c]_{ll'} = k_c(v_{lc}, v_{l'c}) \tag{D.11}$$

$$K_{(h_i,\lambda_j)V} \in R^{C\times LC} = \begin{pmatrix} K^1_{(h_i,\lambda_j)V_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & K^C_{(h_i,\lambda_j)V_C} \end{pmatrix} \text{ with elements } [K^c_{(h_i,\lambda_j)V_c}]_{1l} = k_c((h_i,\lambda_j), v_{lc}), \tag{D.12}$$

and

$$K_{(h_i,\lambda_j)(h_i,\lambda_j)} \in R^{C\times C} = \begin{pmatrix} k_1((h_i,\lambda_j),(h_i,\lambda_j)) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & k_C((h_i,\lambda_j),(h_i,\lambda_j)) \end{pmatrix}. \tag{D.13}$$

These matrices allow us to write expressions for the expectations over F as

$$\langle s_{ij}\rangle_{p(F|U,H)} = \mu^s_{ij} = K_{(h_i,\lambda_j)V} K^{-1}_{VV} u \tag{D.14}$$

$$\langle s_{ij} s^T_{ij}\rangle_{p(F|U,H)} = K^s_{ij} + \mu^s_{ij}\mu^{s\,T}_{ij} \quad \text{where} \quad K^s_{ij} = K_{(h_i,\lambda_j)(h_i,\lambda_j)} - K_{(h_i,\lambda_j)V} K^{-1}_{VV} K^T_{(h_i,\lambda_j)V}. \tag{D.15}$$

Using these results we continue the derivation as

$$
\begin{aligned}
\mathcal{E} = & \int \frac{-1}{2\sigma^2}\Bigg(\sum_{i=1}^{N}\sum_{j=1}^{M} -2d_{ij} r_i^T \mu_{ij}^s + Tr\left(r_i r_i^T \left(\mu_{ij}^s \mu_{ij}^{f\,T} + K_{ij}^s \right) \right) \\
& + \sum_{i=1}^{N^*}\sum_{j=1}^{M} -2d_{ij} \mu_i^{r\,T} \mu_{ij}^{s^*} + Tr\left(\left(\mu_i^r \mu_i^{r\,T} + \Sigma_i^r \right) \left(\mu_{ij}^{s^*} \mu_{ij}^{s^*\,T} + K_{ij}^{s^*} \right) \right) \Bigg) q(H) q(H^*) q(U) dH dH^* dU \\
& - \frac{1}{2\sigma^2}\left(\sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij}^2 + \sum_{i=1}^{N^*}\sum_{j=1}^{M} d_{ij}^{*\,2} \right) - \frac{NM}{2}\log(\sigma^2) - \frac{NM}{2} log(2\pi) - D_{KL}\left(q(U) || p(U) \right) \\
= & \int \frac{-1}{2\sigma^2}\Bigg(\sum_{i=1}^{N}\sum_{j=1}^{M} -2d_{ij} r_i^T K_{(h_i\lambda_j)V} K_{VV}^{-1} u + u^T K_{VV}^{-1} K_{(h_i,\lambda_j)V}^T r_i r_i^T K_{(h_i,\lambda_j)V} K_{VV}^{-1} u \\
& + Tr\left(r_i r_i^T \left(K_{(h_i,\lambda_j)(h_i,\lambda_j)} - K_{(h_i,\lambda_j)V} K_{VV}^{-1} K_{(h_i,\lambda_j)V}^T \right) \right) \\
& + \sum_{i=1}^{N^*}\sum_{j=1}^{M} -2d_{ij} \mu_i^{r\,T} K_{(h_i^*\lambda_j)V} K_{VV}^{-1} u + u^T K_{VV}^{-1} K_{(h_i^*\lambda_j)V}^T \left(\mu_i^r \mu_i^{r\,T} + \Sigma_i^r \right) K_{(h_i^*\lambda_j)V} K_{VV}^{-1} u \\
& + Tr\left(\left(\mu_i^r \mu_i^{r\,T} + \Sigma_i^r \right) \left(K_{(h_i^*,\lambda_j)(h_i^*,\lambda_j)} - K_{(h_i^*,\lambda_j)V} K_{VV}^{-1} K_{(h_i^*,\lambda_j)V}^T \right) \right) \Bigg) \\
& \quad q(H) q(H^*) q(U) dH dH^* dU - \frac{1}{2\sigma^2}\left(\sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij}^2 + \sum_{i=1}^{N^*}\sum_{j=1}^{M} d_{ij}^{*\,2} \right) \\
& - \frac{NM}{2}\log(\sigma^2) - \frac{NM}{2} log(2\pi) - D_{KL}\left(q(U) || p(U) \right).
\end{aligned}
\tag{D.16}
$$

By performing some algabraic manipulations it is possible to arrive at the form

$$
\begin{aligned}
\mathcal{E} = & \int \frac{-1}{2\sigma^2}\Bigg(-2\left(\sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij} r_i^T K_{(h_i,\lambda_j)V} + \sum_{i=1}^{N^*}\sum_{j=1}^{M} d_{ij}^* \mu_i^{r\,T} K_{(h_i^*,\lambda_j)V} \right) K_{VV}^{-1} u \\
& + u^T K_{VV}^{-1} \left(\sum_{i=1}^{N}\sum_{j=1}^{M} K_{(h_i,\lambda_j)V}^T r_i r_i^T K_{(h_i,\lambda_j)V} + \sum_{i=1}^{N^*}\sum_{j=1}^{M} K_{(h_i^*,\lambda_j)V}^T \left(\mu_i^r \mu_i^{r\,T} + \Sigma_i^r \right) K_{(h_i^*,\lambda_j)V} \right) K_{VV}^{-1} u \\
& + \sum_{i=1}^{N}\sum_{j=1}^{M} Tr\left(r_i r_i^T \left(K_{(h_i,\lambda_j)(h_i,\lambda_j)} - K_{(h_i\lambda_j)V}^T r_i r_i^T K_{(h_i\lambda_j)V} K_{VV}^{-1} \right) \right) \\
& + \sum_{i=1}^{N^*}\sum_{j=1}^{M} Tr\left(\left(\mu_i^r \mu_i^{r\,T} + \Sigma_i^r \right) \left(K_{(h_i^*\lambda_j)(h_i^*\lambda_j)} - K_{(h_i\lambda_j)V} K_{VV}^{-1} K_{(h_i\lambda_j)V}^T \right) \right) \Bigg) q(H) q(H^*) q(U) dH dH^* dU \\
& - \frac{1}{2\sigma^2}\left(\sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij}^2 + \sum_{i=1}^{N^*}\sum_{j=1}^{M} d_{ij}^{*\,2} \right) - \frac{NM}{2}\log(\sigma^2) - \frac{NM}{2} log(2\pi) - D_{KL}\left(q(U) || p(U) \right).
\end{aligned}
\tag{D.17}
$$

which gathers the terms for which we need to compute the expectation over H and H^*. We

can write these terms using three quantities which we introduce here:

$$\xi_0 = \left\langle \sum_{i=1}^{N}\sum_{j=1}^{M} Tr\left(r_i r_i^T K_{(h_i\lambda_j)(h_i\lambda_j)}\right) + \sum_{i=1}^{N^*}\sum_{j=1}^{M}\left(Tr\left(\left(\mu_i^r \mu_i^{r\,T} + \Sigma_{i\cdot}^r\right) K_{(h_i^*\lambda_j)(h_i^*\lambda_j)}\right)\right)\right\rangle_{q(H,H^*)} \tag{D.18}$$

$$\begin{aligned}\xi_1 &= \left\langle \sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij} r_i^T K_{(h_i\lambda_j)V} + \sum_{i=1}^{N^*}\sum_{j=1}^{M} d_{ij}^* \mu_i^{r\,T} K_{(h_i^*\lambda_j)V}\right\rangle_{q(H,H^*)} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij} r_i^T \left\langle K_{(h_i,\lambda_j)V}\right\rangle_{q(H)} + \sum_{i=1}^{N^*}\sum_{j=1}^{M} d_{ij}^* \mu_i^{r\,T} \left\langle K_{(h_i^*\lambda_j)V}\right\rangle_{q(H^*)}\end{aligned} \tag{D.19}$$

$$\begin{aligned}\xi_2 &= \left\langle \sum_{i=1}^{N}\sum_{j=1}^{M} K_{(h_i\lambda_j)V}^T r_i r_i^T K_{(h_i\lambda_j)V} + \sum_{i=1}^{N^*}\sum_{j=1}^{M} K_{(h_i^*\lambda_j)V}^T \left(\mu_i^r \mu_i^{r\,T} + \Sigma_i^r\right) K_{(h_i\lambda_j)V}\right\rangle_{q(H,H^*)} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} \left\langle K_{(h_i\lambda_j)V}^T r_i r_i^T K_{(h_i\lambda_j)V}\right\rangle_{q(H)} + \sum_{i=1}^{N^*}\sum_{j=1}^{M} \left\langle K_{(h_i^*\lambda_j)V}^T \left(\mu_i^r \mu_i^{r\,T} + \Sigma_i^r\right) K_{(h_i^*\lambda_j)V}\right\rangle_{q(H^*)}\end{aligned} \tag{D.20}$$

Using these quantities and the the fact that $p(u) = N(u|0, K_{VV})$ we can write

$$\begin{aligned}\mathcal{E} = &\int \left(\frac{1}{\sigma^2}\xi_1 K_{VV}^{-1} u - \frac{1}{2} u^T K_{VV}^{-1}\left(\frac{-1}{\sigma^2}\xi_2 + K_{VV}\right) K_{VV}^{-1} u\right) \log\left(\frac{1}{q(u)}\right) q(U)\, dU - \frac{M}{2} log(2\pi) \\ &- \frac{1}{2} log(det(K_{VV})) + \frac{1}{2\sigma^2} Tr\left(K_{VV}^{-1}\xi_2\right) + \frac{1}{2\sigma^2}\xi_0 - \frac{1}{2\sigma^2}\left(\sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij}^2 + \sum_{i=1}^{N^*}\sum_{j=1}^{M} d_{ij}^{*\,2}\right) \\ &- \frac{NM}{2}\log(\sigma^2) - \frac{NM}{2} log(2\pi).\end{aligned} \tag{D.21}$$

We can use the approach from King and Lawrence [2006] to find the optimal distribution for u. We do this by noting that the expression in the integral can be rewritten as a quadratic form

with the quantities $\Sigma_u = K_{VV}\left(\frac{1}{\sigma^2}\xi_2 + K_{VV}\right)^{-1} K_{VV}$ and $\mu_u = \Sigma_u K_{VV}^{-1}\xi_1$, giving

$$\begin{aligned}\mathcal{E} = \int -\frac{1}{2}(u-\mu_u)^T\Sigma_u^{-1}(u-\mu_u)\log\left(\frac{1}{q(u)}\right)q(u)du + \frac{M}{2}log(2\pi) - \frac{1}{2}log(det(K_{VV})) \\ + \frac{1}{2\sigma^4}\xi_1^T\left(\frac{1}{\sigma^2}\xi_2 + K_{VV}\right)^{-1}\xi_1 - \frac{1}{2}log\left(det\left(\frac{1}{\sigma^2}\xi_2 + K_{VV}\right)\right) + \frac{1}{2}\log(det(K_{VV})) \\ - \frac{1}{2\sigma^2}\xi_0 + \frac{1}{2\sigma^2}Tr\left(K_{VV}^{-1}\xi_2\right) - \frac{1}{2\sigma^2}\left(\sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij}^2 + \sum_{i=1}^{N^*}\sum_{j=1}^{M} {d_{ij}^*}^2\right) - \frac{NM}{2}\log(\sigma^2) - \frac{NM}{2}log(2\pi).\end{aligned} \tag{D.22}$$

By setting $q(u) \sim N(\mu^u, \Sigma^u)$ and performing some algebra we arrive at the final form of $\mathcal{E}$ of

$$\begin{aligned}\mathcal{E} = \frac{1}{2\sigma^4}\xi_1^T\left(\frac{1}{\sigma^2}\xi_2 + K_{VV}\right)^{-1}\xi_1 - \frac{1}{2}log\left(det\left(\frac{1}{\sigma^2}\xi_2 + K_{VV}\right)\right) + \frac{1}{2}\log(det(K_{VV})) \\ - \frac{1}{2\sigma^2}\xi_0 + \frac{1}{2\sigma^2}Tr\left(K_{VV}^{-1}\xi_2\right) - \frac{1}{2\sigma^2}\left(\sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij}^2 + \sum_{i=1}^{N^*}\sum_{j=1}^{M} {d_{ij}^*}^2\right) - \frac{NM}{2}\log(\sigma^2) - \frac{NM}{2}log(2\pi).\end{aligned} \tag{D.23}$$

By combining this with the terms which are independent of u that we set aside initially we arrive at

$$\begin{aligned}\mathcal{L} = &\frac{1}{2\sigma^4}\xi_1^T\left(\frac{1}{\sigma^2}\xi_2 + K_{VV}\right)^{-1}\xi_1 - \frac{1}{2}log\left(det\left(\frac{1}{\sigma^2}\xi_2 + K_{VV}\right)\right) - \frac{1}{2\sigma^2}\xi_0 \\ &+ \frac{1}{2\sigma^2}Tr\left(K_{VV}^{-1}\xi_2\right) - \frac{1}{2\sigma^2}\left(\sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij}^2 + \sum_{i=1}^{N^*}\sum_{j=1}^{M} {d_{ij}^*}^2\right) - \frac{NM}{2}\log(\sigma^2) - \frac{NM}{2}log(2\pi) \\ &+ \frac{1}{2}\log(det(K_{VV})) - KL\left(q(h)||p(h)\right) - KL\left(q(R^*)||p(R^*)\right).\end{aligned} \tag{D.24}$$

D.2.2 Derivation of the ELBO with Independent Measurements

Note:

In this derivation we present a derivation of the ELBO where we do not assume that the measurement is smooth and can be described by a Gaussian Process across the observed inputs. The change in the Gaussian Process prior is that it it now factorizes across the MC vectors of

$s_{jc} \in \mathbb{R}^N$

$$p(F) = \prod_{c=1}^{C} \prod_{j=1}^{M} N(s_{jc}|0, K^{jc}_{V_{jc}V_{jc}}). \tag{D.25}$$

As we are working with the variational sparse changing the prior for F requires changing the inducing points and their prior such that they are drawn from the output of the GP. This requires increasing the number of inducing points from LC to MLC, as now the Gaussian process factorizes over the M observations. To account for this we change the notation in this section so that $U \in \mathbb{R}^{M\times L\times C}$, made up of MC sets of $u_{jc} \in \mathbb{R}^L$, and $V \in \mathbb{R}^{M\times L\times C\times A}$, made up of MC sets of $u_{jc} \in \mathbb{R}^{L\times A}$, and has a prior

$$p(U) = \prod_{c=1}^{C} \prod_{j=1}^{M} N\left(u_{jc}|0, K^{jc}_{V_{jc}V_{jc}}\right), \tag{D.26}$$

where the kernel matrix is now a $N \times N$ matrix with elements

$$\left[K^{jc}_{V_{jc}V_{jc}}\right]_{ii'} = k_{jc}(h_i, h_{i'}). \tag{D.27}$$

As it is impractical to find independent kernels and inducing points for each of the measurements and locations, for the remainder of this derivation we assume that $k_{jc}(h_i, h_{i'}) = k_c(h_i, h_{i'})$ and $V_{jc} = V$ for all j and c, allowing us to maintain the same definitions for the kernel matrix K_{VV} as defined in Equation (D.11), and the kernel matrices defined in Equation (D.12) and Equation (D.13) only need to be adapted to K_{h_iV} and $K_{h_ih_i}$ by changing the kernel function to

$$k_c(h_i, h_{i'}) = \sigma^2_{s_c} exp\left(\frac{1}{2}(h_i - h_{i'})^T\beta^{-1}(h_i - h_{i'})\right). \tag{D.28}$$

With this the derivations are identical until Equation (D.17), which is changed to include the fact that now each of the measurement locations depends on a different set of inducing points

$u_j \in \mathbb{R}^{LC}$. This modifies the equation to be

$$\begin{aligned}\mathcal{E} = &\sum_{j=1}^{M}\Bigg[\int \frac{-1}{2\sigma^2}\Bigg(-2\left(\sum_{i=1}^{N} d_{ij} r_i^T K_{h_iV} + \sum_{i=1}^{N} d_{ij}\mu_i^{r\,T} K_{h_iV}\right) K_{VV}^{-1} u_j \\ &+ u_j^T K_{VV}^{-1}\left(\sum_{i=1}^{N} K_{h_iV}^T r_i r_i^T K_{h_iV} + \sum_{i=1}^{N^*} K_{h_iV}^T\left(\mu_i^r \mu_i^{r\,T} + \Sigma_i^r\right) K_{h_iV}\right) K_{VV}^{-1} u_j \\ &+ \sum_{i=1}^{N}\sum_{j=1}^{M} Tr\left(r_i r_i^T K_{h_ih_i}\right) - Tr\left(K_{h_iV}^T r_i r_i^T K_{h_iV} K_{VV}^{-1}\right) \\ &+ \sum_{i=1}^{N^*}\sum_{j=1}^{M} Tr\left(\left(\mu_i^r \mu_i^{r\,T} + \Sigma_i^r\right) K_{h_ih_i} - Tr\left(K_{h_iV} K_{VV}^{-1} K_{h_iV}^T\right)\right)\Bigg) q(H) q(H^*) q(U) dH dU\Bigg] \\ &- \frac{1}{2\sigma^2}\left(\sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij}^2 + \sum_{i=1}^{N^*}\sum_{j=1}^{M} d_{ij}^{*\,2}\right) - \frac{NM}{2}\log(\sigma^2) - \frac{NM}{2} log(2\pi) - D_{KL}\left(q(U)||p(U)\right).\end{aligned} \tag{D.29}$$

Note that the terms inside of the sum are identical to those those in the correlated measurements case, once the kernel matrices and inducing points have been redefined. So, by simply redefining the values of ξ_i to remove dependence on λ all of the same arguments as above hold. These new terms, equivalent to Equations (D.18)-(D.20), are

$$\xi_0 = \sum_{i=1}^{N}\left\langle Tr\left(r_i r_i^T K_{(h_i\lambda_j)(h_i\lambda_j)}\right)\right\rangle_{q(H)} + \sum_{i=1}^{N^*}\left\langle Tr\left(\left(\mu_i^r \mu_i^{r\,T} + \Sigma_{i\cdot}^r\right) K_{h_ih_i}\right)\right\rangle_{q(H^*)} \in \mathbb{R} \tag{D.30}$$

$$\xi_{1j} = \sum_{i=1}^{N} d_{ij} r_i^T \left\langle K_{h_iV}\right\rangle_{q(H)} + \sum_{i=1}^{N^*} d_{ij}^* \mu_i^{r\,T} \left\langle K_{h_iV}\right\rangle_{q(H^*)} \in \mathbb{R}^{C\times MC} \tag{D.31}$$

and

$$\xi_2 = \sum_{i=1}^{N}\left\langle K_{h_iV}^T r_i r_i^T K_{h_iV}\right\rangle_{q(H)} + \sum_{i=1}^{N^*}\left\langle K_{h_iV}^T\left(\mu_i^r \mu_i^{r\,T} + \Sigma_i^r\right) K_{h_iV}\right\rangle_{q(H^*)}. \in \mathbb{R}^{MC\times MC} \tag{D.32}$$

Note that (provided the same kernel and inducing points are used for all j) the expressions for ξ_0 and ξ_2 are both independent of the measurement location, whereas ξ_{1j} does depend on the values measured at the specific measurement locations - demanding the additional index j.

Using this insight we skip to the final definition of the ELBO as

$$\begin{aligned}\mathcal{L} = &\sum_{j=1}^{M}\left[\frac{1}{2\sigma^4}\xi_{1j}^T\left(\frac{1}{\sigma^2}\xi_2 + K_{VV}\right)^{-1}\xi_{1j} - \frac{1}{2}log\left(det\left(\frac{1}{\sigma^2}\xi_2 + K_{VV}\right)\right) - \frac{1}{2\sigma^2}\xi_0 + \frac{1}{2\sigma^2}Tr\left(K_{VV}^{-1}\xi_2\right)\right] \\ &- \frac{1}{2\sigma^2}\left(\sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij}^2 + \sum_{i=1}^{N^*}\sum_{j=1}^{M} d_{ij}^{*\,2}\right) - \frac{NM}{2}\log(\sigma^2) - \frac{NM}{2}log(2\pi) + \frac{1}{2}\log(det(K_{VV})) \\ &- KL\left(q(H)||p(H)\right) - KL\left(q(H^*)||p(H^*)\right) - KL\left(q(R^*)||p(R^*)\right).\end{aligned} \tag{D.33}$$

Relative Computational Cost

While increasing the number of latent variables from LC, as for the first ELBO derivation, to LMC, for the derivation presented here, may sound computationally intensive, the new inducing points now only need to describe the variation across latent variables of a single of a single measurement location, rather than needing to describe the variation across all the latent values and all the measurements together. As the information of a single measurement location is clearly significantly less than that for all the measurements collectively, a drastically lower number of inducing points can be used to describe each of the vectors $s_{:jc} \in \mathbb{R}^N$ than would be required to describe $s_{::c} \in \mathbb{R}^{NM}$. Furthermore, as working with the MO-GPLVM requires inverting a $LC \times LC$ matrix, it is generally more computationally efficient to work considering all of the M observations to be independent - although at the cost of throwing away the information associated with the measurement locations.

D.3 Expressions for ξ_i

The ξ_i values are expectations over three kernel matrices, with expressions given in Equation (D.18)-(D.20). For this work we have only considered a single ARD kernel shared between all of the components as this allows us to calculate the expressions for ξ_i analytically, however it is possible to use other Kernels and compute the expectation using numerical methods de Souza

et al. [2021], Lalchand et al. [2022].

For the ARD kernels $k_c(\cdot,\cdot) = k(\cdot,\cdot)\ \forall\ c = 1, ..., C$.

$$k((h_i, \lambda_j), (h_{i'}, \lambda_{j'})) = \sigma_s^2 exp\left(-\frac{1}{2}\left(h_i - h_{i'}\right)^T \beta^{-1}\left(h_i - h_{i'}\right)\right) exp\left(-\frac{1}{2}\left(\lambda_j - \lambda_{j'}\right)^T \gamma^{-1}\left(\lambda_j - \lambda_{j'}\right)\right) \tag{D.34}$$

the values are given by

$$\xi_0 = M\left(\sum_{c=1}^{C} \sigma_s^2 \left(\sum_{i=1}^{N} r_{ic}^2 + \sum_{i=1}^{N^*} {\mu_{ic}^r}^2 + [\Sigma_i^r]_{cc}\right)\right) \tag{D.35}$$

$$\xi_1 = \begin{pmatrix} [\xi_1]_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & [\xi_1]_C \end{pmatrix} \text{ where } [\xi_1]_c \in \mathbb{R}^{1\times L} \tag{D.36}$$

and the elements of

$$\begin{aligned}[[\xi_1]_c]_{1l} = \\ \sum_{ij} d_{ij} r_{ic} \sigma_{s_c}^2 \sqrt{\frac{\det(\beta)}{\det\left(\beta_c + \Sigma_i^h\right)}} exp\left(-\frac{1}{2}(\mu_i^h - v_{lc}^h)^T \left(\beta + \Sigma_i^h\right)^{-1} (\mu_i^h - v_{lc}^h) + (\lambda_j - v_{lc}^\lambda)^T \gamma^{-1} (\lambda_j - v_{lc}^\lambda)\right) \\ + \sum_{ij} d_{ij}^* \mu_{ic}^r \sigma_{s_c}^2 \sqrt{\frac{\det(\beta)}{\det\left(\beta + \Sigma_i^{h^*}\right)}} exp\left(-\frac{1}{2}(\mu_i^{h^*} - v_{lc}^{h^*})^T \left(\beta + \Sigma_i^{h^*}\right)^{-1} (\mu_i^{h^*} - v_{lc}^{h^*}) + (\lambda_j - v_{lc}^\lambda)^T \gamma^{-1} (\lambda_j - v_{lc}^\lambda)\right)\end{aligned} \tag{D.37}$$

$$\xi_2 = \begin{pmatrix} [\xi_2]_{11} & \dots & [\xi_2]_{1C} \\ \vdots & \ddots & \vdots \\ [\xi_2]_{C1} & \dots & [\xi_2]_{CC} \end{pmatrix} \text{ where } \xi_{2(cc)} \in \mathbb{R}^{L\times L} \tag{D.38}$$

and

$$[[\xi_2]_{cc'}]_{ll'} = \left(-\frac{1}{4}(v_l^h - v_{l'}^h)^T\beta^{-1}(v_l^h - v_{l'}^h)\right)\sum_{j=1}^{M}\sigma_{s_c}^4 exp\left((\lambda_j - v_{lc}^\lambda)^T\gamma^{-1}(\lambda_j - v_{lc}^\lambda)\right)exp\left((\lambda_j - v_{l'c}^\lambda)^T\gamma^{-1}(\lambda_j - v_{l'c}^\lambda\right.$$
$$\left(\sum_{i=1}^{N} d_{ij}r_{ic}r_{ic'}\sqrt{\frac{\det(\beta)}{\det\left(\beta + 2\Sigma_i^h\right)}}exp\left(-\frac{1}{2}\left(\mu_i^h - \frac{v_{lc}^h - v_{l'c}^h}{2}\right)^T\left(\beta + 2\Sigma_i^h\right)^{-1}\left(\mu_i^h - \frac{v_{lc}^h - v_{l'c}^h}{2}\right)\right)\right.$$
$$+\sum_{i=1}^{N^*} d_{ij}^*\left(\mu_{ic}^r\mu_{ic'}^r + [\Sigma_i^r]_{cc'}\right)\sqrt{\frac{\det(\beta)}{\det\left(\beta + 2\Sigma_i^{h^*}\right)}}$$
$$\left.exp\left(-\frac{1}{2}\left(\mu_i^{h^*} - \frac{v_{lc}^h - v_{l'c}^h}{2}\right)^T\left(\beta + 2\Sigma_i^{h^*}\right)^{-1}\left(\mu_i^{h^*} - \frac{v_{lc}^h - v_{l'c}^h}{2}\right)\right)\right) \tag{D.39}$$

D.4 Code, Data and Computation

D.4.1 Comparison Methods

Classical Least Squares/Classical Least Squares Gaussian Process

The first method we compare against is the Classical Least Squares/Classical Least Squares Gaussian-Process (CLS/CLS-GP), which correspond the the baseline is equivalent to the WS-GPLVM without the latent variables. CLS corresponds to the WS-GPLVM-ind. with no covariance between wavelengths, while CLS-GP includes a GP prior across the wavelengths.

We treat this model in a Bayesian way, with the same priors as the WS-GPLVM/WS-GPLVM-ind., with the omission of the latent variables. This model does not have a conjugate posterior and we use a variational approximation, using a variational posterior. For the CLS model we use the variational posterior

$$q(S) = \prod_{j=1}^{M} q(s_j) \quad \text{where} \quad q(s_j) = N(\mu_j^s, \Sigma_j^s) \tag{D.40}$$

where $\mu_j^s \in \mathbb{R}^C, \Sigma_j^s \in \mathbb{R}^{C\times C}$ are the variational parameters. For the CLS-GP we use

$$q(S) = N(s|\mu^s, \Sigma^s) \tag{D.41}$$

where $\mu^s \in \mathbb{R}^{MC}, \Sigma^s \in \mathbb{R}^{MC \times MC}$ are the variational parameters. Note that because the pure signals do not vary with a latent variable in this model, they are equal across all data points, so we do not need to index over i. With this distribution and the likelihood in Equation (4.1) the ELBO can be found using

$$\mathcal{L}_{CLS} = \langle p(D|S, R^*, R) \rangle_{q(F,R^*)} - KL(q(S)||p(S)) - KL(q(R^*)||p(R^*)) + log(p(R)). \quad \text{(D.42)}$$

The variational parameters (Dirichlet concentration parameter α_i, μ_j^s and Σ_j^s), along with the GP parameters for CLS-GP, are optimized numerically.

Inverse Linear Model of Coregionalization (ILMC)

While the true data generating process that we consider in spectroscopy is one in which the observations are a function of the weights, it is also possible to predict the weights using an inverse model, such as

$$r_i = g(d_i) + \eta_i, \quad \text{(D.43)}$$

where $\eta_i \in \mathbb{R}^C$ is some noise. As this model reverses causality, in that the weights are a function of the data rather than the data being a function of the weights, we refer to it as an inverse model.

As we are comparing to Bayesian methodologies we model $g(\cdot)$ with a Gaussian Process. We use a linear model of coregionalisation model as, due to the sum-to-one constraint, pure components covary with one another. Additionally, we used a sparse GP Hensman et al. [2013] for computational speed.

For datasets with high input dimensions, *i.e.* the spectroscopy regression and classification examples, principal component analysis (PCA) is applied to Y and d^* before fitting the model, similar to Chen et al. [2007b]. The Gaussian process then takes the PCA reduced data, $d_i^{PCA} \in \mathbb{R}^{M_{\text{PCA}}}$ as input

$$r_i = g(d_i^{\text{PCA}}) + \eta_i, \quad \text{(D.44)}$$

where $M_{\text{PCA}} < M$. We selected the number of principal components, N_{PCA}, by fitting multiple ILMC models to PCA-reduced data with different dimensions and selecting the one with the highest ELBO.

For classification the prediction from the LMC was put through a Softmax Likelihood and a sparse GP variational inference was used to make predictions Hensman et al. [2015].

D.5 Implementation Details for Comparison Methods in Chapter 4

Hyperparameter	Spectroscopy regression	Oil flow regression	Spectroscopy classification
noise variance σ^2	1	1	1
kernel variance σ_s^2	Uniform(0.5, 1)	Uniform(0.5, 1)	Uniform(0.5, 1)
latent lengthscale β	Uniform(0.5, 5.5)	Uniform(0.1,1.1)	Uniform(0.1,1.1)
latent inducing points V^h	Normal(0, 1)	Normal(0, 1)	Normal(0, 1)
number of latent inducing points L_λ	6	50	16
input lengthscale γ	1000	n/a	40000
input inducing points V^λ	grid	n/a	grid
number of input inducing points L_h	28	n/a	20
latent dimensions A	2	5	5

Table D.3: Hyperparameter initializations for WS-GPLVM and WS-GPLVM-ind. When the initialization is a distribution, the initial value is drawn from that distribution. Initalization is 'grid' means a grid of points across the whole input domain.

Optimization As described in Section 4.3, the optimization of the WS-GPLVM is done in three steps. (1) The hyperparameters are trained on the training data without allowing h to train using Adam [Kingma and Ba, 2014] with a learning rate of 0.05. (2) The noise variance is attenuated over 20 steps from 1 to the value it had trained to in the previous stage. At each stage, two steps of the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimizer are taken. (3) All parameters are trained together using L-BFGS.

Random Restarts Optimizing the ELBO with respect to GP hyperparameters often encounters local optima [Rasmussen, 2003, Ch. 5], making the results sensitive to initialization. To address this, we use random restarts with varied initializations, selecting the run that yields the highest ELBO.

Increasing the number of restarts can either improve the final ELBO or maintain the current best if the optimal solution has already been found. To determine an appropriate number of restarts for the WS-GPLVM, we conducted additional trials: 20 restarts for the oil flow experiment and 32 for the remote sensing rock classification. While it can't be proven if the global optimum has been reached, the ELBO plateaued after 15 restarts for the oil flow data and after just 5 for the rock classification, indicating that a few restarts are typically sufficient to reach a good local optimum.

D.5.1 Inverse Linear model of Coregionalization

Kernel The ILMC uses the squared exponential kernel:

$$k(d_n^{PCA}, d_{n'}^{PCA}) = \sigma_f^2 \exp\left(-\frac{1}{2}(d_n^{PCA} - d_{n'}^{PCA})^T \gamma^{-1} (d_n^{PCA} - d_{n'}^{PCA})\right). \tag{D.45}$$

Hyperparameter and Inducing Point Initialization For the ILMC baseline, the lengthscales γ and kernel variances σ_f^2 are initialized to one. 50 inducing points are initialized as $v_\lambda \sim \text{Normal}(0, 1)$.

Optimization We use Adam [Kingma and Ba, 2014] to optimize the ILMC hyperparameters. We use a learning rate of 0.01 for the spectroscopy regression and classification experiments, and 0.1 for the oil flow experiment.

D.5.2 Classical Least Squares Gaussian Process

The Classical Least Squares Gaussian process corresponds to the WS-GPLVM model without the latent space to allow the pure component signals to vary.

Kernel The CLS-GP uses the squared exponential kernel, shown in Equation D.45 for the pure component signals.

Hyperparameter and Inducing Point Initialization The CLS-GP hyperparameters are initialized in the same way as the WS-GPLVM, shown in Table D.3, except the CLS-GP has no latent lengthscale, dimensions or inducing points. The weights are initialized in the same way.

Optimization All the CLS parameters are optimized together using L-BFGS.

D.5.3 Classical Least Squares

Classical Least Squares corresponds to the WS-GPLVM-ind. model without the latent space to allow the pure component signals to vary.

Hyperparameter Initialization For CLS, the weights were initialized to the prior as they are for the WS-GPLVM. The noise variance was initialized to 0.01.

Optimization The weights and CLS parameters are optimized together using L-BFGS.

D.5.4 Partial Least Squares

For both Chapter 4 and 5 we use the Scikit Learn version 1.2 Pedregosa et al. [2011] implementation for PLS. The only hyperparameter in PLS is the number of components. We select the number of components by running a 10-fold cross validation, or two-fold for the oil flow dataset, on the training set and selecting the one with the lowest mean squared error.

D.6 Additional Details for Experiment Setup

D.6.1 Toy Example

For this data set, we used a 5 dimensional latent for the WS-GPLVM with a grid of inducing points with 8 inducing point locations in the variable space and 16 inducing points in the observed space, for a total of 128 inducing points.

D.6.2 Near Infra Red Spectroscopy with varying temperatures

The original near infra red spectroscopy dataset contained a region of constant signal at low wavelengths was wel as very noisy signals at the highest wavelengths. We removed both these regions before the pre-processing and analysis steps, truncating the wavelengths to be between 800-1000nm. For this experiments we used 2 latent dimensions, 6 latent inducing points and 28 input inducing points.

D.6.3 Oil Flow Regression

This example was run with independent measurement locations. For the WS-GPLVM, we used 50 inducing points and 5 latent dimensions.

D.6.4 Remote Sensing Rock Classification

Similar to the near infra-red spectroscopy regression example, the largest wavelengths ($>$ $2250nm$) were noisy so we removed them from the data. For the WS-GPLVM we used 5 latent dimensions, 16 latent inducing points and 20 input inducing points. We found this number of inducing points to give a reasonable trade-off between computational time and quality of the ELBO optimization.