Latent representation for the characterisation of mental diseases

by

Carlos Sevilla Salcedo

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in

Multimedia and Communications

Universidad Carlos III de Madrid

Advisor: Vanessa Gómez Verdejo

July 2021

A mi esposa y mi familia, la sal y la luz de mi vida.

Acknowledgements

Let me switch to my mother tongue for only part of these acknowledgements, so that my mother can understand who I am mentioning and what I am saying. I will revert to English at the end to thank the international people who have helped me throughout this PhD.

Que mi querida madre me lo perdone, pero a la primera persona a la que me gustaría agradecer su colaboración, ayuda y empatía es a mi directora, Vanessa. No podía imaginarme yo cuando empecé tu asignatura en el máster hasta que punto iba a formar parte de mi vida, entre otras cosas porque ni me había planteado hacer nunca un doctorado. Realmente has sido y eres una inspiración, sobre todo, como profesora. Probablemente no lo recuerdes, pero en una de nuestras conversaciones en tu despacho me dijiste algo así como: "*Yo primero soy profesora, lo demás va después*" y la verdad es que se nota en la forma en la que te desvives por todos los que hemos tenido la suerte de tenerte como profesora. Gracias por tu paciencia a la hora de trabajar en desarrollos matemáticos, probar código y en darle vueltas y más vueltas a todo lo que hemos escrito en nuestras investigaciones. Y no sólo eso, es que me siento apreciado y querido, y eso no lo da un puesto, un cargo o una posición, eso lo da una persona con auténtica vocación por su trabajo y con un gran corazón.

Por supuesto, como anticipaba, mi madre tampoco se queda muy atrás, y mi padre va siempre acompañándola. Pudiendo haber criado una piara de marranos, en un alarde de bondad y amor extremo decidieron, en su lugar, criarme a mí y acompañarme y preocuparse por mí el resto de sus vida. Creo que los cerdos les hubieran salido más rentables. Hablando en serio, gracias a los dos por vuestra preocupación por mi formación, por haber sido un ejemplo constante de que la vida es maravillosa y hasta la situación más extraña se puede disfrutar con intensidad. Y de ellos paso a su otro hijo favorito. Javier, no podías faltar en estos agradecimientos tú tampoco porque tampoco has faltado en mi tiempo de doctorado. Y la verdad es que parecía que de canciones y comida iba la cosa, pero desde el año pasado también hemos tenido nuestro proceso de reencuentro en la universidad. Gracias también por demostrarme lo que es sentir pasión por la ciencia, por la tecnología y por la investigación. Aunque no te lo creas, eres todo un ejemplo de dedicación y esfuerzo para un trabajo bien hecho.

Mi maravillosa esposa se conoce todo lo que aparece en esta tesis doctoral casi mejor que yo. Tiene mérito tener la paciencia para aguantar a una persona frustrada divagando durante horas porque unos datos que se supone que deberían seguir una distribución Gamma tienen forma de Normal. Y es que esta tesis doctoral es tan mía como tuya. Gracias por tu cariño y tu respeto, por apoyarme en esta cruzada y en las que nos quedan por vivir. Por supuesto, también quiero agradecer enormemente a Marga, Juan y Alejandra porque siempre habéis mostrado interés por mi investigación y eso me da fuerza para seguir adelante. Además, no hay cosa más bonita que lo que me decía una vez Marga sobre mis artículos: "*Yo, la verdad, es que no entiendo nada, pero me parecen súper interesantes y muy bonitos*".

Por otro lado, mi experiencia doctoral me ha regalado conocer a personas maravillosas que han sido un fuerte apoyo en este doctorado. Empezando por la última incorporación, Dani, si decía que Elena me ha aguantado mis divagaciones, tú no sólo me las has aguantado, sino que también me has intentado ayudar, aunque muchas veces acabes diciendo: "*¡Si al final no te he ayudado!*". Y es que esta última etapa de la escritura de la tesis no habría sido lo mismo sin el pequeño resumen de lo que he hecho y lo que me queda por hacer antes de irme a casa. Gracias también a Alexander y Lorena por, respectivamente, acompañarme con mis maravillosos chistes malos y no entenderlos, pero aún así hacer que tengamos siempre ese clima cómodo en el que trabajar. Gracias a Elena, la mejor secretaria del departamento, por preocuparse tanto por mí. Ha sido corto el tiempo que hemos estado juntos en esta universidad, pero parece que te conociera desde hace mucho tiempo.

También he tenido la suerte de poder contar con Pablo, que me ha enseñado a apreciar los modelos Bayesianos con mucha paciencia y disponibilidad. Por otro lado, también quiero agradecer a Albert su ayuda en la identificación de áreas del cerebro, así como las funciones que tienen. También he tenido el gran apoyo de Fernando, gran profesor y mejor persona, que ha sido partícipe de todo el proceso de esta tesis, en especial de las últimas revisiones con Vanessa antes y después de sus clases.

Back to English, I would like to thank Jussi for opening the doors of his lab and showing me the wonders of neuroscience. Finally, I would like to thank Angus for his help in the writing process and for his eagerness to exchange Scottish and Granada's slang to make our videoconferences more entertaining.

Published and presented contents

Most of the analyses and results presented in this thesis have been either published or are submitted for publication. In particular, the published articles are:

- Regularized bagged canonical component analysis for multiclass learning in brain imaging. Published in Neuroinformatics in 2020. Written by Carlos Sevilla Salcedo (main author), Vanessa Gómez Verdejo and Jussi Tohka. DOI: 10.1007/s12021-020-09470-y. This article is fully included in Chapter 3. The code of the proposed model and an exemplary notebook are openly available at regMVA. The material from this source included in this thesis is not singled out with typographic means and references, although it is indicated at the beginning of the chapter.

- Sparse semi-supervised heterogeneous interbattery bayesian analysis. Published in Pattern Recognition in 2021. Written by Carlos Sevilla Salcedo (main author), Vanessa Gómez Verdejo and Pablo Martínez Olmos. DOI: 10.1016/j.patcog.2021.108141. This article is fully included in Chapter 4. A library with the proposed model and exemplary notebooks are openly available at SSHIBA. The material from this source included in this thesis is not singled out with typographic means and references, although it is indicated at the beginning of the chapter.

The articles that are submitted for publication are:

- Bayesian sparse factor analysis with kernelized observations. Submitted to Neurocomputing in 2021. Written by Carlos Sevilla Salcedo (main author), Alejandro Guerrero López, Pablo Martínez Olmos and Vanessa Gómez Verdejo. Available at Arxiv. This article is fully included in Chapter 4. The code of the proposed model and exemplary notebooks are openly available at KSSHIBA. The material from this source included in this thesis is not singled out with typographic means and references, although it is indicated at the beginning of the chapter.

Furthermore, I wrote another document to explain in detail some formulations of the state-of-the-art bayesian models. The objective is to develop the formulation of some of the models in detail for a better understanding of the models. This document is available at GitHub.

Abstract

Machine learning (ML) techniques are becoming crucial in the field of health and, in particular, in the analysis of mental diseases. These are usually studied with neuroimaging, which is characterised by a large number of input variables compared to the number of samples available. The main objective of this PhD thesis is to propose different ML techniques to analyse mental diseases from neuroimaging data including different extensions of these models in order to adapt them to the neuroscience scenario. In particular, this thesis focuses on using brainimaging latent representations, since they allow us to endow the problem with a reduced low dimensional representation while obtaining a better insight on the internal relations between the disease and the available data. This way, the main objective of this PhD thesis is to provide interpretable results that are competent with the state-of-the-art in the analysis of mental diseases.

This thesis starts proposing a model based on classic latent representation formulations, which relies on a bagging process to obtain the relevance of each brainimaging voxel, Regularised Bagged Canonical Correlation Analysis (RB-CCA). The learnt relevance is combined with a statistical test to obtain a selection of features. What's more, the proposal obtains a class-wise selection which, in turn, further improves the analysis of the effect of each brain area on the stages of the mental disease. In addition, RB-CCA uses the relevance measure to guide the feature extraction process by using it to penalise the least informative voxels for obtaining the low-dimensional representation. Results obtained on two databases for the characterisation of Alzheimer's disease and Attention Deficit Hyperactivity Disorder show that the model is able to perform as well as or better than the baselines while providing interpretable solutions.

Subsequently, this thesis continues with a second model that uses Bayesian approximations to obtain a latent representation. Specifically, this model focuses on providing different functionalities to build a common representation from different data sources and particularities. For this purpose, the proposed generative model, Sparse Semi-supervised Heterogeneous Interbattery Bayesian Factor Analysis (SSHIBA), can learn the feature relevance to perform feature selection, as well as automatically select the number of latent factors. In addition, it can also model heterogeneous data (real, multi-label and categorical), work with kernels and use a semi-supervised formulation, which naturally imputes missing values by sampling from the learnt distributions. Results using this model demonstrate the versatility of the formulation, which allows these extensions to be combined interchangeably, expanding the scenarios in which the model can be applied and improving the interpretability of the results.

Finally, this thesis includes a comparison of the proposed models on the Alzheimer's disease dataset, where both provide similar results in terms of performance; however, RB-CCA provides a more robust analysis of mental diseases that is more easily interpretable. On the other hand, while RB-CCA is more limited to specific scenarios, the SSHIBA formulation allows a wider variety of data to be combined and is easily adapted to more complex real-life scenarios.

Resumen

Las técnicas de aprendizaje automático (ML) están siendo cruciales en el campo de la salud y, en particular, en el análisis de las enfermedades mentales. Estas se estudian habitualmente con neuroimagen, que se caracteriza por un gran número de variables de entrada en comparación con el número de muestras disponibles. El objetivo principal de esta tesis doctoral es proponer diferentes técnicas de ML para el análisis de enfermedades mentales a partir de datos de neuroimagen incluyendo diferentes extensiones de estos modelos para adaptarlos al escenario de la neurociencia. En particular, esta tesis se centra en el uso de representaciones latentes de imagen cerebral, ya que permiten dotar al problema de una representación reducida de baja dimensión a la vez que obtienen una mejor visión de las relaciones internas entre la enfermedad mental y los datos disponibles. De este modo, el objetivo principal de esta tesis doctoral es proporcionar resultados interpretables y competentes con el estado del arte en el análisis de las enfermedades mentales.

Esta tesis comienza proponiendo un modelo basado en formulaciones clásicas de representación latente, que se apoya en un proceso de bagging para obtener la relevancia de cada voxel de imagen cerebral, el Análisis de Correlación Canónica Regularizada con Bagging (RB-CCA). La relevancia aprendida se combina con un test estadístico para obtener una selección de características. Además, la propuesta obtiene una selección por clases que, a su vez, mejora el análisis del efecto de cada área cerebral en los estadios de la enfermedad mental. Por otro lado, RB-CCA utiliza la medida de relevancia para guiar el proceso de extracción de características, utilizándola para penalizar los vóxeles menos relevantes para obtener la representación de baja dimensión. Los resultados obtenidos en dos bases de datos para la caracterización de la enfermedad de Alzheimer y el Trastorno por Déficit de Atención e Hiperactividad demuestran que el modelo es capaz de rendir igual o mejor que los baselines a la vez que proporciona soluciones interpretables.

Posteriormente, esta tesis continúa con un segundo modelo que utiliza aproximaciones Bayesianas para obtener una representación latente. En concreto, este modelo se centra en proporcionar diferentes funcionalidades para construir una representación común a partir de diferentes fuentes de datos y particularidades. Para ello, el modelo generativo propuesto, Sparse Semi-supervised Heterogeneous Interbattery Bayesian Factor Analysis (SSHIBA), puede aprender la relevancia de las características para realizar la selección de las mismas, así como seleccionar automáticamente el número de factores latentes. Además, también puede modelar datos heterogéneos (reales, multietiqueta y categóricos), trabajar con kernels y utilizar una formulación semisupervisada, que imputa naturalmente los valores perdidos mediante el muestreo de las distribuciones aprendidas. Los resultados obtenidos con este modelo demuestran la versatilidad de la formulación, que permite combinar indistintamente estas extensiones, ampliando los escenarios en los que se puede aplicar el modelo y mejorando la interpretabilidad de los resultados.

Finalmente, esta tesis incluye una comparación de los modelos propuestos en el conjunto de datos de la enfermedad de Alzheimer, donde ambos proporcionan resultados similares en términos de rendimiento; sin embargo, RB-CCA proporciona un análisis más robusto de las enfermedades mentales que es más fácilmente interpretable. Por otro lado, mientras que RB-CCA está más limitado a escenarios específicos, la formulación SSHIBA permite combinar una mayor variedad de datos y se adapta fácilmente a escenarios más complejos de la vida real.

Contents

Chapter 1

Introduction

Machine learning (ML) is a scientific discipline that focuses on analysing data to build models. These models find patterns in the data and learn to make decisions without the need for direct human interaction. In recent years, ML has become a well-known tool with the ability to facilitate different human tasks. In fact, its impact has transcended industries and an increasing number of everyday life applications are within the reach of any user, e.g., deep neural networks of *DeepL* to accurately translate complete and complex texts in real time, the new artificial intelligence of *Photoshop* to automatically fix images, the recommendation algorithm of *Youtube* or the tagging and grouping of images of *Google Photos*.

And the medical field is not far behind. ML algorithms can automatically find patterns in data that can be applied to detect diseases (Watson et al., 2019). A 2005 study reviewed the effect of ML algorithms on the decision-making process and treatment of patients, concluding with an improvement in the decision-making process in 64% of the studies and in the patient's treatment in 13% (Garg et al., 2005). Moreover, in recent years, ML-based algorithms have obtained significant performance improvements in different problems in medicine such as breast tumour classification (Kyono et al., 2020), detection of cardiac disorders (Chang et al., 2021) or analysis of amyotrophic lateral sclerosis (Bereman et al., 2018). At the same time, these advances in the use of ML in medicine are also becoming more accessible to all users and clinicians with systems such as Google health, which provides different ML-based functionalities such as the identification of information related to skin diseases. IBM's Watson Health recently introduced another application for medicine that combines clinical information with different ML techniques to identify patterns that, in turn, create dynamic groups of patients that can improve the clinical interpretation of a disease. Similarly, KENSCI offers an alternative for predicting diseases from patient analysis and identifying health risks, as well as managing health records. Equivalently, some studies are beginning to focus on the development of ML techniques for personalised medicine, which provides tailored treatments for different groups of patients (Zhang et al., 2018; Schork, 2019).

Recently, technological advances in neuroimaging have made it possible to collect enough data to benefit from ML techniques (Poldrack and Gorgolewski, 2014). Such datasets have the particularity of having tens of thousands of variables, corresponding to three-dimensional neuroimaging voxels (equivalent to pixels), while having a reduced number of samples (Smith and

Nichols, 2018; Bzdok et al., 2019). This asymmetry in the dimensionality is mainly caused by the expense of obtaining these neuroimages, which subsequently reduces the number of samples obtained for each dataset (Grant and Chamberlain, 2018). The fact that, in some cases, the number of variables is almost 1000 times the number of samples means that parametric models need to determine considerably more parameters, leading to severe overfitting problems (Friedman et al., 2001; Stephan et al., 2015), and non-parametric models do not have enough samples to fit the problem, leading to underfitting problems (Ghahramani, 2015).

For this reason, models developed for neuroimaging often need to be combined with techniques that alleviate the low sample-to-feature ratio. Some studies have focused on including different regularisation terms to penalise brain areas that are less significant for the analysis (Ye and Wang, 2006; Lange et al., 2020). Other studies use dimensionality reduction techniques to work with compact representations of the original data, such as Principal Component Analysis (PCA) (Hansen et al., 1999; Zhong et al., 2009; Georgieva and De la Torre, 2013), Partial Least Squares (PLS) (Krishnan et al., 2011; Long et al., 2019) or Canonical Correlation Analysis (CCA) (Lee et al., 2020). In addition, another common technique used to work with neuroimaging data is to reduce the original number of voxels by eliminating those that are the least relevant to the problem, such as the Recursive Feature Elimination (RFE) method (Gholami et al., 2012; Yin et al., 2017) or permutation tests (Hemmelmann et al., 2004).

There are also other peculiarities of neuroimaging problems that make the application of ML methods to these problems non-trivial. The complicated procedures required to obtain the brainimaging often mean that in this type of problem there can be a considerable number of missing values (Vaden Jr et al., 2012). This situation is usually solved by different imputation techniques (Mulugeta et al., 2017), which allow retrieving as much information as possible to use for training the model. This problem is also combined with the under-representation of certain categories in the analysis of multiple diagnoses (multiclass), where there tends to be a class imbalance that biases the ML results and makes the prediction of minority diagnoses a challenge (He and Ma, 2013). Another situation commonly linked to neuroimaging scenarios is the need to combine data from different sources and types into a common framework (Ding et al., 2017). To this end, some studies have combined the available fMRI information (based on models and functional connectivity) using kernels (Castro et al., 2011) or by learning a sparse aggregate of the available data using multiple kernel learning (Donini et al., 2019). In particular, Yuan et al. (2012) works simultaneously with heterogeneous data from multiple sources using model ensembles based on majority voting and missing value imputation.

Looking at neuroscience studies, ML methods have two distinct ways of tackling these problems: models centred on **prediction** and models that, additionally, focus on **explanation** (Rosenberg et al., 2018). While one concentrates solely on finding an estimate for a given problem, the other aims to improve the interpretability of the results and, in turn, provide a better understanding of the disease. In the context of ML, most studies focus on finding classification models for automatic diagnosis (Zhang et al., 2017a; Shi and Nathoo, 2018; Yassin et al., 2020; Lanka et al., 2020). However, the real challenge lies in disease characterisation. ML models that focus on feature reduction are helpful tools that provide a good insight into this explanation. In this sense, we can find that the main approaches utilised in the neuroimaging context can be divided into:

- **Feature Selection (FS)**. The main objective of this type of models is to find the relevance associated with each characteristic to, later, eliminate those that do not contribute to the problem. This allows to conserve the original input data and only use the features that are relevant for the problem. In a classification/regression problem, this is usually done either before the classifier/regressor (filters), after the classifier/regressor is trained (wrapper) or simultaneously as a regularisation term (embedded) (Michel et al., 2011; Cheng et al., 2017). FS has been widely analysed in neuroimaging problems due to its ability to reduce the input dimension for the classifier/regressor and, subsequently, improve the interpretability of the results (Tohka et al., 2016).
- **Feature Extraction (FE)**. These models use the available information to construct a lower dimension representation of the data (Klöppel et al., 2008; Risacher et al., 2010; Hinrichs et al., 2011). Specifically, these techniques are also divided into unsupervised, models that only project the neuroimaging into a low dimensional space (e.g. PCA), and supervised, which combine the neuroimaging data and the output data (diagnosis, biomarkers,...) to define a shared projection space (e.g. CCA, PLS).

There is a wide range of examples of FS techniques used in neuroimaging problems (Remeseiro and Bolon-Canedo, 2019; Xu et al., 2020; Hao et al., 2020) as they not only reduce the dimension of the data but also provide a greater understanding of the brain. Among the FS techniques used in neuroimaging problems, some of the most widely used are permutation tests, which study the sign consistency of learnt weights (Nichols and Holmes, 2002; Gaonkar and Davatzikos, 2013; Rondina et al., 2013; Abdulkadir et al., 2014) based on a score to rank features (Inza et al., 2004; Eloyan et al., 2014; Knorr et al., 2020) and RFE, which iteratively eliminates features and provides a ranking based on their relevance (Hanson and Halchenko, 2008; Zhang et al., 2017b; Lai et al., 2017; Wottschel et al., 2019). However, although these approaches provide good insight into the basic relationships, the fact that they are external to the classification/regression model makes them less robust to the task (Guyon and Elisseeff, 2003). To get around this, some studies proposed embedded FS that simultaneously select relevant features and train a classifier using a cost function that penalises less relevant features, e.g., using a regularisation term (Grosenick et al., 2013; Tohka et al., 2016; Song and Lu, 2017).

Classification studies with FS in neuroimaging problems predominantly focus on binary classification, while multiclass frameworks are scarce. Most of the algorithms used in multiclass neuroimaging classification can not be adapted for a different scenario, as they are ad-hoc for the specific situation analysed and do not exploit the multiclass information (Yu et al., 2013; Bron et al., 2015). In Qureshi et al. (2016) the authors propose to combine models that learn to classify only one class using a one-vs.-all framework, losing the internal relationships inherent in the simultaneous classification of all classes. Moreover, to the best of our knowledge, FS is not combined with any other type of dimensionality reduction technique in neuroscience, with the exception of regularised multinomial logistic regression (Huttunen et al., 2013).

These approaches can also be combined or replaced by a FE model which also provides a dimensionality reduction, through a projection of the information into a low-dimensional space, as well as an explanation of the relationships between the data and that latent representation. This technique reduces the dimensionality of the data by transforming it into a latent space, removing noisy components and feature correlations (Suk and Lee, 2012). Among them, PCA is the most widespread technique in neuroscience, as it obtains a direct low-dimensional representation of brainimaging data. Specifically, different areas of neuroscience have been shown to benefit from the inclusion of PCA in the characterisation of diseases, such as schizophrenia, (Pan et al., 2020; Sartipi et al., 2020), Alzheimer's Disease (Pagani et al., 2009; López et al., 2011; Ahmad and Dar, 2018) or psychosis (Paolini et al., 2016; Tibber et al., 2018). Conversely, the CCA stands out for its ability to define the latent space with the correlation between all available data. Although it is mainly used to find the correlation between two views (input features and output labels (Kursun et al., 2011; Wang et al., 2020)), the usage trend for multi-view or multi-task problems is increasing recently (Kamronn et al., 2015; Li et al., 2018; Tan et al., 2019).

Another popular adaptation of FE methods is the use of Bayesian formulations to include probabilistic distributions in the definition of the latent space (Zhang et al., 2016; Chen et al., 2019). This approach, also known as Factor Analysis (FA), defines the distribution associated with each variable in the model, and introduces prior distributions. These prior distributions can improve the FA techniques by using external information from the researchers regarding their expectations about the problem (Stefan et al., 2019). Furthermore, the use of the probabilistic formulation also yields a posterior distribution on each variable in the model, instead of a deterministic result, as well as a predictive distribution. These not only provide an estimate of a value, but also a measure of certainty of the result. In particular, Bayesian formulations have been used with both PCA (Woolrich et al., 2011; Oswal et al., 2014; Hinrich et al., 2016) and CCA (Fujiwara et al., 2009; Zhuang et al., 2020) on neuroimaging problems to enhance their performance. In particular, Bayesian formulations introduce different constraints to the model by declaring specific priors on the model variables. Namely, previous works use the Bayesian formulation to induce sparsity in the latent space by means of an Automatic Relevance Determination (ARD) prior and, subsequently, to eliminate the factors that are irrelevant (Connor et al., 2015; Yu et al., 2020). While the use of a priori information is a powerful tool in neuroimaging problems, the novelty of the study, the need for a deeper explanation of brain function or the lack of expert information often means that it can not be used or is combined with other dimensionality reduction techniques. Another interesting FA model is Bayesian Interbattery Factor Analysis (BIBFA) presented by Klami et al. (2013), a Bayesian extension of CCA that combines automatic latent feature determination with multivariate learning.

On the other hand, the inclusion of a probabilistic formulation also facilitates the appropriate modelling of heterogeneous (not only continuous) data, such as labelled or ordinal data. Although most FA methods model real data, some authors have found promising results in modelling categorical data (Terzi and Cengiz, 2013; Pauger et al., 2019) and multi-labelled data (Gönen, 2012; Zhang et al., 2014) including the specific particularities of these data types. Furthermore, Bayesian formulations instinctively work with missing values. These models can be naturally extended to obtain semi-supervised formulations to train the model with available data and impute unavailable data (Münch et al., 2021). Besides, some methods combine this functionality with other Bayesian properties to build more versatile models, such as Toutanova and Johnson (2008), which combines it with sparsity for word labelling or Lian et al. (2015) where the author jointly models categorical and multi-label data.

Among the models based on Bayesian formulations, the Gaussian Process (GP) is one of the most frequently used for its ability to provide a measure of uncertainty in its estimation, working for both classification and regression problems and using kernels to work with high-dimensional data (Lukic et al., 2007). In recent years, several studies have started to use its extension for dimensionality reduction, GP Latent Variable Model (GPLVM), which also allows modelling the data using kernels (Gunawardena et al., 2020; Bahg et al., 2020; Wu et al., 2021). Damianou et al. (2012) propose an approach on GPLVMs that combines multi-view learning with non-linear representation. This was extended in Damianou et al. (2016) to include a feature relevance determination on the kernel.

1.1 Motivation

Despite all that is available, we see that the application of ML in neuroimaging requires specific models that cover the particularities of neuroimaging characterisation problems:

- Having **high dimensionality** where the number of features is considerably larger than the number of samples.
- Needing **interpretable** results to understand the nature of mental diseases.
- Having a large number of **missing values**.
- Dealing with **multi-source heterogeneity** in the available data.

However, the available solutions in the literature only cover these requirements partially. For this reason, here, we will focus on FE models with FS capabilities able to incorporate the stated needs for the characterisation of mental diseases.

1.2 Objectives

With these problems in mind, this thesis proposes new ML tools for the state-of-the-art with the capacity to:

1. Have **interpretable** results that provide a better insight on the analysed problem.
2. Obtain **versatile** formulations that can adapt to different scenarios.
3. Provide **useful** tools for the characterisation of mental diseases.

Specifically, we draw on two existing models initially designed for other applications and adapt them to the particularities and needs of neuroimaging problems, finally proposing two distinct models. The first one is based on classical multivariate analysis (MVA) methods for FE. It focuses on a bagging procedure to obtain the relevance of the input features. Thereafter, the aim of the model is twofold: (1) to use the relevance of the learnt features to perform the FS and (2) to combine the relevance as a regularisation term to perform a guided FE. Furthermore, through the bagging process, the model is also able to determine which features are relevant for each class analysed, providing a guided FE per class. This can be achieved by means of a statistical test capable of automatically selecting the most relevant features, avoiding the computational cost of the CV of the number of selected features. By including another adaptation to the regularisation, we also endow the model with balancing the least populated classes. In this way, we expect to have a model that provides a detailed explanation of brain relationships that allow for interpretable results as well as competent classification performance.

The second proposed approach is based on a Bayesian formulation for FA. By using the Bayesian formulation, we are able to include prior information to not only automatically select latent factors, but also to select relevant features. Another advantage of working with the Bayesian formulation is that it allows us to include a specific probability distribution to model multi-label and categorical data. Furthermore, the presented model has the ability to work in a semi-supervised way, being able to automatically impute any missing values in the data. All this can be used in a multi-view framework, where we can combine different types of data in different views and with or without each of the above functionalities. Nonetheless, we also propose an extension of the model to work in dual space and include nonlinearities based on kernel learning. This is of particular interest in neuroimaging problems with high-dimensional data. In particular, we adapt the kernel formulation to automatically remove the least relevant vectors from the kernel while maintaining the available feature relevance determination. Finally, we can combine different kernels in different views to have a multiple kernel learning model that intrinsically learns kernel weights.

1.3 Organisation

The remainder of this thesis is organised as follows. Chapter 2 presents a review of different FE methods. The aim of this chapter is to lay the foundations of the proposed models by introducing the formulation of the models on which they are based. Chapter 3 presents the novel Regularised Bagged Canonical Correlation Analysis (RB-CCA) based on the classical CCA formulation and includes results obtained on two different neuroimaging databases. The model and the results presented in it are published in Sevilla-Salcedo et al. (2020a). Next, Chapter 4 includes the recent Sparse Semi-supervised Heterogeneous Interbattery Bayesian Analysis (SSHIBA) model together to its kernelised extension, as well as the results obtained with the different functionalities of the model on various datasets and in the context of neuroimaging datasets. This model is published in Sevilla-Salcedo et al. (2021) and Sevilla-Salcedo et al. (2020b). Finally, Chapter 5 presents some concluding remarks on the results presented, as well as a comparison of the two models.

Chapter 2

A review of feature extraction methods

2.1 Multivariate Analysis

Multivariate Analysis (MVA) is an interesting tool when working with most machine learning problems. In particular, high dimensional problems benefit from using it, since MVA is able to exploit the correlations between the input variables and reduce their redundancies by projecting the data into a low dimensional space.

Along this PhD thesis, we are going to work with multisource data problems, i.e., we will deal with different data representations, provided by different data sources, which are considered as data views. Let's consider $\mathbf{X}^{(\mathrm{m})} \in \mathbb{R}^{\mathrm{N} \times \mathrm{D_m}}$, as the m-th data view with $\mathrm{D_m}$ features and N samples. Note that in a classification problem the labels can be treated as an additional view codified in a one-vs.-all fashion, where $\mathrm{D_m}$ corresponds to the number of classes. For each data view $\mathbf{X}^{(\mathrm{m})}$, let us represent the n-th data (row) as $\mathbf{x}^{(\mathrm{m})}_{\mathrm{n},:}$, the d-th feature (column) as $\mathbf{x}^{(\mathrm{m})}_{:,\mathrm{d}}$ and, therefore, $\mathrm{x}^{(\mathrm{m})}_{\mathrm{n,d}}$ is the d-th feature of the n-th sample.

In this context, the general idea of MVA algorithms is to find a projection matrix $\mathbf{W}^{(\mathrm{m})} \in \mathbb{R}^{\mathrm{D_m} \times \mathrm{K_c}}$ that transforms our data source $\mathbf{X}^{(\mathrm{m})}$ into a latent representation $\mathbf{Z} = \mathbf{X}^{(\mathrm{m})} \mathbf{W}^{(\mathrm{m})}$ of size $\mathrm{N} \times \mathrm{K_c}$, where $\mathrm{K_c} \leq \min(\mathrm{N}, \mathrm{D_1}, \ldots, \mathrm{D_m})$ is the number of latent features of the new latent representation. To obtain this projection matrix, different MVA approaches will aim to exploit inter-view or intra-view correlations. In the next sections we will describe Principal Component Analysis (PCA), which exploits the intra-view data correlations, and Canonical Correlation Analysis (CCA), which searches for inter-view correlations.

2.1.1 Principal Component Analysis

PCA (Pearson, 1901) obtains a low dimensional data representation by projecting the data over the directions of maximum variance. For this purpose, PCA only analyses the intra-view correlations, so its formulation deals with a single data view. So, for the sake of simplicity, let us denote the model data as $\mathbf{X} \in \mathbb{R}^{\mathrm{N} \times \mathrm{D}}$ and the projection matrix as $\mathbf{W}$. Then, the PCA goal relies in finding a projection matrix, $\mathbf{W}$, which is able to map the input data, $\mathbf{X}$, onto a lower dimensional space while maximising the variance of the projected data. In this context, we have to consider that the projection of $\mathbf{X}$ into the low dimensional space is computed as $\mathbf{Z} = \mathbf{X}\,\mathbf{W}$, then defining $\mathbf{C_{XX}} = \mathbf{X}^{\mathrm{T}}\,\mathbf{X}$ as the covariance matrix of the data view $\mathbf{X}$, the projection matrix $\mathbf{W}$ can be obtained as the solution of the following maximisation problem

$$\begin{aligned} \max_{\mathbf{W}} \quad & \mathbf{W}^{\mathrm{T}}\,\mathbf{C_{XX}}\,\mathbf{W}, \\ \text{s.t.} \quad & \mathbf{W}^{\mathrm{T}}\,\mathbf{W} = \mathbf{I}_{\mathrm{K_c}} \end{aligned} \tag{2.1}$$

which can be reformulated, using Lagrange multipliers, as

$$\max_{\mathbf{W}} \quad \mathbf{W}^{\mathrm{T}}\,\mathbf{C_{XX}}\,\mathbf{W} - \boldsymbol{\Lambda}\big(\mathbf{W}^{\mathrm{T}}\,\mathbf{W} - \mathbf{I}_{\mathrm{K_c}}\big), \tag{2.2}$$

where $\boldsymbol{\Lambda}$ represents a diagonal matrix with the Lagrange multipliers.

We can now determine the value of $\mathbf{W}$ by deriving (2.2) with respect to $\mathbf{W}$ and equating to 0, showing

$$\mathbf{C_{XX}}\,\mathbf{W} \;=\; \mathbf{W}\,\boldsymbol{\Lambda}, \tag{2.3}$$

which implies that the optimum solution of $\mathbf{W}$ corresponds to the eigenvectors of $\mathbf{C_{XX}}$ and $\boldsymbol{\Lambda}$ is a diagonal matrix with their associated eigenvalues. Given that the eigenvectors are ordered by the eigenvalues from higher to lower ($\lambda_1 > \lambda_2 > \ldots > \lambda_{\mathrm{K_c}}$), we know that the first eigenvector provides the direction of maximum variance and is known as the first principal component. Subsequently, every consecutive eigenvector provides the following direction of maximum variance, orthonormal to the previous eigenvectors.

It is well-known that, the eigenvector problem of Equation (2.3) can also be solved with the Single Value Decomposition (SVD) of $\mathbf{C_{XX}}$. That is, given that the SVD of $\mathbf{C_{XX}}$ is

$$\mathbf{C_{XX}} \;=\; \mathbf{L_{C_{XX}}}\,\boldsymbol{\Lambda}_{\mathbf{C_{XX}}}\,\mathbf{V}^{\mathrm{T}}_{\mathbf{C_{XX}}}, \tag{2.4}$$

$\mathbf{L_{C_{XX}}}$ being a $\mathrm{N}\times\mathrm{N}$ matrix with the left singular vectors which, due to the symmetry of $\mathbf{C_{XX}}$, is the same as the right singular vectors $\mathbf{V_{C_{XX}}}$, and $\boldsymbol{\Lambda}_{\mathbf{C_{XX}}}$ is a $\mathrm{D}\times\mathrm{D}$ diagonal matrix with their associated eigenvalues in descending order, $\lambda_1 > \lambda_2 > \ldots > \lambda_{\mathrm{D}}$. Consequently, using only the first $\mathrm{K_c}$ columns of $\mathbf{V_{C_{XX}}}$, we can obtain the projection matrix $\mathbf{W} \in \mathbb{R}^{\mathrm{D}\times\mathrm{K_c}}$ with the first $\mathrm{K_c}$ eigenvectors of $\mathbf{C_{XX}}$. On another note, we can also obtain the PCA solution by means of the SVD of $\mathbf{X}$. In this case, if

$$\mathbf{X} \;=\; \mathbf{L_X}\,\boldsymbol{\Lambda}_{\mathbf{X}}\,\mathbf{V}^{\mathrm{T}}_{\mathbf{X}}, \tag{2.5}$$

where $\mathbf{L_X}$ is a $\mathrm{N}\times\mathrm{N}$ matrix with the left singular vectors, $\mathbf{V_X}$ is a $\mathrm{D}\times\mathrm{D}$ matrix with the right singular vectors and $\boldsymbol{\Lambda}^{\mathrm{T}}_{\mathbf{X}}\,\boldsymbol{\Lambda}_{\mathbf{X}} = \boldsymbol{\Lambda}_{\mathbf{C_{XX}}}$ is a $\mathrm{N}\times\mathrm{D}$ diagonal matrix in its upper ($\mathrm{D} < \mathrm{N}$) or left ($\mathrm{D} > \mathrm{N}$) part and zeros in the rest of the matrix, having in its diagonal the singular values in descending order, $\lambda_1 > \lambda_2 > \cdots > \lambda_{\mathrm{K_c}}$. If we now calculate $\mathbf{C_{XX}}$ we get

$$\mathbf{C_{XX}} \;=\; \mathbf{X}^{\mathrm{T}}\,\mathbf{X} = \mathbf{V_X}\,\boldsymbol{\Lambda}^{\mathrm{T}}_{\mathbf{X}}\,\mathbf{L}^{\mathrm{T}}_{\mathbf{X}}\,\mathbf{L_X}\,\boldsymbol{\Lambda}_{\mathbf{X}}\,\mathbf{V}^{\mathrm{T}}_{\mathbf{X}} = \mathbf{V_X}\,\boldsymbol{\Lambda}^{\mathrm{T}}_{\mathbf{X}}\boldsymbol{\Lambda}_{\mathbf{X}}\,\mathbf{V}^{\mathrm{T}}_{\mathbf{X}}, \tag{2.6}$$

where $\mathbf{L}^{\mathrm{T}}_{\mathbf{X}}\,\mathbf{L_X} = \mathbf{I}$. This way, by comparing Equations (2.4) and (2.6), the projection matrix $\mathbf{W}$ also corresponds to the first $\mathrm{K_c}$ columns of the right singular vectors, $\mathbf{V_X}$ of $\mathbf{X}$.

Furthermore, the PCA formulation can also be seen as a least square error optimisation problem where the reconstruction error is minimised, i.e.,

$$\begin{aligned} \min_{\mathbf{W},\mathbf{U}} \quad & \big\|\mathbf{X} - \mathbf{X}\,\mathbf{W}\,\mathbf{U}^{\mathrm{T}}\big\|^2_{\mathrm{F}}, \\ \text{s.t.} \quad & \mathbf{U}^{\mathrm{T}}\,\mathbf{U} = \mathbf{I}_{\mathrm{K_c}} \end{aligned} \tag{2.7}$$

where $\|\cdot\|_2^F$ is the Frobenius norm operator and $\mathbf{U}$ is a $\mathrm{D} \times \mathrm{K_c}$ reconstruction matrix. Note that we made use of the definition of the projected data as $\mathbf{Z} = \mathbf{X}\,\mathbf{W}$ to determine the reconstructed data as $\mathbf{Z}\,\mathbf{U}^{\mathrm{T}}$. If we develop the Frobenius norm, we get the following equivalent optimisation problem

$$\begin{aligned} \min_{\mathbf{W},\mathbf{U}} \quad & \mathrm{Tr}\{\mathbf{W}^{\mathrm{T}}\,\mathbf{C_{XX}}\,\mathbf{W}\} - 2\mathrm{Tr}\{\mathbf{C_{XX}}\,\mathbf{W}\,\mathbf{U}^{\mathrm{T}}\}, \\ \text{s.t.} \quad & \mathbf{U}^{\mathrm{T}}\,\mathbf{U} = \mathbf{I}_{\mathrm{K_c}}\,. \end{aligned} \tag{2.8}$$

Now, if we calculate the partial derivative of the objective function in (2.8) with respect to $\mathbf{W}$, we get

$$2\,\mathbf{C_{XX}}\,\mathbf{W} - 2\,\mathbf{C_{XX}}\,\mathbf{U} \;=\; \mathbf{0}, \tag{2.9}$$

which implies that $\mathbf{U} = \mathbf{W}$, i.e., the projection matrix also corresponds to the reconstruction matrix. Considering now that $\mathbf{U} = \mathbf{W}$ and making use of the lagrange multipliers to include the constraint in the objective function of (2.7), we arrive to

$$\min_{\mathbf{W}} \quad \|\mathbf{X} - \mathbf{X}\,\mathbf{W}\,\mathbf{W}^{\mathrm{T}}\|_{\mathrm{F}}^2 - \|\mathbf{\Lambda}(\mathbf{W}^{\mathrm{T}}\,\mathbf{W} - \mathbf{I}_{\mathrm{K_c}})\|_{\mathrm{F}}^2\,. \tag{2.10}$$

Again, developing the Frobenius norm, we get that

$$\begin{aligned} \min_{\mathbf{W}} \quad & \mathrm{Tr}\{\mathbf{W}^{\mathrm{T}}\,\mathbf{C_{XX}}\,\mathbf{W}\} - 2\mathrm{Tr}\{\mathbf{C_{XX}}\,\mathbf{W}\,\mathbf{W}^{\mathrm{T}}\} \\ & - \mathrm{Tr}\{\mathbf{\Lambda}(\mathbf{W}^{\mathrm{T}}\,\mathbf{W}\,\mathbf{W}^{\mathrm{T}}\,\mathbf{W} + \mathbf{I}_{\mathrm{K_c}} - 2\,\mathbf{W}\,\mathbf{W}^{\mathrm{T}})\}. \end{aligned} \tag{2.11}$$

Operating with trace properties we can simplify this problem as

$$\max_{\mathbf{W}} \quad \mathrm{Tr}\{\mathbf{C_{XX}}\,\mathbf{W}\,\mathbf{W}^{\mathrm{T}}\} - \mathrm{Tr}\{\mathbf{\Lambda}(\mathbf{W}^{\mathrm{T}}\,\mathbf{W} - \mathbf{I}_{\mathrm{K_c}})\}. \tag{2.12}$$

and, if we calculate the partial derivative with respect to $\mathbf{W}$, we finally obtain

$$\mathbf{C_{XX}}\,\mathbf{W} \;=\; \mathbf{W}\,\mathbf{\Lambda} \tag{2.13}$$

which corresponds to Equation (2.3), proving that the error reconstruction minimisation problem is equivalent to maximising the variance of the projected data and, thus, leads to the same solutions where $\mathbf{W}$ is given by the eigenvectors of $\mathbf{C_{XX}}$.

2.1.2 Canonical Correlation Analysis

The other relevant MVA algorithm we will present in this section is the Canonical Correlation Analysis (CCA) (Hotelling, 1992). Unlike PCA, the objective of CCA is to maximise the inter-view correlation, so in this case we will work with two data views. To do so, CCA aims to find a lower dimensional projection space common to both views such that the projected views, i.e. $\mathbf{Z}^{(1)} = \mathbf{X}^{(1)}\,\mathbf{W}^{(1)}$ and $\mathbf{Z}^{(2)} = \mathbf{X}^{(2)}\,\mathbf{W}^{(2)}$, are maximally correlated. Accordingly, to find the projection vectors, CCA needs to maximise the correlation between the two views as

$$\max_{\mathbf{w}_{:,k}^{(1)},\mathbf{w}_{:,k}^{(2)}} \quad \frac{\mathbf{w}_{:,k}^{(1)\mathrm{T}}\,\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}}\,\mathbf{w}_{:,k}^{(2)}}{\sqrt{\mathbf{w}_{:,k}^{(1)\mathrm{T}}\,\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\,\mathbf{w}_{:,k}^{(1)}\,\mathbf{w}_{:,k}^{(2)\mathrm{T}}\,\mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}\,\mathbf{w}_{:,k}^{(2)}}}, \tag{2.14}$$

where $\mathbf{w}^{(1)}_{:,k}$ and $\mathbf{w}^{(2)}_{:,k}$ correspond to the k-th projection vectors of views 1 and 2, $\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}} = \mathbf{X}^{(1)^{\mathrm{T}}}\mathbf{X}^{(2)}$ is the cross-covariance and $\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}} = \mathbf{X}^{(1)^{\mathrm{T}}}\mathbf{X}^{(1)}$ and $\mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}} = \mathbf{X}^{(2)^{\mathrm{T}}}\mathbf{X}^{(2)}$ are the covariance matrices of the first and second data views, respectively. As maximising this correlation is invariant to any scaling factor, we can rewrite (2.14) adding some constraints and reformulating it to work with matrices as

$$\begin{aligned} \max_{\mathbf{W}^{(1)},\mathbf{W}^{(2)}} \quad & \mathrm{Tr}\Big\{\mathbf{W}^{(1)^{\mathrm{T}}}\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}}\mathbf{W}^{(2)}\Big\}, \\ \text{s.t.} \quad & \mathbf{W}^{(1)^{\mathrm{T}}}\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\mathbf{W}^{(1)} = \mathbf{W}^{(2)^{\mathrm{T}}}\mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}\mathbf{W}^{(2)} = \mathbf{I}_{\mathrm{K_c}}\,. \end{aligned} \tag{2.15}$$

As happened with the PCA formulation, we can solve this problem using the Lagrange multipliers, reaching, in this case, a generalised eigenvector problem

$$\begin{bmatrix} \mathbf{0} & \mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}} \\ \mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(1)}} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{W}^{(1)} \\ \mathbf{W}^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}} \end{bmatrix}\begin{bmatrix} \mathbf{W}^{(1)} \\ \mathbf{W}^{(2)} \end{bmatrix}\mathbf{\Lambda}, \tag{2.16}$$

which can be rewritten as the following standard eigenvector problem

$$\begin{aligned} & \begin{bmatrix} \mathbf{C}^{-1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^{-1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}} \end{bmatrix}\begin{bmatrix} \mathbf{0} & \mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}} \\ \mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(1)}} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{C}^{-1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^{-1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}} \end{bmatrix}\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \\ & = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}\mathbf{\Lambda}, \end{aligned} \tag{2.17}$$

where $\mathbf{A} = \mathbf{C}^{1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\mathbf{W}^{(1)}$ and $\mathbf{B} = \mathbf{C}^{1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}\mathbf{W}^{(2)}$. We define that the square root of a matrix, $\mathbf{X}^{1/2}$, corresponds to a matrix such that $\mathbf{X}^{1/2}\mathbf{X}^{1/2} = \mathbf{X}$. This eigenvector problem can be reformulated as a SVD problem where the left singular vectors of $\mathbf{C}^{-1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}}\mathbf{C}^{-1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}$ correspond to $\mathbf{A}$ while, conversely, the left singular vectors of $\mathbf{C}^{-1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}\mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(1)}}\mathbf{C}^{-1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}$ correspond to $\mathbf{B}$.

However, if we define $\mathbf{Y} = \mathbf{C}^{-1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}}\mathbf{C}^{-1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}$, we can jointly determine $\mathbf{A}$ and $\mathbf{B}$ as the left and right singular vectors of the SVD of $\mathbf{Y}$, only needing to calculate one SVD. That is, if the SVD of $\mathbf{Y}$ is

$$\mathbf{Y} = \mathbf{L}_{\mathbf{Y}}\mathbf{\Lambda}_{\mathbf{Y}}\mathbf{V}^{\mathrm{T}}_{\mathbf{Y}} \tag{2.18}$$

we have

$$\mathbf{A} = \mathbf{L}_{\mathbf{Y}} \tag{2.19}$$

$$\mathbf{B} = \mathbf{V}_{\mathbf{Y}}\,. \tag{2.20}$$

Furthermore, if we substitute the definitions of $\mathbf{A}$ and $\mathbf{B}$ previously stated into these equations and isolate the projection matrices, we obtain

$$\mathbf{W}^{(1)} = \mathbf{C}^{-1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\mathbf{L}_{\mathbf{Y}} \tag{2.21}$$

$$\mathbf{W}^{(2)} = \mathbf{C}^{-1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}\mathbf{V}_{\mathbf{Y}}\,. \tag{2.22}$$

The intuition behind this formulation is equivalent to PCA, so we can also redefine it as a least square error problem between the projected data views:

$$\begin{aligned} \min_{\mathbf{W}^{(1)},\mathbf{W}^{(2)}} \quad & \Big\|\mathbf{X}^{(1)}\mathbf{W}^{(1)} - \mathbf{X}^{(2)}\mathbf{W}^{(2)}\Big\|^2_{\mathrm{F}}, \\ \text{s.t.} \quad & \mathbf{W}^{(1)^{\mathrm{T}}}\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\mathbf{W}^{(1)} = \mathbf{W}^{(2)^{\mathrm{T}}}\mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}\mathbf{W}^{(2)} = \mathbf{I}_{\mathrm{K_c}}\,. \end{aligned} \tag{2.23}$$

If we develop the Frobenius norm operator we get that

$$\begin{aligned} \max_{\mathbf{W}^{(1)},\mathbf{W}^{(2)}} \quad & \mathrm{Tr}\left\{\mathbf{W}^{(1)^{\mathrm{T}}}\,\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}}\,\mathbf{W}^{(2)}\right\}, \\ \text{s.t.} \quad & \mathbf{W}^{(1)^{\mathrm{T}}}\,\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\,\mathbf{W}^{(1)} = \mathbf{W}^{(2)^{\mathrm{T}}}\,\mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}\,\mathbf{W}^{(2)} = \mathbf{I}_{\mathrm{K_c}}\,. \end{aligned} \tag{2.24}$$

As can be seen, this optimisation problem corresponds to the correlation maximisation problem between the projected data from each view defined in Equation (2.15). This proves that maximising the intra-view covariance is equivalent to minimising the square error between the two projected data views.

2.1.3 A common Multivariate Analysis formulation

This section reviews the generalised MVA formulation presented in Muñoz-Romero et al. (2017, 2016), which unifies the previous MVA methods: PCA and CCA, into a single framework.

Given two data views, the formulation here presented is only feasible when one of the data views, e.g. $\mathbf{X}^{(2)}$, corresponds to one-vs.-all categorical data codification since, in this case, the covariance matrix of this view, $\mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}$, is diagonal. This framework aims to find two projection matrices: where $\mathbf{W}^{(1)} \in \mathbb{R}^{\mathrm{D_1} \times \mathrm{K_c}}$ maps the input data view $\mathbf{X}^{(1)}$ onto a lower dimensional space with $\mathrm{K_c}$ features and $\mathbf{W}^{(2)} \in \mathbb{R}^{\mathrm{D_2} \times \mathrm{K_c}}$ maps this transformed data into the space of $\mathbf{X}^{(2)}$, minimising the Mean Square Error (MSE) between $\mathbf{X}^{(2)}$ and the mapped data. Hence, the framework can be formulated as the following MSE minimisation problem:

$$\begin{aligned} \min_{\mathbf{W}^{(1)},\mathbf{W}^{(2)}} \quad & \left\|(\mathbf{X}^{(2)} - \mathbf{X}^{(1)}\,\mathbf{W}^{(1)}\,\mathbf{W}^{(2)^{\mathrm{T}}})\mathbf{\Gamma}^{1/2}\right\|_{\mathrm{F}}^{2}, \\ \text{s.t.} \quad & \mathbf{W}^{(1)^{\mathrm{T}}}\,\mathbf{X}^{(1)^{\mathrm{T}}}\,\mathbf{X}^{(1)}\,\mathbf{W}^{(1)} = \mathbf{I}_{\mathrm{K_c}}, \end{aligned} \tag{2.25}$$

where $\mathbf{W}^{(2)}$ is a $\mathrm{D_2} \times \mathrm{K_c}$ regression matrix and $\mathbf{\Gamma}$ is an auxiliary matrix that will allow us to recover specific MVA formulations ((2.7) and (2.23)) as we will define later. Furthermore, the constraint imposes that, while projecting the data $\mathbf{X}^{(1)}$, the data is whitened.

As Muñoz-Romero et al. (2015) states, the constraint in (2.25) can be replaced by one over $\mathbf{W}^{(2)}$, obtaining an equivalent optimisation problem

$$\begin{aligned} \min_{\mathbf{W}^{(1)},\mathbf{W}^{(2)}} \quad & \left\|(\mathbf{X}^{(2)} - \mathbf{X}^{(1)}\,\mathbf{W}^{(1)}\,\mathbf{W}^{(2)^{\mathrm{T}}})\mathbf{\Gamma}^{1/2}\right\|_{\mathrm{F}}^{2}, \\ \text{s.t.} \quad & \mathbf{W}^{(2)^{\mathrm{T}}}\,\mathbf{\Gamma}\,\mathbf{W}^{(2)} = \mathbf{I}_{\mathrm{K_c}}\,. \end{aligned} \tag{2.26}$$

If we develop the Frobenius norm and leave only the terms that depend on the parameters to be minimised, we get the equivalent optimisation problem

$$\begin{aligned} \min_{\mathbf{W}^{(1)},\mathbf{W}^{(2)}} \quad & \mathrm{Tr}\left\{\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\,\mathbf{W}^{(1)^{\mathrm{T}}}\,\mathbf{W}^{(1)}\right\} - 2\mathrm{Tr}\left\{\mathbf{\Gamma}^{1/2}\,\mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(1)}}\,\mathbf{W}^{(1)}\,\mathbf{W}^{(2)^{\mathrm{T}}}\,\mathbf{\Gamma}^{1/2}\right\}, \\ \text{s.t.} \quad & \mathbf{W}^{(2)^{\mathrm{T}}}\,\mathbf{\Gamma}\,\mathbf{W}^{(2)} = \mathbf{I}_{\mathrm{K_c}}, \end{aligned} \tag{2.27}$$

which can be derived with respect to $\mathbf{W}^{(1)}$ and equated to 0 to obtain the following relation between $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$:

$$\mathbf{W}^{(1)} \;=\; \mathbf{C}^{-1}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\,\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}}\,\mathbf{\Gamma}\,\mathbf{W}^{(2)}\,. \tag{2.28}$$

This way, we can substitute this result into Equation (2.26) to reach the following eigenvalue problem

$$\mathbf{\Gamma}^{1/2}\,\mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(1)}}\,\mathbf{C}^{-1}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\,\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}}\,\mathbf{\Gamma}^{1/2}\,\mathbf{W}^{(2)^*} \;=\; \mathbf{W}^{(2)^*}\,\mathbf{\Lambda} \tag{2.29}$$

where $\mathbf{W}^{(2)^*} = \mathbf{\Gamma}^{1/2}\,\mathbf{W}^{(2)}$ is introduced to simplify the notation and whose solution allows us to recover $\mathbf{W}^{(2)}$. Here we can see that when matrix $\mathbf{\Gamma} = \mathbf{I}$ and $\mathbf{X}^{(2)} = \mathbf{X}^{(1)}$, we obtain the PCA formulation (see (2.3)). To obtain the formulation of CCA we have to consider $\mathbf{\Gamma} = \mathbf{C}^{-1}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}$, to reach

$$\mathbf{C}^{-1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}\,\mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(1)}}\,\mathbf{C}^{-1}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\,\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}}\,\mathbf{C}^{-1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}\,\mathbf{W}^{(2)^*} \;=\; \mathbf{W}^{(2)^*}\,\mathbf{\Lambda} \tag{2.30}$$

which, using the previous definition $\mathbf{Y} = \mathbf{C}^{-1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\,\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}}\,\mathbf{C}^{-1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}$, is equivalent to

$$\mathbf{C}_{\mathbf{YY}}\,\mathbf{W}^{(2)^*} \;=\; \mathbf{W}^{(2)^*}\,\mathbf{\Lambda} \tag{2.31}$$

so

$$\mathbf{W}^{(2)^*} \;=\; \mathbf{V}_{\mathbf{Y}} \tag{2.32}$$

where $\mathbf{V}_{\mathbf{Y}}$ corresponds to the right singular vectors of $\mathbf{Y}$ such that $\mathbf{Y} = \mathbf{L}_{\mathbf{Y}}\,\mathbf{\Lambda}_{\mathbf{Y}}\,\mathbf{V}^{\mathrm{T}}_{\mathbf{Y}}$. This way, we have

$$\mathbf{W}^{(2)} \;=\; \mathbf{C}^{1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}\,\mathbf{V}_{\mathbf{Y}} \tag{2.33}$$

which, for a classification problem, conserves the projection direction obtained in (2.22) re-scaled by the correlation matrix. Equivalently, we can calculate the projection matrix $\mathbf{W}^{(1)}$ substituting $\mathbf{\Gamma} = \mathbf{C}^{-1}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}$ in (2.28), that is

$$\mathbf{W}^{(1)} \;=\; \mathbf{C}^{-1}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\,\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}}\,\mathbf{C}^{-1}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}\,\mathbf{W}^{(2)}\,. \tag{2.34}$$

At this point, we can substitute (2.33) in (2.34) to eliminate the dependence on $\mathbf{W}^{(2)}$. Next, we can include the definition of matrix $\mathbf{Y}$, contained in this equation, and substitute it by its SVD, getting

$$\mathbf{W}^{(1)} \;=\; \mathbf{C}^{-1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}\,\mathbf{L}_{\mathbf{Y}}\,\mathbf{\Lambda}, \tag{2.35}$$

which corresponds to (2.21) re-scaled by the eigenvalues of $\mathbf{Y}$.

When we have to deal with high dimensional small-sample problems on any view, we can extend this formulation to work with its dual formulation, since it results in a more computationally efficient algorithm. For this purpose, let us consider view one has much more input features than data ($\mathrm{D}_1 >> \mathrm{N}$), known as fat-data, then $\mathbf{W}^{(1)}$ can be expressed as a linear combination of the inputs and some dual variables $\mathbf{A}^{(1)}$, i.e., $\mathbf{W}^{(1)} = \mathbf{X}^{(1)^{\mathrm{T}}}\,\mathbf{A}^{(1)}$. This way, we can express (2.26) as:

$$\begin{aligned} \min_{\mathbf{W}^{(1)},\mathbf{W}^{(2)}} \quad & \left\|(\mathbf{X}^{(2)} - \mathbf{K}_{\mathbf{X}^{(1)}}\,\mathbf{A}^{(1)}\,\mathbf{W}^{(2)^{\mathrm{T}}})\mathbf{\Gamma}^{1/2}\right\|_{\mathrm{F}}^{2} + \lambda \left\|\mathbf{A}^{(1)}\right\|_{\mathrm{F}}^{2}, \\ \text{s.t.} \quad & \mathbf{W}^{(2)^{\mathrm{T}}}\,\mathbf{\Gamma}\,\mathbf{W}^{(2)} = \mathbf{I}_{\mathrm{K_c}} \end{aligned} \tag{2.36}$$

where $\mathbf{K}_{\mathbf{X}^{(1)}} = \mathbf{X}^{(1)}\,\mathbf{X}^{(1)^{\mathrm{T}}}$ is the linear kernel matrix of the input data and we have included a regularisation term over $\mathbf{A}^{(1)}$ to overcome the ill-conditioned problems[1].

Following a similar process to the primal formulation, we can obtain the lagrange function of (2.36), derive it with respect to $\mathbf{A}^{(1)}$ and equate it to 0 to get:

$$\mathbf{A}^{(1)} \;=\; (\mathbf{K}_{\mathbf{X}^{(1)}}\mathbf{K}_{\mathbf{X}^{(1)}} + \lambda\mathbf{I})^{-1}\mathbf{K}_{\mathbf{X}^{(1)}}\,\mathbf{X}^{(2)}\,\mathbf{\Gamma}\,\mathbf{W}^{(2)}\,. \tag{2.37}$$

Now, we can substitute (2.37) into (2.36), and derive the resulting expression with respect to $\mathbf{W}^{(2)}$. After this, if we equate to zero to find the minimum, we get that $\mathbf{W}^{(2)^*} = \mathbf{\Gamma}^{1/2}\,\mathbf{W}^{(2)}$ can be obtained as the solution of the following eigenvector problem:

$$\mathbf{\Gamma}^{1/2}\,\mathbf{X}^{(2)^{\mathrm{T}}}\,\mathbf{K}_{\mathbf{X}^{(1)}}(\mathbf{K}_{\mathbf{X}^{(1)}}\mathbf{K}_{\mathbf{X}^{(1)}} + \lambda\mathbf{I})^{-1}\mathbf{K}_{\mathbf{X}^{(1)}}\,\mathbf{X}^{(2)}\,\mathbf{\Gamma}^{1/2}\,\mathbf{W}^{(2)^*} \;=\; \mathbf{W}^{(2)^*}\,\mathbf{\Lambda}. \tag{2.38}$$

And, finally, $\mathbf{W}^{(2)^*}$ can be used to calculate $\mathbf{A}^{(1)}$ as:

$$\mathbf{A}^{(1)} \;=\; (\mathbf{K}_{\mathbf{X}^{(1)}}\mathbf{K}_{\mathbf{X}^{(1)}} + \lambda\mathbf{I})^{-1}\mathbf{K}_{\mathbf{X}^{(1)}}\,\mathbf{X}^{(2)}\,\mathbf{\Gamma}^{1/2}\,\mathbf{W}^{(2)^*}\,. \tag{2.39}$$

Note that the solution of (2.38) involves operating with matrices of size D_2, instead of classical MVA approaches which work with matrices of size N. This advantage is especially significant in classification/regression problems, where the number of dimensions of the output view (number of classes or number of regressors) is usually lower than the number of training data ($\mathrm{D}_2 << \mathrm{N}$), thus leading to significant computational cost reduction.

2.2 Review of Factor Analysis

In this section we will present the previous MVA algorithms from a Bayesian point of view, commonly known as Factor Analysis (FA). To introduce FA methods, we will start with their probabilistic formulations, where we will obtain the model parameters (projection matrix) by the maximisation of their likelihood. Next, we will introduce their Bayesian versions, where we will include a prior over the model variables to be able to characterise them with the posterior distribution.

Finally, we will rely on the ability of these algorithms to include prior information over the model variables and describe how specific priors can lead to models that automatically select the number of latent factors ($\mathrm{K_c}$).

2.2.1 Probabilistic Factor Analysis

The first approaches to analyse consist in the probabilistic versions of PCA and CCA methods. These formulations combine the definition of a latent space with the likelihood function which measures the goodness of the model parameters to fit the data observations. This way, we can find the optimum value of the model parameters by a Maximum Likelihood (ML) strategy.

[1] Note that the inclusion of the regularisation term over $\mathbf{A}^{(1)}$ prevents problems in the calculation of the inverse of $\mathbf{K}_{\mathbf{X}^{(1)}}\mathbf{K}_{\mathbf{X}^{(1)}}$ in (2.37). These issues should not appear when working with high dimensional data, however they can occur in case of high redundancy among variables.

2.2.1.1 Probabilistic Principal Component Analysis

As happened with PCA, here we will work with a single data view, $\mathbf{X} \in \mathbb{R}^{\mathrm{N} \times \mathrm{D}}$. The Probabilistic PCA (PPCA) (Tipping and Bishop, 1999) considers that some independent latent variables $\mathbf{Z}$ exist, distributed following a Gaussian distribution with zero mean and unitary variance,

$$\mathbf{Z} \sim \mathcal{N}(0, \mathbf{I}_{\mathrm{K_c}}), \tag{2.40}$$

where $\mathrm{K_c} < \mathrm{D}$. As shown in the graphic model in Figure 2.1, we can generate the data observations as a linear combination of these latent random variables (r.v.) with a projection matrix $\mathbf{W}$ plus some Gaussian noise of zero mean and precision τ, so that

$$\mathbf{X} \,|\, \mathbf{Z} \sim \mathcal{N}\big(\mathbf{Z}\,\mathbf{W}^{\mathrm{T}}, \tau^{-1}\,\mathbf{I}_{\mathrm{D}}\big). \tag{2.41}$$

From this model definition, we can now obtain the likelihood of the model parameters, $\mathbf{W}$ and τ, using the properties of Gaussian distributions to marginalise the latent variables reaching to

$$\mathbf{X} \,|\, \mathbf{W}, \tau \sim \int \mathcal{N}\big(\mathbf{Z}\,\mathbf{W}^{\mathrm{T}}, \tau^{-1}\,\mathbf{I}_{\mathrm{D}}\big)\mathcal{N}(0, \mathbf{I}_{\mathrm{K_c}})d\,\mathbf{Z} = \mathcal{N}\big(0, \mathbf{W}\,\mathbf{W}^{\mathrm{T}} + \tau^{-1}\,\mathbf{I}_{\mathrm{D}}\big). \tag{2.42}$$

This way, we can determine the log-likelihood as

$$\ln \mathrm{p}(\mathbf{X} \,|\, \mathbf{W}, \tau) = -\frac{\mathrm{N}}{2} \ln |\,\mathbf{\Sigma}\,| + \mathrm{Tr}\{\mathbf{\Sigma}^{-1}\,\mathbf{C}_{\mathbf{XX}}\} + \mathrm{const} \tag{2.43}$$

where $\mathbf{\Sigma} = \mathbf{W}\,\mathbf{W}^{\mathrm{T}} + \tau^{-1}\,\mathbf{I}_{\mathrm{D}}$.

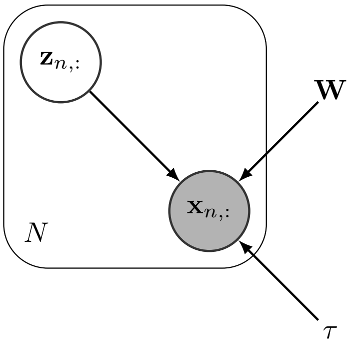


Figure 2.1: Plate diagram for the probabilistic PCA graphical model. Grey circles denote observed variables, white circles unobserved r.v. Nodes without a circle correspond to the model parameters.

We can now obtain the ML solution for $\hat{\mathbf{W}}_{\mathrm{ML}}$, by calculating the gradient of (2.43) with respect to $\mathbf{W}$ and making it equal to 0:

$$\frac{\partial \ln \mathrm{p}(\mathbf{X} \,|\, \mathbf{W}, \tau)}{\partial\,\mathbf{W}}\bigg|_{\mathbf{W}=\hat{\mathbf{W}}_{\mathrm{ML}}} = N\Big(-\mathbf{\Sigma}^{-1}\,\hat{\mathbf{W}}_{\mathrm{ML}} + \mathbf{\Sigma}^{-1}\,\mathbf{C}_{\mathbf{XX}}\,\mathbf{\Sigma}^{-1}\,\hat{\mathbf{W}}_{\mathrm{ML}}\Big) = \mathbf{0}. \tag{2.44}$$

If we remove the trivial solutions $\mathbf{\Sigma}^{-1} = 0$ and $\hat{\mathbf{W}}_{\mathrm{ML}} = 0$, which corresponds to minimums of the log-likelihood (see Tipping and Bishop (1999) for further details), we have that the likelihood is maximised when

$$-\mathbf{I}_{\mathrm{D}} + \mathbf{C}_{\mathbf{XX}}\Big(\hat{\mathbf{W}}_{\mathrm{ML}}\,\hat{\mathbf{W}}_{\mathrm{ML}}^{\mathrm{T}} + \tau^{-1}\,\mathbf{I}_{\mathrm{D}}\Big)^{-1} = \mathbf{0}. \tag{2.45}$$

Therefore, the solution of this problem leads to

$$\hat{\mathbf{W}}_{\mathrm{ML}}\,\hat{\mathbf{W}}_{\mathrm{ML}}^{\mathrm{T}} \;=\; \mathbf{C}_{\mathbf{XX}} - \tau^{-1}\,\mathbf{I}_{\mathrm{D}}, \tag{2.46}$$

where we can define the SVD of $\mathbf{C}_{\mathbf{XX}}$ as $\mathbf{V}_{\mathbf{C}_{\mathbf{XX}}}\,\mathbf{\Lambda}_{\mathbf{C}_{\mathbf{XX}}}\,\mathbf{V}_{\mathbf{C}_{\mathbf{XX}}}^{\mathrm{T}}$ and substitute it into (2.46), to get

$$\begin{aligned}\hat{\mathbf{W}}_{\mathrm{ML}}\,\hat{\mathbf{W}}_{\mathrm{ML}}^{\mathrm{T}} &= \mathbf{V}_{\mathbf{C}_{\mathbf{XX}}}\left(\mathbf{\Lambda}_{\mathbf{C}_{\mathbf{XX}}} - \tau^{-1}\,\mathbf{I}_{\mathrm{D}}\right)\mathbf{V}_{\mathbf{C}_{\mathbf{XX}}}^{\mathrm{T}}\\ \hat{\mathbf{W}}_{\mathrm{ML}} &= \mathbf{V}_{\mathbf{C}_{\mathbf{XX}}}\left(\mathbf{\Lambda}_{\mathbf{C}_{\mathbf{XX}}} - \tau^{-1}\,\mathbf{I}_{\mathrm{D}}\right)^{1/2}\mathbf{R} \end{aligned}\tag{2.47}$$

where $\mathbf{R}$ is an arbitrary orthogonal matrix so that $\mathbf{R}\,\mathbf{R}^{\mathrm{T}} = \mathbf{I}$, e.g. an identity matrix ($\mathbf{I}$).

If we now substitute this result in the loglikelihood, Equation (2.43), and calculate the partial derivative with respect to τ we get that the noise precision can be estimated as

$$\tau_{ML}^{-1} \;=\; \frac{1}{\mathrm{D}-\mathrm{K_c}}\sum_{j=\mathrm{K_c}+1}^{\mathrm{D}}\lambda_j \tag{2.48}$$

which implies that this variable contains the information lost doing the data projection. That is, as we are making a dimensionality reduction into a low dimensional latent space, this variable retains the information of the difference between the original and the projected data.

Furthermore, note that if $\tau_{ML}^{-1} \rightarrow 0$ in equation (2.47), the resulting projection matrix $\hat{\mathbf{W}}_{\mathrm{ML}} \rightarrow \mathbf{V}_{\mathbf{C}_{\mathbf{XX}}}\,\mathbf{\Lambda}_{\mathbf{C}_{\mathbf{XX}}}^{1/2}$, which is equivalent to the result obtained in Section 2.1.1 re-scaled by the eigenvalues. This implies that the directions are conserved while including the re-scaling factor $\mathbf{\Lambda}_{\mathbf{C}_{\mathbf{XX}}}^{1/2}$.

2.2.1.2 Probabilistic Canonical Correlation Analysis

For this formulation, we are working with M data views, given that $\mathbf{X}^{(\mathrm{m})} \in \mathbb{R}^{\mathrm{N}\times\mathrm{D_m}}$ with $\mathrm{m} = 1,\ldots,\mathrm{M}$. This way, $\mathbf{X}^{(\mathrm{m})}$ represents the matrix $\mathbf{X}$ of the m-th view and $\mathbf{X}^{\{\mathcal{M}\}}$ represents all the matrices $\mathbf{X}$ of the views in the $\mathcal{M}$ set. If $\mathcal{M} = \{1,2,\ldots,\mathrm{M}\}$, then $\mathbf{x}_{\mathrm{n},:}^{\{\mathcal{M}\}} = \{\mathbf{x}_{\mathrm{n},:}^{(1)}, \mathbf{x}_{\mathrm{n},:}^{(2)}, \ldots, \mathbf{x}_{\mathrm{n},:}^{(\mathrm{M})}\}$ is the complete n-th observation.

The Probabilistic version of CCA (PCCA) was first introduced by Bach and Jordan (2005) and relies on the PPCA formulation extending it to more than one data view. Considering there exists a set of latent variables $\mathbf{Z}$

$$\mathbf{Z} \;\sim\; \mathcal{N}(0, \mathbf{I}_{\mathrm{K_c}}), \tag{2.49}$$

where $0 < \mathrm{K_c} \leq \min(\mathrm{D}_1,\ldots,\mathrm{D_m})$. As the graphical model of Figure 2.1 shows, for the particular case with two data views, each data view is generated by combining these latent variables with the projection matrix, $\mathbf{W}^{(\mathrm{m})}$, and adding some Gaussian noise with zero noise and precision $\tau^{(\mathrm{m})}$,

$$\mathbf{X}^{(\mathrm{m})}\,|\,\mathbf{Z} \;\sim\; \mathcal{N}\Big(\mathbf{Z}\,\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}, \tau^{(m)^{-1}}\,\mathbf{I}_{\mathrm{D_m}}\Big) \tag{2.50}$$

We can now apply Bayes rule to marginalise the distribution with respect to the latent variables $\mathbf{Z}$ and obtain the likelihood of the model parameters

$$\begin{aligned}\mathbf{X}^{\{\mathcal{M}\}}\,|\,\mathbf{W}^{\{\mathcal{M}\}}, \boldsymbol{\tau}^{\{\mathcal{M}\}} &\sim \int \mathcal{N}\Big(\mathbf{Z}\,\mathbf{W}^{\{\mathcal{M}\}^{\mathrm{T}}}, \boldsymbol{\tau}^{\{\mathcal{M}\}^{-1}}\,\mathbf{I}_{\mathrm{D_{tot}}}\Big)\mathcal{N}(0, \mathbf{I}_{\mathrm{K_c}})d\,\mathbf{Z}\\ &= \mathcal{N}\Big(0, \mathbf{W}^{\{\mathcal{M}\}}\,\mathbf{W}^{\{\mathcal{M}\}^{\mathrm{T}}} + \boldsymbol{\tau}^{\{\mathcal{M}\}^{-1}}\,\mathbf{I}_{\mathrm{D_{tot}}}\Big). \end{aligned}\tag{2.51}$$

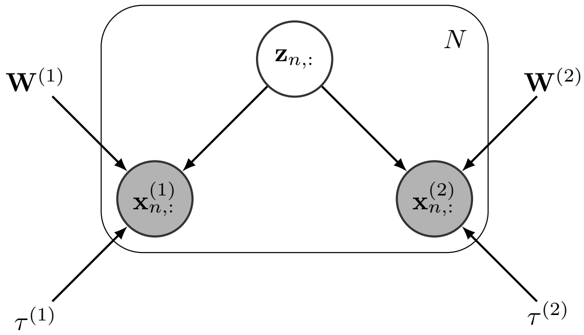


Figure 2.2: Plate diagram for the probabilistic CCA graphical model with only two views. Grey circles denote observed variables, white circles unobserved r.v. Nodes without a circle correspond to the model parameters.

where $\mathrm{D_{tot}} = \sum_{\mathrm{m}=1}^{\mathrm{M}} \mathrm{D_m}$, $\mathbf{X}^{\{\mathcal{M}\}} = \left[\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\mathrm{M})}\right]^{\mathrm{T}}$, $\mathbf{W}^{\{\mathcal{M}\}} = \left[\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(\mathrm{M})}\right]^{\mathrm{T}}$ and the noise precision is a vector stacked to have the same dimensions as the previous matrices, having $\mathrm{D_m}$ times $\tau^{(\mathrm{m})}$, so $\boldsymbol{\tau}^{\{\mathcal{M}\}} = \left[\tau^{(1)}, \ldots, \tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(2)}, \ldots, \tau^{(\mathrm{M})}\right]^{\mathrm{T}}$ of dimension $\mathrm{D_{tot}}$. Furthermore, we can now determine the value of each $\mathbf{W}^{(\mathrm{m})}$ that maximises the likelihood by deriving with respect to $\mathbf{W}^{(\mathrm{m})}$ and equating to zero, as in (2.44). Bach and Jordan (2005) proved that through ML

$$\hat{\mathbf{W}}^{(1)}_{\mathrm{ML}} = \mathbf{C}^{1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}} \mathbf{V}^{(1)} \mathbf{R}_1 \tag{2.52}$$

$$\hat{\mathbf{W}}^{(2)}_{\mathrm{ML}} = \mathbf{C}^{1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}} \mathbf{V}^{(2)} \mathbf{R}_2 \tag{2.53}$$

where $\mathbf{V}^{(1)}$ corresponds to the left eigenvector of $\mathbf{Y} = \mathbf{C}^{\text{-}1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}} \mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(2)}} \mathbf{C}^{\text{-}1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}$, $\mathbf{V}^{(2)}$ corresponds to the right eigenvector, $\mathbf{R}_1$ and $\mathbf{R}_2$ are two arbitrary matrices such that $\mathbf{R}_1 \mathbf{R}_2^{\mathrm{T}} = \mathbf{\Lambda}_{\mathbf{Y}}$ with the spectral norms of $\mathbf{R}_1$ and $\mathbf{R}_2$ smaller than one and $\mathbf{\Lambda}_{\mathbf{Y}}$ is a diagonal matrix with the eigenvalues of $\mathbf{Y}$. Note that equations (2.52) and (2.53) correspond to equations (2.21) and (2.22) re-scaled by the eigenvalues of $\mathbf{Y}$ and with a rotation given by the change of sign of $\mathbf{C}^{1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}$ and $\mathbf{C}^{1/2}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}}$[2].

2.2.2 Bayesian Factor Analysis

Bayesian formulations allow us to include prior information over the model parameters, r.v., so that we can characterise them with their posterior distribution. This way, we can use this posterior distribution to make estimations (such as MSE or Maximum A Posteriori (MAP)), or to generate realisations by sampling from their posterior distribution. In this section, we will make use of Bayesian inference to estimate this posterior distribution based on the observed data.

Considering Θ to be the group of all model variables, the goal of Bayesian inference is to model the posterior distribution of all variables, $p(\Theta|\,\mathbf{X})$. However, this posterior distribution is not always tractable, as happens with the models hereunder. In these cases, we can use variational

[2] Calculating the SVD of $\mathbf{C}^{\text{-}1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}} = \mathbf{V}^{(1)} \mathbf{\Lambda}^{-1}_{\mathbf{C}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}} \mathbf{V}^{(1)^{\mathrm{T}}}$ we can observe that it is equivalent to the SVD of $\mathbf{C}^{1/2}_{\mathbf{X}^{(1)}\mathbf{X}^{(1)}}$ with re-scaled singular values, maintaining the original vectors directions.

inference to approximate $p(\Theta|\,\mathbf{X})$ by a treatable distribution $q(\Theta)$. In order to adjust $q(\Theta)$, variational inference minimises the Kullback-Leibler (KL) divergence between q and p defined as

$$KL(q||p) \quad = \quad \int q(\Theta)\ln\left(\frac{q(\Theta)}{p(\Theta|\,\mathbf{X})}\right)d\Theta, \tag{2.54}$$

which measures the difference between the distributions p and q. If we develop this equation, we have

$$\begin{aligned} KL(q||p) &= \int q(\Theta)\ln\left(\frac{q(\Theta)}{\frac{p(\mathbf{X},\Theta)}{p(\mathbf{X})}}\right)d\Theta = \int q(\Theta)\ln\left(\frac{q(\Theta)}{p(\mathbf{X},\Theta)}\right)d\Theta + \int q(\Theta)\ln(p(\mathbf{X}))d\Theta \\ &= \int q(\Theta)\ln\left(\frac{q(\Theta)}{p(\mathbf{X},\Theta)}\right)d\Theta + \ln(p(\mathbf{X})). \end{aligned} \tag{2.55}$$

Defining

$$L(q) = -\int q(\Theta)\ln\left(\frac{q(\Theta)}{p(\mathbf{X},\Theta)}\right)d\Theta, \tag{2.56}$$

we can see that $L(q)$ acts as a lower bound to the data distribution

$$L(q) \quad = \quad \ln(p(\mathbf{X})) - KL(q||p) \leq \ln(p(\mathbf{X})). \tag{2.57}$$

For this reason, maximising $L(q)$ implies minimising $KL(q||p)$; in fact, $L(q)$ will reach its maximum value when $p = q$ (i.e., $KL(q||p) = 0$). This simple trick allows us to approximate the posterior distribution of $p(\Theta|\,\mathbf{X})$ to another distribution $q(\Theta)$ that can be calculated, e.g. using the mean field approximation.

The mean field method

The mean field approach (Blei et al., 2017) considers that the posterior distribution of all the model variables $p(\Theta|\,\mathbf{X})$ can be approximated by a $q(\Theta)$ distribution, which factorises over all variables, i.e.,

$$q(\Theta) \quad = \quad \prod_i q(\Theta_i) = \prod_i q_i. \tag{2.58}$$

where Θ_i represents the i-th model variable and q_i is its corresponding approximated distribution. Using the variational inference, we can now obtain each q_i factor by maximising $L(q)$. To do so, let us substitute (2.58) into (2.56) and apply some mathematical manipulations

$$\begin{aligned} L(q_j) &= \int q(\Theta)\ln\left(\frac{p(\mathbf{X},\Theta)}{q(\Theta)}\right)d\Theta = \int \prod_i q_i\left[\ln(p(\mathbf{X},\Theta)) - \sum_i \ln(q_i)\right]d\Theta \\ &= \int \prod_i q_i \ln(p(\mathbf{X},\Theta))d\Theta - \int \prod_i q_i \sum_i \ln(q_i)d\Theta \\ &= \int q_j \prod_{i\neq j} q_i \ln(p(\mathbf{X},\Theta))d\Theta - \int q_j \prod_{i\neq j} q_i \left(\ln(q_j) + \sum_{i\neq j}\ln(q_i)\right)d\Theta \\ &= \int q_j \prod_{i\neq j} q_i \ln(p(\mathbf{X},\Theta))d\Theta - \int q_j \prod_{i\neq j} q_i \sum_{i\neq j}\ln(q_i)d\Theta - \int q_j \prod_{i\neq j} q_i \ln(q_j)d\Theta \\ &= \int q_j\left[\int \prod_{i\neq j} q_i \ln(p(\mathbf{X},\Theta))d\Theta_i\right]d\Theta_j - \int q_j \ln(q_j)d\Theta_j + \text{const.} \end{aligned} \tag{2.59}$$

If we now define

$$\ln(f_j) \;=\; \mathbb{E}_{-q_j}[\ln(p(\mathbf{X},\Theta))] + \text{const}, \tag{2.60}$$

where $-q_j$ means that we calculate this expectation on all model variables except the j-th variable, we can rewrite the lower bound as

$$L(q_j) \;=\; \int q_j \ln(f_j) d\Theta_j - \int q_j \ln(q_j) d\Theta_j + \text{const}. \tag{2.61}$$

Now, taking into account that (2.61) is a negative KL between q_j and f_j, we can maximise $L(q_j)$ by minimising $KL(q_j||f_j)$. We have that the value of $\ln(q_j)$ that minimises the $KL(q_j||f_j)$ and, therefore, provides the optimum solution has the following expression

$$\ln(q_j^*) \;=\; \mathbb{E}_{-q_j}[\ln(p(\mathbf{X},\Theta))] + \text{const}. \tag{2.62}$$

This constitutes the basis of the mean field variational inference that we will use throughout this thesis to solve the different inference problems presented. You can find a more in-depth explanation on variational inference in Bishop (2006); Murphy (2012).

2.2.2.1 Bayesian Principal Component Analysis

Here we explain the Bayesian extension of the PPCA firstly presented in Bishop (1999). To do so, we adapt its probabilistic version by including a prior over the model parameters $(\mathbf{W}, \tau)$. We will firstly define its generative model, including the variable distributions and the graphic model, to later present the result obtained by variational inference.

Generative model

As happened with its probabilistic counterpart, for this model we start with a latent representation of the samples, $\mathbf{Z}$. These latent variables are linearly combined with the projection matrix $\mathbf{W}$ and some Gaussian noise of zero mean and precision τ is added to generate the observations, $\mathbf{X}$. Figure 2.3 shows the graphical model associated to Bayesian PCA, which includes the different variables of the model as well as the relation between them.

To complete the model, we define the following distributions:

$$\mathbf{z}_{\mathrm{n},:} \sim \mathcal{N}(0, \mathbf{I}_{\mathrm{K_c}}) \tag{2.63}$$

$$\mathbf{W} \sim \mathcal{N}(0, \mathbf{I}_{\mathrm{K_c}}) \tag{2.64}$$

$$\mathbf{x}_{\mathrm{n},:} \,|\, \mathbf{z}_{\mathrm{n},:} \sim \mathcal{N}(\mathbf{z}_{\mathrm{n},:}\mathbf{W}^{\mathrm{T}}, \tau^{-1}\mathbf{I}_{\mathrm{D}}) \tag{2.65}$$

$$\tau \sim \Gamma(a^{\tau}, b^{\tau}) \tag{2.66}$$

where $\mathbf{z}_{\mathrm{n},:} \in \mathbb{R}^{1\times \mathrm{K_c}}$ is the low-dimension latent variable for the n-th data point $\mathbf{x}_{\mathrm{n},:}$ and $\Gamma(a,b)$ is a Gamma distribution with parameters a and b.

Variational inference

As previously stated, the posterior distribution is not computable, so we can resort to calculate an approximate distribution through mean-field variational inference. To do so, as proposed

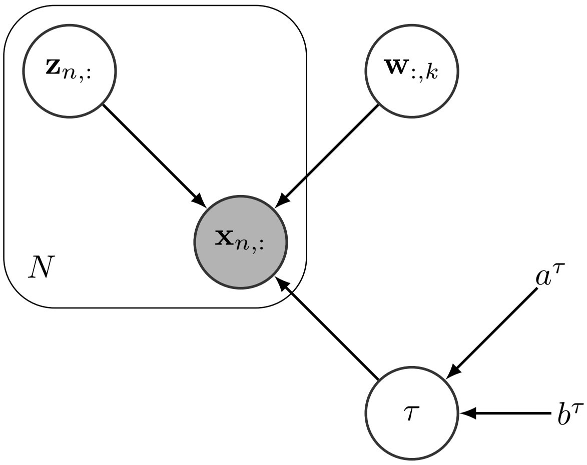


Figure 2.3: Plate diagram for the Bayesian PCA graphical model. Grey circles denote observed variables and white circles unobserved r.v. Nodes without a circle are hyperparameters.

in the mean-field approach, we choose a fully factorised variational family to approximate the posterior distribution

$$p(\Theta|\,\mathbf{X}) \;\; \approx \; q(\Theta) = q(\mathbf{W})q(\tau)\prod_{n=1}^{N} q(\mathbf{z}_{\text{n},:}). \tag{2.67}$$

The mean-field method maximises the lower bound by means of an iterative coordinate-ascent like optimisation problem where the optimum value of each factor is obtained by (2.62), when the rest of the variables are fixed. This way we can determine these distributions by calculating the expectations with respect to the different r.v. of the problem, see Sevilla-Salcedo (2021) for the full development of the r.v. update rules. Table 2.1 includes the q_j distribution of each parameter as well as the final mean-field factors update rules, where we stuck the projected matrix, $\mathbf{z}_{\text{n},:}$, of all samples, $\mathbf{Z}$, for a simpler notation and we use <> to represent the mean value of the r.v. Once these approximated distributions are determined, we can generate new observations by sampling from the approximated posterior distributions or making estimations of MSE or MAP. As the estimated distribution for each parameter also depends on some other parameters, e.g. $\langle\mathbf{z}_{\text{n},:}\rangle$ depends on $\langle\tau\rangle$ and $\langle\mathbf{W}\rangle$, we need to iterate over the variables, analysing the evolution of the lower bound on each iteration.

2.2.2.2 Bayesian PCA with Automatic Relevance Determination

One of the main advantages of the Bayesian formulation is the ability to include some prior information to better adapt the model to the problem needs or expert knowledge. In particular, in this section we include an ARD prior (Neal, 2012). This new formulation of the BPCA with ARD (Bishop, 1999) allows the model to automatically determine the number of latent factors, which would otherwise need to be determined through different time consuming methods such as cross-validation.

Variable	q^* distribution	Parameters
$\mathbf{z}_{\mathrm{n},:}$	$\mathcal{N}(\mathbf{z}_{\mathrm{n},:}\,\lvert\langle\mathbf{z}_{\mathrm{n},:}\rangle, \Sigma_{\mathbf{Z}})$	$\langle\mathbf{z}_{\mathrm{n},:}\rangle = \langle\tau\rangle\,\mathbf{X}\langle\mathbf{W}\rangle\Sigma_{\mathbf{Z}}$ $\Sigma_{\mathbf{Z}}^{-1} = \mathbf{I}_{\mathrm{K_c}} + \langle\tau\rangle\langle\mathbf{W}^{\mathrm{T}}\,\mathbf{W}\rangle$
$\mathbf{W}$	$\prod_{\mathrm{d}=1}^{\mathrm{D}}\mathcal{N}(\mathbf{W}\,\lvert\langle\mathbf{W}\rangle, \Sigma_{\mathbf{W}})$	$\langle\mathbf{W}\rangle = \langle\tau\rangle\,\mathbf{X}^{\mathrm{T}}\langle\mathbf{Z}\rangle\Sigma_{\mathbf{W}}$ $\Sigma_{\mathbf{W}}^{-1} = \mathbf{I}_{\mathrm{K_c}} + \langle\tau\rangle\langle\mathbf{Z}^{\mathrm{T}}\,\mathbf{Z}\rangle$
τ	$\Gamma(\tau\lvert a^{\tau}, b^{\tau})$	$a^{\tau} = \frac{DN}{2} + a_0^{\tau}$ $b^{\tau} = b_0^{\tau} + \frac{1}{2}\sum_{\mathrm{n}=1}^{\mathrm{N}}\sum_{\mathrm{d}=1}^{\mathrm{D}}\mathrm{x}_{\mathrm{n,d}}^2$ $-\,\mathrm{Tr}\big\{\langle\mathbf{W}\rangle\langle\mathbf{Z}^{\mathrm{T}}\rangle\,\mathbf{X}\big\}$ $+\frac{1}{2}\,\mathrm{Tr}\big\{\langle\mathbf{W}^{\mathrm{T}}\,\mathbf{W}\rangle\langle\mathbf{Z}^{\mathrm{T}}\,\mathbf{Z}\rangle\big\}$

Table 2.1: Updated rules for q distributions for the different r.v. of the graphical model. These expressions have been obtained using the update rules of the mean field approximation (2.67). See section Bayesian PCA at Sevilla-Salcedo (2021) for further details.

Generative model

As Figure 2.4 shows, the model uses the latent factors $\mathbf{Z}$ to, combined with $\mathbf{W}$ and some Gaussian noise with zero mean and precision τ, generate the data observations $\mathbf{X}$. Unlike the previous Bayesian model, for each column $\mathbf{w}_{:,\mathrm{k}}$ of the projection matrix, we have a common r.v. α_k that governs its variance. The inclusion of this r.v. modelled as a normal-gamma prior, constitutes an ARD prior Neal (2012), which allows the model to enforce column-wise sparsity in the latent factors. According to this generative model, the distributions of their r.v. are given by

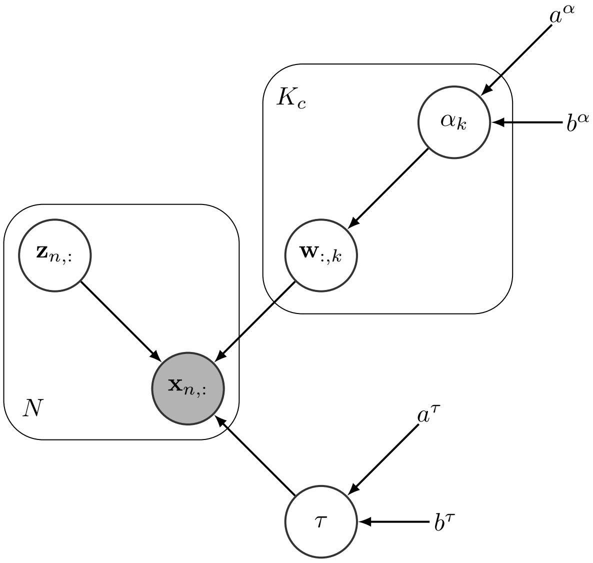


Figure 2.4: Plate diagram for the Bayesian PCA with ARD prior graphical model. Grey circles denote observed variables, white circles unobserved r.v. Hyperparameters do not have a circle.

$$\begin{aligned}
\mathbf{z}_{\mathrm{n},:} &\sim \mathcal{N}(0, \mathbf{I}_{\mathrm{K_c}}) && (2.68)\\
\mathbf{w}_{:,\mathrm{k}} &\sim \mathcal{N}\big(0, {\alpha_k}^{-1}\,\mathbf{I}_{\mathrm{K_c}}\big) && (2.69)\\
\mathbf{x}_{\mathrm{n},:}\,|\,\mathbf{z}_{\mathrm{n},:} &\sim \mathcal{N}(\mathbf{z}_{\mathrm{n},:}\,\mathbf{W}^{\mathrm{T}}, \tau^{-1}\,\mathbf{I}_{\mathrm{D}}) && (2.70)\\
\alpha_k &\sim \Gamma(a^{\boldsymbol{\alpha}}, b^{\boldsymbol{\alpha}}) && (2.71)\\
\tau &\sim \Gamma(a^{\tau}, b^{\tau}). && (2.72)
\end{aligned}$$

where $\mathbf{z}_{\mathrm{n},:} \in \mathbb{R}^{1\times\mathrm{K_c}}$ is the low-dimension latent variable for the n-th data point, $\Gamma(a, b)$ is a Gamma distribution with parameters a and b, and $\mathbf{w}_{:,\mathrm{k}}$ is the k-th column of matrix $\mathbf{W}$ (of dimensions $\mathrm{D}\times\mathrm{K_c}$). Note that the prior distribution of $\boldsymbol{\alpha}$ along with the distribution of $\mathbf{W}$ makes an ARD prior which induces column-wise sparsity. Consequently, when α_k takes high values, the associated k-th latent factor will tend to 0, so the associated $\mathbf{z}_{:,\mathrm{k}}$ will not have influence in the generation of $\mathbf{x}_{\mathrm{n},:}$ and, thus, the k-th latent factor can be considered unnecessary for the problem. This way, we have an automatic latent factor selection where, by initially defining a high number of latent factors, it can automatically determine which of them are really necessary for the problem, avoiding the need to cross-validate the number of latent factors, $\mathrm{K_c}$.

Variational inference

Parting from this generative model, we can use mean-field variational inference to approximate the posterior distribution of the variables given the observed data,

$$p(\Theta|\,\mathbf{X}) \;\approx\; q(\Theta) = q(\mathbf{W})q(\tau)\prod_{\mathrm{k=1}}^{\mathrm{K_c}} q(\alpha_k)\prod_{\mathrm{n=1}}^{\mathrm{N}} q(\mathbf{z}_{\mathrm{n},:}). \qquad (2.73)$$

Therefore, we can compute this mean-field to maximise (2.57), obtaining the value of each factor that maximises the lower bound by (2.62), fixing the rest of the factors. Consequently, we can determine the estimated distribution for each r.v.by calculating the expectations with respect to the rest see Sevilla-Salcedo (2021) for the complete development of these updates. Table 2.1 describes each parameter q_j estimated distribution along with the mean-field factors update rules for the version of PCA with an ARD prior over $\mathbf{W}$, where we stuck the projected matrix, $\mathbf{z}_{\mathrm{n},:}$, of all samples, $\mathbf{Z}$, for a simpler notation and <> represents the mean value of the r.v.

2.2.2.3 Bayesian CCA or Bayesian Inter-Battery Factor Analysis (BIBFA)

Following the Bayesian formulation of PCA, we can extend it to work with multiple views, which can be interpreted as the Bayesian CCA, also known as Bayesian Inter-Battery Factor Analysis (BIBFA) (Klami et al., 2013). In this section we briefly review this model, describing the generative model, the variational inference and, additionally, introducing its predictive model. The extended justification of the variational equations and the calculation of the lower bound are available in section BIBFA at Sevilla-Salcedo (2021).

Generative model

BIBFA is a FA analysis model which stands out for its ability to determine a common latent projection space for different data views. This algorithm looks for linear projections of the

Variable	q^* distribution	Parameters
$\mathbf{z}_{\mathrm{n},:}$	$\mathcal{N}(\mathbf{z}_{\mathrm{n},:}\,\vert\langle\mathbf{z}_{\mathrm{n},:}\rangle,\Sigma_{\mathbf{Z}})$	$\langle\mathbf{z}_{\mathrm{n},:}\rangle = \langle\tau\rangle\,\mathbf{X}\langle\mathbf{W}\rangle\Sigma_{\mathbf{Z}}$ $\Sigma_{\mathbf{Z}}^{-1} = \mathbf{I}_{\mathrm{K_c}} + \langle\tau\rangle\langle\mathbf{W}^{\mathrm{T}}\,\mathbf{W}\rangle$
$\mathbf{W}$	$\prod_{\mathrm{d}=1}^{\mathrm{D}}\mathcal{N}(\mathbf{W}\,\vert\langle\mathbf{W}\rangle,\Sigma_{\mathbf{W}})$	$\langle\mathbf{W}\rangle = \langle\tau\rangle\,\mathbf{X}^{\mathrm{T}}\langle\mathbf{Z}\rangle\Sigma_{\mathbf{W}}$ $\Sigma_{\mathbf{W}}^{-1} = diag(\langle\boldsymbol{\alpha}\rangle) + \langle\tau\rangle\langle\mathbf{Z}^{\mathrm{T}}\,\mathbf{Z}\rangle$
α_k	$\Gamma(\alpha_k\,\vert a^{\alpha_k}, b^{\alpha_k})$	$a^{\alpha_k} = \frac{\mathrm{D}}{2} + a_0^{\boldsymbol{\alpha}}$ $b^{\alpha_k} = b_0^{\boldsymbol{\alpha}} + \frac{1}{2}\langle\mathbf{W}^{\mathrm{T}}\,\mathbf{W}\rangle_{k,k}$
τ	$\Gamma(\tau\vert a^{\tau}, b^{\tau})$	$a^{\tau} = \frac{\mathrm{D\,N}}{2} + a_0^{\tau}$ $b^{\tau} = b_0^{\tau} + \frac{1}{2}\sum_{\mathrm{n}=1}^{\mathrm{N}}\sum_{\mathrm{d}=1}^{\mathrm{D}}\mathrm{x_{n,d}}^2$ $-\,\mathrm{Tr}\big\{\langle\mathbf{W}\rangle\langle\mathbf{Z}^{\mathrm{T}}\rangle\,\mathbf{X}\big\}$ $+\frac{1}{2}\,\mathrm{Tr}\big\{\langle\mathbf{W}^{\mathrm{T}}\,\mathbf{W}\rangle\langle\mathbf{Z}^{\mathrm{T}}\,\mathbf{Z}\rangle\big\}$

Table 2.2: Updated rules for q distributions for the different r.v.of the graphical model. These expressions have been obtained using the update rules of the mean field approximation (2.73). See section Bayesian PCA with ARD at Sevilla-Salcedo (2021) for further details.

observable data in a latent space by finding data correlations inter- and intra-views. In particular, we have a r.v. $\mathbf{Z}$ in a common latent space which, when combined with different per view projection matrices $\left\{\mathbf{W}^{(\mathrm{m})}\right\}_{\mathrm{m}=1}^{\mathrm{M}}$ and some Gaussian noise, can generate each data view $\mathbf{X}^{(\mathrm{m})}$. Therefore, we can define the joint probability distributions of the different model variables as:

$$\mathbf{z}_{\mathrm{n},:} \sim \mathcal{N}(0,\mathbf{I}_{\mathrm{K_c}}) \tag{2.74}$$

$$\mathbf{w}_{:,\mathrm{k}}^{(\mathrm{m})} \sim \mathcal{N}\left(0,\left(\alpha_{\mathrm{k}}^{(\mathrm{m})}\right)^{-1}\mathbf{I}_{\mathrm{K_c}}\right) \tag{2.75}$$

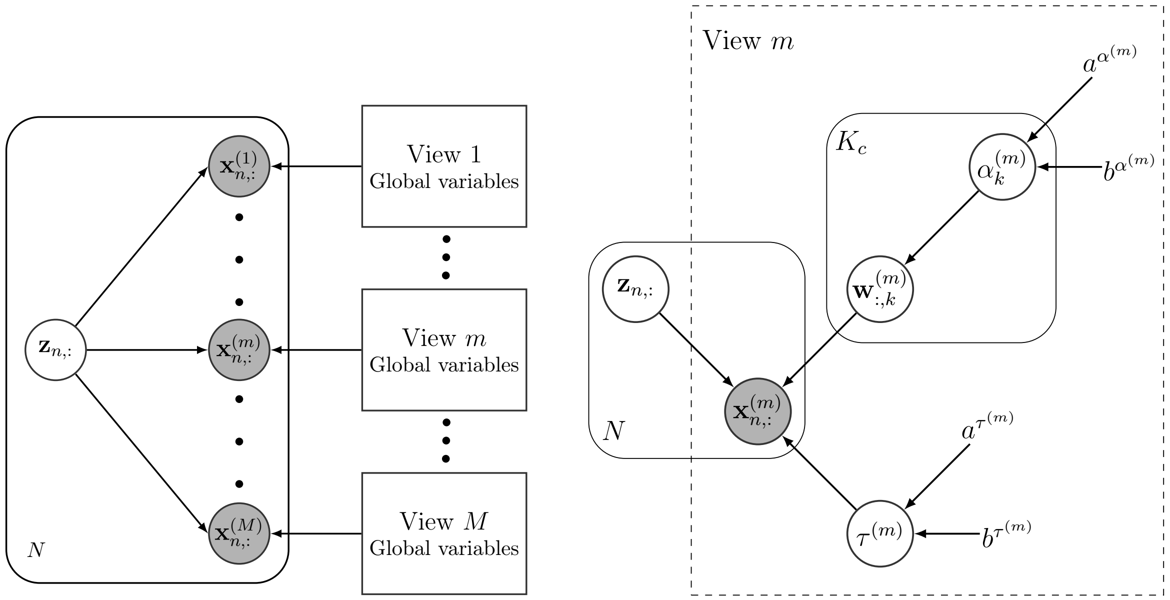
$$\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}\,\vert\,\mathbf{z}_{\mathrm{n},:} \sim \mathcal{N}(\mathbf{z}_{\mathrm{n},:}\,\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}},{\tau^{(\mathrm{m})}}^{-1}\,\mathbf{I}_{\mathrm{D_m}}) \tag{2.76}$$

$$\alpha_{\mathrm{k}}^{(\mathrm{m})} \sim \Gamma\left(a^{\boldsymbol{\alpha}^{(\mathrm{m})}},b^{\boldsymbol{\alpha}^{(\mathrm{m})}}\right) \tag{2.77}$$

$$\tau^{(\mathrm{m})} \sim \Gamma\left(a^{\tau^{(\mathrm{m})}},b^{\tau^{(\mathrm{m})}}\right) \tag{2.78}$$

where most variables keep the same definition from BPCA with ARD, $\mathbf{z}_{\mathrm{n},:} \in \mathbb{R}^{1\times\mathrm{K_c}}$ is the common low-dimension latent variable for the n-th data point, $\tau^{(\mathrm{m})}$ is the noise precision, $\Gamma(a,b)$ is a Gamma distribution with parameters a and b and $\mathbf{w}_{:,\mathrm{k}}^{(\mathrm{m})}$ is the k-th column of matrix $\mathbf{W}^{(\mathrm{m})}$ (of dimensions $\mathrm{D_m}\times\mathrm{K_c}$). The general model for any number of views of BIBFA is presented in Figure 2.5(a). The detailed graphical model for a single view, describing the generation of each data view is included in Figure 2.5(b).

If we analyse the structure obtained in the posterior distribution of the projection matrices $\mathbf{W}^{(\mathrm{m})}$ by the inclusion of the ARD prior we can determine four different scenarios for each latent factor: (1) The latent feature k has no value close to zero in any view ($\mathbf{w}_{:,\mathrm{k}}^{(\mathrm{m})} \neq 0 \quad \forall m$), having that this variable is learning information that is common to every view; (2) The latent factor k has all its values close to zero for every view ($\mathbf{w}_{:,\mathrm{k}}^{(\mathrm{m})} = 0 \quad \forall m$) and, therefore, this latent factor is not relevant and could be pruned; (3) The latent factor k is close to zero for some views and not for others ($\mathbf{w}_{:,\mathrm{k}}^{(\mathcal{M}_0)} = 0, \mathbf{w}_{:,\mathrm{k}}^{(\mathcal{M}_1)} \neq 0$), where the latent factor is relevant to explain certain views (where $\mathcal{M}_1$ is the subset of views where $\mathbf{w}_{:,\mathrm{k}}^{(\mathcal{M}_1)} \neq 0$); (4) The latent factor k only has non



(a) Multi-view model.

(b) Zoom on view m.

Figure 2.5: Plate diagram for the BIBFA graphical model. Grey circles denote observed variables, white circles unobserved r.v. The nodes without a circle correspond to the hyperparameters.

zero values for one view and the rest are zero, i.e. $\mathcal{M}_1$ is composed by just one view, which we will call private. This structure increases the interpretability of the results and improves the information learnt by the model. Furthermore, we could analyse either $\alpha_{\mathrm{k}}^{(\mathrm{m})}$ or $||\,\mathbf{w}_{:,\mathrm{k}}^{(\mathrm{m})}\,||_2^2$ to see whether a latent feature is relevant for any view and remove the latent feature if it is not relevant, automatically pruning the irrelevant ones. This implies that we can set the number of latent features $\mathrm{K_c}$ to a high value, so that the model only uses the relevant ones. This, in turn, removes an hyperparameter from the model, as the number of latent factors is automatically learnt.

Variational Inference

Once the generative model is defined, the next step would be to calculate the posterior distribution of all the model variables, Θ, given the observed data. However, due to the fact that we need to calculate the marginal likelihood of the data, i.e. the normalising factor in Baye's rule, this is not attainable. Nevertheless, we can make use of the mean-field variational inference to calculate an approximation Blei et al. (2017). To do so, we maximise a lower bound to the posterior and choose a fully factorised variational family, having that the posterior can be approximated as

$$p(\Theta|\,\mathbf{X}^{\{\mathcal{M}\}}) \;\approx\; q(\Theta) = \prod_{\mathrm{m}=1}^{\mathrm{M}} \left(q\left(\mathbf{W}^{(\mathrm{m})}\right) q\left(\tau^{(\mathrm{m})}\right) \prod_{\mathrm{k}=1}^{\mathrm{K_c}} q\left(\alpha_{\mathrm{k}}^{(\mathrm{m})}\right) \right) \prod_{n=1}^{N} q(\mathbf{z}_{\mathrm{n},:}) \tag{2.79}$$

where Θ comprises all r.v. in the model and q is the new distribution for each variable. This mean-field posterior can be calculated using a coordinate-ascent-like optimization, where we calculate the optimal value of each r.v. by fixing the rest and evaluating in (2.62). This formulation allows us to obtain the posterior distribution without simultaneously marginalising Θ from the joint distribution, see section BIBFA at Sevilla-Salcedo (2021) for a full equation development.

As with BPCA, Table 2.3 summarises the q_j distribution of each parameter calculated using Equation (2.62) over the r.v. to obtain the mean-field update rules. As previously explained, with these results, we can choose between sampling from the distributions, using the mean or using MSE or MAP to determine the estimation of each r.v.

Variable	**q^* distribution**	**Parameters**
$\mathbf{z}_{\mathrm{n},:}$	$\mathcal{N}(\mathbf{z}_{\mathrm{n},:} \,\vert \langle \mathbf{z}_{\mathrm{n},:} \rangle, \Sigma_{\mathbf{Z}})$	$\langle \mathbf{z}_{\mathrm{n},:} \rangle = \sum_{\mathrm{m}=1}^{\mathrm{M}} \langle \tau^{(\mathrm{m})} \rangle \, \mathbf{X}^{(\mathrm{m})} \langle \mathbf{W}^{(\mathrm{m})} \rangle \Sigma_{\mathbf{Z}}$ $\Sigma_{\mathbf{Z}}^{-1} = \mathbf{I}_{\mathrm{K_c}} + \sum_{\mathrm{m}=1}^{\mathrm{M}} \langle \tau^{(\mathrm{m})} \rangle \langle \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}} \, \mathbf{W}^{(\mathrm{m})} \rangle$
$\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})}$	$\prod_{\mathrm{d}=1}^{\mathrm{D_m}} \mathcal{N}\left(\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})} \,\vert \langle \mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})} \rangle, \Sigma_{\mathbf{W}^{(\mathrm{m})}} \right)$	$\langle \mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})} \rangle = \langle \tau^{(\mathrm{m})} \rangle \, \mathbf{X}^{(\mathrm{m})^{\mathrm{T}}} \langle \mathbf{Z} \rangle \Sigma_{\mathbf{W}^{(\mathrm{m})}}$ $\Sigma_{\mathbf{W}^{(\mathrm{m})}}^{-1} = \mathrm{diag}(\langle \boldsymbol{\alpha}^{(\mathrm{m})} \rangle) + \langle \tau^{(\mathrm{m})} \rangle \langle \mathbf{Z}^{\mathrm{T}} \, \mathbf{Z} \rangle$
$\alpha_{\mathrm{k}}^{(\mathrm{m})}$	$\Gamma\left(\alpha_{\mathrm{k}}^{(\mathrm{m})} \,\vert a_{\alpha_{\mathrm{k}}^{(\mathrm{m})}}, b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}} \right)$	$a_{\alpha_{\mathrm{k}}^{(\mathrm{m})}} = \frac{\mathrm{D_m}}{2} + a^{\boldsymbol{\alpha}^{(\mathrm{m})}}$ $b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}} = b^{\boldsymbol{\alpha}^{(\mathrm{m})}} + \frac{1}{2} \langle \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}} \, \mathbf{W}^{(\mathrm{m})} \rangle_{k,k}$
$\tau^{(\mathrm{m})}$	$\Gamma\left(\tau^{(\mathrm{m})} \,\vert a_{\tau^{(\mathrm{m})}}, b_{\tau^{(\mathrm{m})}} \right)$	$a_{\tau^{(\mathrm{m})}} = \frac{\mathrm{D_m \, N}}{2} + a^{\tau^{(\mathrm{m})}}$ $b_{\tau^{(\mathrm{m})}} = b^{\tau^{(\mathrm{m})}} + \frac{1}{2} \sum_{\mathrm{n}=1}^{\mathrm{N}} \sum_{\mathrm{d}=1}^{\mathrm{D_m}} \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})^2}$ $- \mathrm{Tr}\{ \langle \mathbf{W}^{(\mathrm{m})} \rangle \langle \mathbf{Z}^{\mathrm{T}} \rangle \, \mathbf{X}^{(\mathrm{m})} \}$ $+ \frac{1}{2} \, \mathrm{Tr}\left\{ \langle \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}} \, \mathbf{W}^{(\mathrm{m})} \rangle \langle \mathbf{Z}^{\mathrm{T}} \, \mathbf{Z} \rangle \right\}$

Table 2.3: Updated q distributions for the different r.v. of the graphical model. These expressions have been obtained using the update rules of the mean field approximation (2.79). We used <> to represent the mean value. See section BIBFA at Sevilla-Salcedo (2021) for further details.

Predictive model

To complete the BIBFA model, in this subsection we will present the predictive formulation. Let's consider we have already obtained in the training step the posterior of the model parameters Θ w.r.t. the training data $\mathcal{D}$, i.e. $q^*(\Theta) \approx p(\Theta|\mathcal{D})$ and we are in the prediction step. Here, we have a test sample $\mathbf{x}_{*,:}$ for which we know its values over the observed views $\mathcal{M}_{\mathrm{in}}$, and we want to compute its predictive distribution in the unobserved view $\mathrm{M}_{\mathrm{out}}$. To obtain $p\left(\mathbf{x}_{*,:}^{\{\mathcal{M}_{\mathrm{out}}\}} \,|\, \mathbf{x}_{*,:}^{\{\mathcal{M}_{\mathrm{in}}\}}, \Theta \right)$, let us start calculating the posterior over the unobserved latent variable $\mathbf{z}_{*,:}$ given the observed test data views and the training data

$$p\left(\mathbf{z}_{*,:} \,|\, \mathbf{x}_{*,:}^{\{\mathcal{M}_{\mathrm{in}}\}}, \mathcal{D} \right) = \int p\left(\mathbf{z}_{*,:} \,|\, \mathbf{x}_{*,:}^{\{\mathcal{M}_{\mathrm{in}}\}}, \Theta \right) p(\Theta|\mathcal{D}) d\Theta. \tag{2.80}$$

If we consider $q^*(\Theta) \approx p(\Theta|\mathcal{D})$, we can solve this equation either using Monte Carlo Integration by sampling from $q^*(\Theta)$, substituting the integral by a summation

$$p\left(\mathbf{z}_{*,:} \,|\, \mathbf{x}_{*,:}^{\{\mathcal{M}_{\mathrm{in}}\}}, \mathcal{D} \right) \approx \frac{1}{\#\Theta} \sum_{\Theta \sim q^*(\Theta)} p\left(\mathbf{z}_{*,:} \,|\, \mathbf{x}_{*,:}^{\{\mathcal{M}_{\mathrm{in}}\}}, \Theta \right), \tag{2.81}$$

where $\#\Theta$ is the number of model parameters, or use a point estimate for Θ from $q^*(\Theta)$, such as the mean value ($\langle\Theta\rangle$)

$$p\Big(\mathbf{z}_{*,:}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}},\mathcal{D}\Big)\;\approx\;p\Big(\mathbf{z}_{*,:}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}},\hat{\Theta}\Big)=p\Big(\mathbf{z}_{*,:}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}},\langle\Theta\rangle\Big). \tag{2.82}$$

Once Θ is fixed, we have that, knowing the prior and likelihood distributions of the latent $\mathbf{z}_{*,:}$ defined in (2.74) and in (2.76), applying Bayes rule we get its posterior

$$p\Big(\mathbf{z}_{*,:}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}},\Theta\Big)\propto p\Big(\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}}\,|\,\mathbf{z}_{*,:},\Theta\Big)p(\mathbf{z}_{*,:}), \tag{2.83}$$

which allows us to define $p\Big(\mathbf{z}_{*,:}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}},\Theta\Big)$ as another Gaussian distribution with mean $\langle\mathbf{z}_{*,:}\rangle$ and covariance $\Sigma_{\mathbf{z}_{*,:}}$ given by

$$\begin{aligned}\Sigma_{\mathbf{z}_{*,:}}^{-1} &= \mathbf{I}_{\text{K}_\text{c}}+\sum_{\text{m}\in\mathcal{M}_{\text{in}}}\Big(\langle\tau^{(\text{m})}\rangle\langle\mathbf{W}^{(\text{m})^{\text{T}}},\mathbf{W}^{(\text{m})}\rangle\Big), \qquad (2.84)\\ \langle\mathbf{z}_{*,:}\rangle &= \sum_{\text{m}\in\mathcal{M}_{\text{in}}}\Big(\langle\tau^{(\text{m})}\rangle\,\mathbf{x}_{*,:}^{(\text{m})}\langle\mathbf{W}^{(\text{m})}\rangle\Big)\Sigma_{\mathbf{z}_{*,:}}.\end{aligned}$$

Once the posterior over the latent projection is calculated, we can now determine the posterior distribution of the unobserved test data views as

$$p\Big(\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{out}}\}}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}},\Theta\Big)\;=\;\prod_{\text{m}_{\text{out}}\in\mathcal{M}_{\text{out}}}p\Big(\mathbf{x}_{*,:}^{(\text{m}_{\text{out}})}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}},\Theta\Big), \tag{2.85}$$

having

$$p\Big(\mathbf{x}_{*,:}^{(\text{m}_{\text{out}})}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}},\Theta\Big)=\int p\Big(\mathbf{x}_{*,:}^{(\text{m}_{\text{out}})}\,|\,\mathbf{z}_{*,:},\Theta\Big)p\Big(\mathbf{z}_{*,:}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}},\Theta\Big)d\,\mathbf{z}_{*,:} \tag{2.86}$$

where $p\Big(\mathbf{x}_{*,:}^{(\text{m}_{\text{out}})}\,|\,\mathbf{z}_{*,:},\Theta\Big)$ is defined in (2.76). Using again the Gaussian properties we can obtain the test samples prediction over an unknown view $\text{m}_{\text{out}}\in\text{M}_{\text{out}}$ as a Gaussian distribution with mean and variance given by

$$\Sigma_{\mathbf{x}_{*,:}^{(\text{m}_{\text{out}})}} = \langle\tau^{(\text{m}_{\text{out}})^{-1}}\rangle\,\mathbf{I}_{\text{D}_\text{m}}+\langle\mathbf{W}^{(\text{m}_{\text{out}})}\rangle\Sigma_{\mathbf{z}_{*,:}}\langle\mathbf{W}^{(\text{m}_{\text{out}})^{\text{T}}}\rangle, \tag{2.87}$$

$$\langle\mathbf{x}_{*,:}^{(\text{m}_{\text{out}})}\rangle = \langle\mathbf{z}_{*,:}\rangle\langle\mathbf{W}^{(\text{m}_{\text{out}})^{\text{T}}}\rangle. \tag{2.88}$$

With these equations, the BIBFA formulation is complete, having a model capable of combining real data in different views, extracting latent factors and calculating predictions on new test data. However, this formulation is limited to specific scenarios and lacks versatility and functionalities. In Chapter 4 we will introduce different extensions of this formulation which combine the existing functionalities with semi-supervised learning, heterogeneous data, kernel representation and additional sparsity constraints, in a way that allows us to use this extended formulation effectively in neuroimaging problems.

Chapter 3

Regularised Bagged Canonical Correlation Analysis

The aim of this chapter is to present an adaptation of the classic MVA formulation (see Section 2.1.3) to include FS capabilities by means of a bagging process and to later apply a guided regularised FE. A first version of this approach was initially presented in Muñoz-Romero et al. (2017), however, we have here adapted this formulation to be efficiently applied to multiclass neuroimaging problems. For this purpose, we have introduced the following extensions:

1) **Class-wise FS**: due to the nature of the analysed neuroimaging problems with low sample population on the most critical classes, we need to adapt the bagging based FS to do a class-wise selection which also improves the interpretability of the results. Furthermore, the inclusion of the class-wise FS also improves the performance of the proposed model in multiclass classification problems, as we will justify below.

2) **Hypothesis test for FS**: we have endowed the method with a statistical test to automatically select the number of relevant features. Thus, the model considerably reduces its computational burden, as we avoid the need for cross-validation (CV) of the number of selected features. The fact that neuroimaging problems are high dimensional entail that the validating the optimum number of selected features is even more computationally expensive, so eliminating this CV greatly streamlines the problem.

3) **Regularisation in the dual space/Guided regularisation**: we improve the guided FE by combining both the magnitude and sign consistency of the projection matrix into a relevance regularisation term, instead of only using the magnitude. This implies that, the relevance learnt for each feature will also be used to regularise them, which in neuroimaging problems allows the model to further consider the effect each feature have in the classification problem.

4) **Balancing**: we extended the formulation to take into account possible class imbalance, having considerably low populated classes with respect to the rest of the classes. In particular, we adapt both the bagging subsampling and the later guided FE to alleviate the class imbalance which is specially present in multiclass neuroimaging scenarios where, typically, there is a larger number of control patients than in an intermediate stage of the disease.

Since we are going to deal with multiclass classification problems, we will focus on a two view CCA formulation. In order to simplify the notation for this scenario, let us denote $\mathbf{X}^{(1)} = \mathbf{X} \in \mathbb{R}^{N \times \mathrm{D}_1}$ as the input data views, and let $\mathbf{X}^{(2)} = \mathbf{Y}$, of dimensions $N \times \mathrm{D}_2$, be the output labels, where $\mathrm{y}_{\mathrm{n,c}} = 1$ when $\mathbf{x}_{\mathrm{n},:}$ belongs to class c and $\mathrm{y}_{\mathrm{n,c}} = 0$ otherwise. Equivalently, we will refer to the projection matrix of each view as $\mathbf{W}^{(1)} = \mathbf{U}$ and $\mathbf{W}^{(2)} = \mathbf{W}$, respectively. The formulation and results here presented are published in Sevilla-Salcedo et al. (2020a).

3.1 Methodology

This section presents the Regularised Bagged - CCA (RB-CCA) method. As shown in the diagram of Figure 3.1, the method consists of two main steps:

1. **FS** process. This first step combines a standard CCA with a bagging procedure to obtain a subset of selected voxels together with a measure of the relevance for each selected feature.
2. **FE for summary components design** to characterise each sample. This second step reduces the input set of selected features to a subset of summary or latent components but, unlike standard CCA, it is based on a regularised version of CCA, guided by the variable relevance obtained from the previous step.

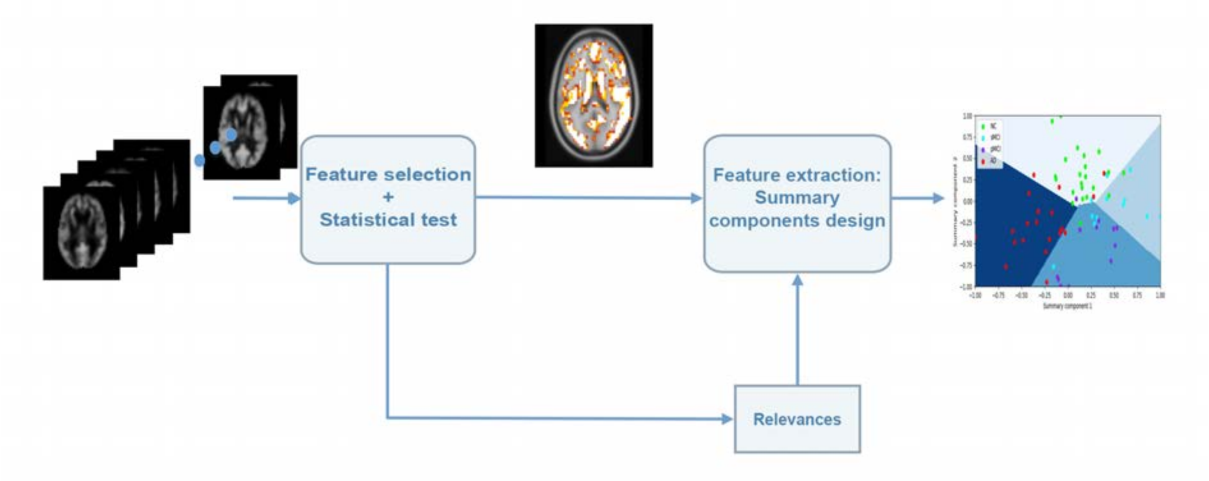


Figure 3.1: RB-CCA scheme for summary components design.

3.1.1 Bagged CCA for feature selection

As previously stated, the goal of the CCA algorithm is to find a low-dimensional projection space where the correlation between two set of data views is maximised. For CCA we can obtain the projection of the input view data by computing

$$\mathbf{z}_{\mathrm{n},:} \quad = \quad \mathbf{x}_{\mathrm{n},:}\mathbf{U} \qquad n = 1, \ldots, N,$$

where $\mathbf{z}_{\mathrm{n},:} \in \mathbb{R}^{\mathrm{K_c}}$ and $\mathrm{K_c}$ is the number of latent factors found by CCA. At this point we could consider analysing matrix $\mathbf{U} \in \mathbb{R}^{\mathrm{D}_1 \times \mathrm{K_c}}$ to use the magnitude of these weights to measure the relevance of each feature and, consequently, generate a FS. Nevertheless, this possible solution could lead to overfitting problems, mainly, when working with high dimensional data (Bi et al., 2003), but this problem could be fixed by the inclusion of a bagging procedure (Breiman, 2001; Parrado-Hernández et al., 2014).

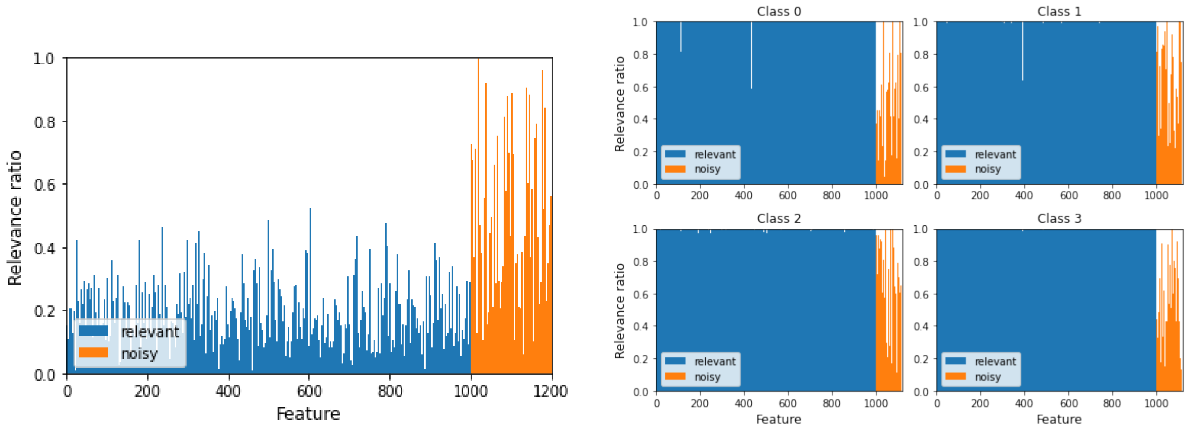
Following this intuition, we propose to train a set of P bagged CCA, where each CCA is trained with a randomly subsampled input data, $\mathbf{X}_{\mathrm{B}^p,:}$ where B^p corresponds to the subset of samples for iteration p. The result will be a set of P projection matrices $\{\mathbf{U}^1, \dots \mathbf{U}^p\}_{p=1}^{\mathrm{P}}$. In order to accelerate the training process, the projection matrix in the dual space, $\mathbf{A}$, can be precalculated before the bagging iterations, as proposed in Muñoz-Romero et al. (2017). This approach is fast, needing just to iterate a single matrix-product, which is a low cost operation, and allowing us to work with the dual formulation. This can also be seen as having features that have random relevance in the different bagging iterations and features that are mostly relevant independently of the number of repetitions.

One problem the original bagging approach has in high dimensional datasets ($\mathrm{D}_2 >> \mathrm{N}$) with high relevance between input features and output classes is that in multiclass classification problems, the projection matrix is calculated using the change of sign of the variables independently of the correlation they have to each class. Hence, one can find problems where the sign influence of a variable for one class has a similar value to another class in the opposite direction, therefore negating the influence of both classes and obtaining a relevance measure that does not fit the real problem. We simulated a 4 class classification toy problem with 1.000 relevant features and 200 noisy ones and calculated the projection vector for each feature over the bagging iterations. The results, in Figure 3.2, show that the original formulation in this scenario, Figure 3.2(a), is not capable of finding the correct relevance to the input features, for this reason we propose using a class-wise bagging. Here, we propose to carry out a stratified class-wise subsampling, randomly subsampling the input data from the subsets of class-wise samples, $\mathbf{X}_{\mathrm{B}^p_c,:}$ with $c = 1, \dots, \mathrm{D}_2$, obtaining D_2 sets of projection matrices for each p bagging iteration, $\{\mathbf{U}_c^1, \dots \mathbf{U}_c^p\}_{p=1}^{\mathrm{P}}$. Then, by calculating the product of both subsampled matrices $\mathbf{U}_c^p = \mathbf{X}^{\mathrm{T}}_{\mathrm{M}_c^p}\mathbf{A}_{\mathrm{M}_c^p}$, with $c = 1, \dots, \mathrm{D}_2$ and $p = 1, \dots, \mathrm{P}$, we can obtain the projection matrix for each class. Figure 3.2(b) shows the results on the toy problem with this proposed configuration, which provides a more adequate relevance for each of these features. With the inclusion of this change in the bagging iterations, the FS based on the information learnt through bagging can work class-wise, making an efficient selection of the relevant features and adding more interpretable information to each feature. Figure 3.3 depicts the scheme of the proposed approach.

Once the bagging is completed and all projection matrices $\mathbf{U}_c^p$ with $c = 1, \dots, \mathrm{D}_2$ and $p = 1, \dots, \mathrm{P}$ have been calculated, we can locate the relevant features through their sign consistency. This will indicate the impact of each input feature on each eigenvector over each class as:

$$\mathbf{\Phi}_c = \frac{1}{\mathrm{P}} \sum_{p=1}^{\mathrm{P}} \mathbb{1}(\mathbf{U}_c^p > 0), \tag{3.1}$$

where $\mathbb{1}(T > 0)$ is the indicator function, which transforms the matrix into a binary matrix, assigning values of 1 to the elements of $\mathbf{U}_c^p$ that are positive and values of 0 to the elements that are negative. The resulting matrix $\mathbf{\Phi}_c \in \mathbb{R}^{\mathrm{D}_1 \times \mathrm{K}_c}$ assigns low values for features that are not sign consistent throughout the bagging iterations and high values for those which are sign consistent. Bi et al. (2003) proved that using sign consistency as a measure of relevance leads



(a) Standard bagging.

(b) Class-wise bagging.

Figure 3.2: Analysis of the learnt feature relevance based on the sign consistency using bagging. The results were obtained using a 4 class classification toy problem where most features are relevant for the task and some are random noisy variables. Here we see how having class-wise bagging might be helpful for some multi-class classification scenarios.

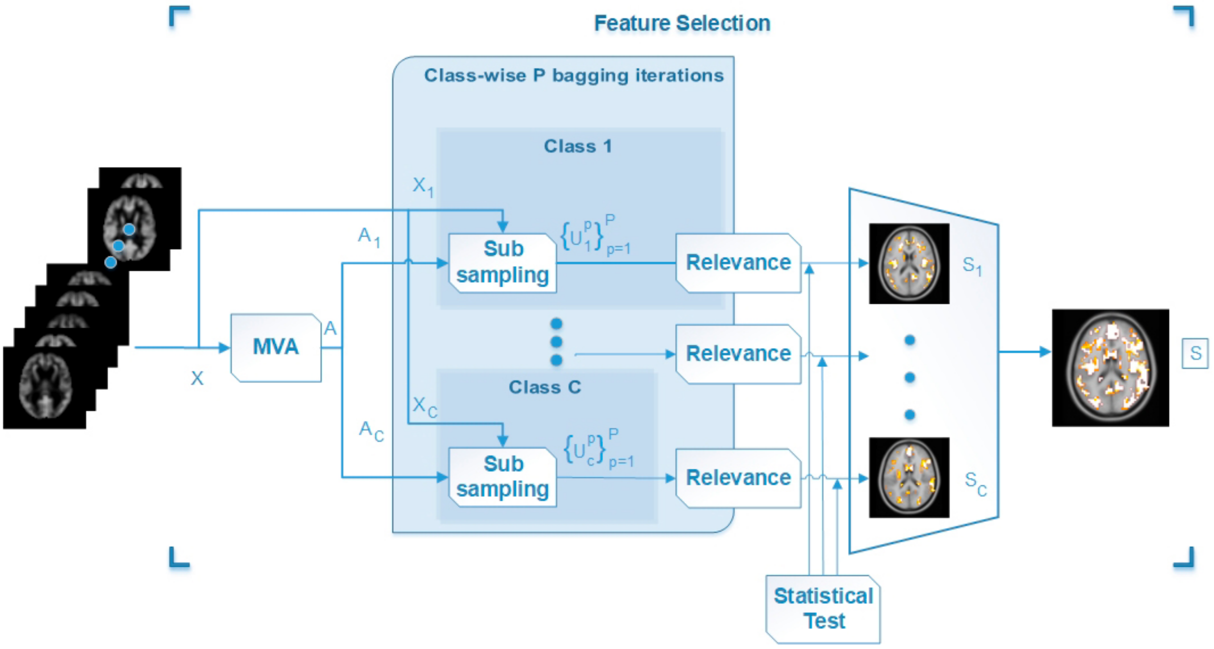


Figure 3.3: Class-wise FS scheme for parsimonious CCA.

to finding features that are relevant regardless of the used samples, and consequently generalise better for the classification problem than using the weight magnitude. Thus, we can convert this matrix containing the relevance for each latent factor, feature and class into a new matrix which combines the relevance across all latent factors by calculating their average value:

$$\mathrm{b}_{\mathrm{d,c}} = \frac{1}{\mathrm{K_c}} \sum_{k=1}^{\mathrm{K_c}} |2\,\phi_{\mathrm{d,c,k}} - 1| \quad d = 1, \dots, \mathrm{D}_1; \quad c = 1, \dots, \mathrm{D}_2\,. \tag{3.2}$$

Note that $\mathrm{b}_{\mathrm{d,c}}$ is normalised, so values close to 1 relate to highly consistent features for that class and values close to 0 to non-consistent features, thus, not relevant for that class. Sorting the D_2 $\mathbf{b}_{:,c}$ arrays we have a class-wise relevance measure of the input features that can be used to only use a subset of the most relevant input features to train the classifier. This selection of the number of relevant features can be carried out by CV to adjust either the percentage of relevant features or a threshold over the class-wise sign consistency $\mathbf{b}_{:,c}$. However, this needs to be done by either averaging the class-wise results in (3.2), losing the class-wise FS, setting a common threshold for every class, which might not be adequate, or by carrying out one CV for each class, which is time consuming. For this reason, in the next subsection we propose a hypothesis test to automatically set the number of selected features, which eliminates the need for CV the number of the number of selected features and, consequently, considerably reduces the computational cost.

3.1.2 Hypothesis test for feature selection

In this section we propose a statistical test that can be combined with the relevance $\mathbf{B}$ (3.2), learnt through the bagging process, to automatically select the most relevant features for the problem. This solution is included as an alternative to the CV of the number of selected features and allows to easily obtain class-wise FS.

To define a statistical test over the sign consistency, we consider that a feature is irrelevant when it has the same probability of being either sign over P bagging iterations. Hence, on the basis of the sign consistency analysis in (3.1), we can establish that a variable d is irrelevant for class c and eigenvector k if its associated success probability $\phi_{\mathrm{d,c,k}}$ is equal to 0.5. Then, we can formulate the following hypothesis test:

$$\begin{cases} H_0 : \phi_{\mathrm{d,c,k}} & = & 0.5,\ d \text{ is not relevant for } c-\text{th class and } k-\text{th eigenvector.} \\ H_1 : \phi_{\mathrm{d,c,k}} & \neq & 0.5,\ d \text{ is relevant for } c-\text{th class and } k-\text{th eigenvector.} \end{cases} \tag{3.3}$$

Taking these hypothesis into consideration and following the intuition behind the t-test, we can statically evaluate if $\phi_{\mathrm{d,c,k}}$ differs from 0.5 by defining the following statistic:

$$\mathrm{t}_{\mathrm{d,c,k}} = \frac{\phi_{\mathrm{d,c,k}} - 0.5}{\sigma_{d,c,k}}, \tag{3.4}$$

where $\sigma_{d,c,k}$ is a scaling factor proportional to the standard deviation of $\phi_{\mathrm{d,c,k}}$.

To derive this scaling factor, let us consider that $\sum_{p=1}^{\mathrm{P}} \mathbb{1}(\mathrm{u}_{\mathrm{d,c,k}}^{\mathrm{p}} > 0)$ determines the number of times a feature is positive over P bagging iterations, if we assume that the iterations are independent, $\sum_{p=1}^{\mathrm{P}} \mathbb{1}(\mathrm{u}_{\mathrm{d,c,k}}^{\mathrm{p}} > 0)$ can be modelled as a binomial distribution with parameters P (number of experiments) and $\phi_{\mathrm{d,c,k}}$ (success probability). Incidentally, considering the number of iterations P, which is very large, we can approximate this binomial distribution by a normal distribution with mean $\mathrm{P} \cdot \phi_{\mathrm{d,c,k}}$ and standard deviation $\mathrm{P} \cdot \phi_{\mathrm{d,c,k}}(1 - \phi_{\mathrm{d,c,k}})$. Taking this into account, we can define $\sigma_{d,c,k}$ as the standard deviation of the term $\frac{1}{\mathrm{P}} \sum_{p=1}^{\mathrm{P}} \mathbb{1}(\mathrm{u}_{\mathrm{d,c,k}}^{\mathrm{p}} > 0)$, which is straightforwardly computed as the square root of the rescaled variance of the normal distribution:

$$\sigma_{d,c,k} = \sqrt{\frac{1}{\mathrm{P}} \cdot \phi_{\mathrm{d,c,k}}(1 - \phi_{\mathrm{d,c,k}})}. \tag{3.5}$$

Nonetheless, in the context of the bagging process, the assumption of independence between bagging iterations is not adequate. To tackle this situation, we can compute the standard deviation with an unbiased estimator (Nadeau and Bengio, 2000) which applied to (3.5) provides the following corrected estimator for the standard deviation:

$$\begin{aligned}\tilde{\sigma}^{\text{corr}}_{d,c,k} &= \sqrt{\frac{1}{\mathrm{P}}\left(1+\mathrm{P}\frac{M}{1-M}\right)\phi_{\mathrm{d,c,k}}(1-\phi_{\mathrm{d,c,k}})} \\ &\simeq \sqrt{\frac{M}{1-M}\phi_{\mathrm{d,c,k}}(1-\phi_{\mathrm{d,c,k}})} \qquad (3.6)\end{aligned}$$

and, accordingly, the statistic $\mathrm{t}_{\mathrm{d,c,k}}$ defined in equation (3.4) becomes

$$\mathrm{t}_{\mathrm{d,c,k}} = \frac{\phi_{\mathrm{d,c,k}} - 0.5}{\sqrt{\frac{M}{1-M}\phi_{\mathrm{d,c,k}}(1-\phi_{\mathrm{d,c,k}})}}. \qquad (3.7)$$

as $\phi_{\mathrm{d,c,k}}$ can be approximated with a normal distribution, this final statistic is distributed according to a t-distribution with $\mathrm{P}-1$ degrees of freedom. Nevertheless, since P is very large, we can simplify this calculation even more. One can safely approximate the t-distribution by the standard normal distribution. This way, under the null hypothesis, the statistic $\mathrm{t}_{\mathrm{d,c,k}}$ follows a normal distribution with zero mean and unit standard deviation. Therefore, we can apply the test by selecting the features that correspond to the tails of the normal distribution.

Once the statistic $\mathrm{t}_{\mathrm{d,c,k}}$ is computed, we propose to determine a class-wise FS by majority vote of the k-th parameter. This way, the selection takes into account whether a feature is relevant for most eigenvectors or just for some of them. Furthermore, the statistical test makes computationally efficient to have the class-wise FS in place of CV the threshold over each class. We obtain a set of features corresponding to each class $\mathbf{S}_c$ with $c = 1, \ldots, \mathrm{D}_2$, that determine which features are relevant for each class. This provides a more in depth insight on the problem features, as well as the correlation between them and the different output classes. However, as we might need to use this selection as input for another model, such as a classifier, it would be necessary to have a single input matrix with all the selected features. To do so, we can define a set of indexes $\mathbf{S} = [\mathbf{S}_1, \ldots, \mathbf{S}_{\mathrm{D}_2}]$ as the union of the class-wise subsets, $\mathbf{S}_c$, in order to have a compact selection. We can use the data matrix $\mathbf{X}_{:,\mathbf{S}}$ as input for CCA or the classifier, being the original input data with only the relevant variables (by indexing column-wise).

By defining the statistical test, we grant the model more versatile utilities, such as the ability to have class-wise selection due to the computational improvement. Furthermore, this efficient solution reduces the number of hyperparameters in the model, while providing a more informative selection.

3.1.3 Guided feature extraction

So far, we have defined the FS stage of the RB-CCA model (see Figure 3.1). Here, we present an adaptation of the CCA formulation to add a regularisation term to guide the construction of the latent space. Let us now retake the projection matrices learnt though the bagging process Making use of the $\mathbf{U}_c^{\mathrm{P}}$ projection matrices we can define two different relevance measurements:

- The **sign consistency** of the eigenvectors. Equivalently to the FS, we can use the variability of sign of each variable to measure the usefulness of each feature:

$$\bar{\mathrm{b}}_{\mathrm{d}} = \max_{c}\{\mathrm{b}_{\mathrm{d,c}}\}. \qquad (3.8)$$

- The associated **magnitude** of the eigenvectors. Furthermore, we can consider the weights associated to the variables, where higher weights are associated to more relevant features while lower ones relate to less relevant. This way, we can define an additional relevance criteria with the magnitude of the eigenvectors:

$$\bar{\mathrm{u}}_{\mathrm{d}} = \left\| \max_{c} \left\{ \frac{1}{\mathrm{P}} \sum_{p=1}^{\mathrm{P}} \left| \mathrm{u}^{\mathrm{p}}_{\mathrm{d,c,k}} \right| \right\} \right\|_{2}. \tag{3.9}$$

The overall relevance for both measures corresponds to the maximum of the relevance per class, so if a feature is relevant for one class, it will also be relevant for these metrics. And, conversely, features that are not relevant for any class will not be relevant for these measures either. Combining both terms we can define our relevance score as:

$$\omega_{\mathrm{d}} = \frac{1}{\bar{\mathrm{b}}_{\mathrm{d}}\, \bar{\mathrm{u}}_{\mathrm{d}}}. \tag{3.10}$$

Note that this measure of relevance differs from the one proposed in Muñoz-Romero et al. (2017), which only uses the magnitude as relevance. We propose to combine the relevance with the learnt sign consistency to have a more cohesive result that doesn't fully rely in either measure and combines both for a stronger regularisation term. Besides, this is critical in high dimensional problems ($\mathrm{D}_1 >> \mathrm{N}$) with high feature redundancy, as the magnitude influence scatters over the features, while the sign consistency is conserved.

Now, we can use this feature relevance to guide the FE step by means of a regularisation term. This way, the FE process will not only operate on the subset of selected variables, but will also include their importance for learning the low-dimensional representation.

For this purpose, we can calculate the inverse of the importance in (3.10) to make the regularisation focus on assigning lower penalties to more relevant features and higher penalties to less relevant ones; thus, we can directly guide the influence of these variables over the projected data, $\mathbf{Z}$.

So, at this point, along with the reduced subset of input variables learnt through FS $\mathbf{X}_{:,\mathbf{S}}$, we can include the proposed regularisation term over the dual CCA formulation[3] in (2.36) having a regularisation over the primal variables while we continue solving the dual problem:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{A}} \quad & \left\| (\mathbf{Y} - \mathbf{K}_{\mathrm{XS}}\, \mathbf{A}\, \mathbf{W}^{\mathrm{T}})\, \mathbf{C}_{\mathbf{YY}}^{-1/2} \right\|_F^2 + \lambda \left\| \boldsymbol{\Omega}^{1/2} \mathbf{X}_{:,\mathbf{S}}^{\mathrm{T}}\, \mathbf{A} \right\|_F^2, \\ \text{s.t.} \quad & \mathbf{W}^{\mathrm{T}}\, \mathbf{C}_{\mathbf{YY}}^{-1}\, \mathbf{W} = \mathbf{I}_{\mathrm{K_c}}, \end{aligned} \tag{3.11}$$

where $\mathbf{C}_{\mathbf{X}^{(2)}\mathbf{X}^{(2)}} = \mathbf{C}_{\mathbf{YY}}$, since we considered $\mathbf{X}^{(2)} = \mathbf{Y}$, $\mathbf{K}_{\mathrm{XS}} = \mathbf{X}_{:,\mathbf{S}}\, \mathbf{X}_{:,\mathbf{S}}^{\mathrm{T}}$ is the linear kernel matrix of the selected data, $\boldsymbol{\Omega}$ is a diagonal matrix of the relevance measure values defined in (3.10) and λ is a regularisation parameter.

Now, defining $\mathbf{V}$ as $\mathbf{C}_{\mathbf{YY}}^{-1/2}\, \mathbf{W}$ and applying Lagrange multipliers, we derive the equation with respect to $\mathbf{V}$ and equate it to zero. Hence, we can firstly obtain the optimum value of $\mathbf{V}$ by

[3] Even though the input variables are reduced at this point by the FS, the selected features might still be more than N. For this reason, it is advisable to work with the dual formulation in a similar way to that of Section 2.1.3. This is specially critical in neuroimaging problems, where it is common to have more features than samples despite the FS.

solving the following eigenvalue problem

$$\mathbf{C}_{\mathbf{YY}}^{-1/2}\,\mathbf{Y}^{\mathrm{T}}\mathbf{K}_{\mathrm{x_S}}(\mathbf{K}_{\mathrm{x_S}}\mathbf{K}_{\mathrm{x_S}} + \lambda\mathbf{X}_{:,\mathbf{S}}\mathbf{\Omega}\mathbf{X}_{:,\mathbf{S}}^{\mathrm{T}})^{-1}\mathbf{K}_{\mathrm{x_S}}\mathbf{Y}\,\mathbf{C}_{\mathbf{YY}}^{-1/2}\,\mathbf{V} = \mathbf{V}\mathbf{\Sigma}, \tag{3.12}$$

to later compute the dual space projection matrix $\mathbf{A}$ as

$$\mathbf{A} \;=\; (\mathbf{K}_{\mathrm{x_S}}\mathbf{K}_{\mathrm{x_S}} + \lambda\mathbf{X}_{:,\mathbf{S}}\mathbf{\Omega}\mathbf{X}_{:,\mathbf{S}}^{\mathrm{T}})^{-1}\mathbf{K}_{\mathrm{x_S}}\mathbf{Y}\,\mathbf{C}_{\mathbf{YY}}^{-1/2}\,\mathbf{V}. \tag{3.13}$$

3.1.4 Balanced regularised CCA

Here we propose a modification of the previous formulation to work with imbalanced databases. The reason for this extension relies on the fact that neuroimaging datasets do not usually have enough representative samples of some particular classes, which leads to biased results to the most populated classes. This way, in order to tackle this situation, we can modify the previous formulation to increase the weight of the samples related to the less populated classes.

The RB-CCA model has been adapted to balance the results in its two distinct blocks. First of all, for the FS step the balancing is easily carried out by subsampling the same number of samples from each class regardless of the class population during the bagging. Secondly, for the FE step we decided to include the class balancing defining a balanced version of the Frobenius norm

$$\|\mathbf{D}\|_{F_{\mathbf{\Theta}}}^{2} \;=\; \mathrm{Tr}\{\mathbf{D}^{\mathrm{T}}\mathbf{\Theta}\mathbf{D}\}, \tag{3.14}$$

where $\mathbf{\Theta}$ is a diagonal matrix of dimensions N × N, where each diagonal element is N/N_c, being N_c the number of samples of class c. By the inclusion of this matrix, we can include weights inversely proportional to the frequencies of each class in $\mathbf{Y}$ to balance the results. Therefore, including this new Frobenius form in Equation (3.11), we have

$$\begin{aligned} \min_{\mathbf{W},\mathbf{A}} \quad & \left\|(\mathbf{Y} - \mathbf{K}_{\mathrm{x_S}}\,\mathbf{A}\,\mathbf{W}^{\mathrm{T}})\,\mathbf{C}_{\mathbf{YY}}^{-1/2}\right\|_{F_{\mathbf{\Theta}}}^{2} + \lambda\left\|\mathbf{\Omega}^{1/2}\mathbf{X}_{:,\mathbf{S}}^{\mathrm{T}}\,\mathbf{A}\right\|_{F}^{2}, \\ \text{s.t.} \quad & \mathbf{W}^{\mathrm{T}}\,\mathbf{C}_{\mathbf{YY}}^{-1}\,\mathbf{W} = \mathbf{I}_{\mathrm{K}_c}, \end{aligned} \tag{3.15}$$

which solution is now given by the following eigenvalue problem

$$\mathbf{C}_{\mathbf{YY}}^{-1/2}\,\mathbf{Y}^{\mathrm{T}}\Theta\mathbf{K}_{\mathrm{x_S}}(\mathbf{K}_{\mathrm{x_S}}\Theta\mathbf{K}_{\mathrm{x_S}} + \lambda_2\mathbf{X}_{:,\mathbf{S}}\mathbf{\Omega}\mathbf{X}_{:,\mathbf{S}}^{\mathrm{T}})^{-1}\mathbf{K}_{\mathrm{x_S}}\Theta\mathbf{Y}\,\mathbf{C}_{\mathbf{YY}}^{-1/2}\,\mathbf{V} \;=\; \mathbf{V}\mathbf{\Sigma}, \tag{3.16}$$

from which we obtain $\mathbf{V}$, to later calculate the dual matrix $\mathbf{A}$ as

$$\mathbf{A} \;=\; (\mathbf{K}_{\mathrm{x_S}}\Theta\mathbf{K}_{\mathrm{x_S}} + \lambda_2\mathbf{X}_{:,\mathbf{S}}\mathbf{\Omega}\mathbf{X}_{:,\mathbf{S}}^{\mathrm{T}})^{-1}\mathbf{K}_{\mathrm{x_S}}\Theta\mathbf{Y}\,\mathbf{C}_{\mathbf{YY}}^{-1/2}\,\mathbf{V}. \tag{3.17}$$

3.2 Results

The objective of this section is to provide an insight on the model performance in the characterisation of mental diseases. Specifically, we will study RB-CCA over two distinct databases for mental disease prediction and characterisation. The first dataset focuses on the detection of different stages of Alzheimer's disease. The second consist in the classification of Attention Deficit Hyperactivity Disorder (ADHD), which we will analyse in terms of the classification results together with the information learnt by the model, i.e. the FS and the FE. This analysis will provide us an insight on the relevance learnt for each input feature as well as their influence on the construction of the projected data for the classification problem.

An exemplary notebook along with the complete code of the proposed method is available at regMVA.

This section is structured in four different subsections. First, we will describe the two databases we will analyse in this chapter, as well as the pre-processing carried out. Next, we will present the different state-of-the-art methods that we will use as baselines to compare the results obtained. We will then indicate the validation procedure and the different set-ups used for each method under study. Finally, we will present the results obtained in the different experiments: analysis of the performance of the proposed algorithm, comparison between the different extensions of the method, and additional knowledge provided by both the learnt FS and FE steps.

3.2.1 Database description

To evaluate the performance and functionalities of the proposed model, we decided to use two well-known neuroimaging databases: the Alzheimer's Disease Neuroimaging Initiative (ADNI) that analyses Alzheimer's disease and ADHD200 that analyses ADHD. Both datasets consist of multiclass classification problems with unbalanced classes.

3.2.1.1 ADNI data

The first data set used in this chapter was obtained from the ADNI database (`adni.loni.usc.edu`). ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI was to test whether serial Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of Mild Cognitive Impairment (MCI) and early Alzheimer's Disease (AD).

The initial goal for ADNI (ADNI-1) was to recruit 800 subjects, but ADNI has been followed by ADNI-GO and ADNI-2, leading to the recruitment of more than 1,500 adults, aged 55-90 years, to participate in the research. These consist of Cognitively Normal older individuals (NC), people with stable MCI (sMCI) or progressive MCI (pMCI), and people with early AD. The conversion status of the MCI subjects was defined as in Moradi et al. (2015). Briefly, a subject was considered to be in pMCI group if the diagnosis was MCI at the baseline and the subject converted to AD in three years. A subject was considered to be in sMCI group if the diagnosis was MCI at the baseline and the subject did not convert to AD during the follow-up. The duration of monitoring for each group is specified in the ADNI-1, ADNI-2 and ADNI-GO protocols. For up-to-date information, see `www.adni-info.org`.

Data used in this work included MRIs from 200 AD patients, 164 pMCI, 100 sMCI and 231 NCs. The characteristics of these subjects are summarised in Table 3.1. Subjects with less than 3 years of follow-up and subjects whose diagnostic status fluctuated were excluded.

The MRIs are T1-weighted MP-RAGE sequences at 1.5 T, with 256 × 256 × 170 voxels where the voxel size is approximately 1mm × 1mm × 1.2mm. These were preprocessed into grey

Characteristic	AD	pMCI	sMCI	NC
#subjects	200	164	100	231
Age	75.6 ± 7.7	74.57 ± 7.0	75.4 ± 7.2	76.0 ± 5.0
Gender (M/F)	103/97	97/67	66/34	119/112

Table 3.1: Characteristic of the ADNI dataset used in this work.

matter tissue images in the stereostatic space, as described by Gaser et al. (2013), and thereafter smoothed with the 8-mm Full Width at Half Maximum (FWHM) Gaussian kernel, resampled to 4 mm spatial resolution and masked into 29.852 voxels.

Finally, since atrophic regions detected in AD patients overlapped with normal age-related declining regions of healthy control subjects (Dukart et al., 2011), the data were age-corrected by regressing out the age of the subject on a voxel-by-voxel basis (Moradi et al., 2015).

3.2.1.2 ADHD data

We have also studied functional connectivity in ADHD using ADHD200 data (Milham et al., 2012). The data used in this study consists of 973 resting state functional MRI (fMRI) collected at eight independent imaging sites, all from children and adolescents from the ages of 7 to 21.

These resting state fMRIs were preprocessed by the Neuro Bureau, who originally created the database, as described by Bellec et al. (2017).

Characteristic	ADHD-C	ADHD-I	TDC
#of subjects	204	127	555
Age	11.94 ± 2.6	11.3 ± 3.2	12.2 ± 3.5
Gender (M/F)	170/34	92/35	288/267

Table 3.2: Characteristic of the ADHD dataset used in this work

Then, time courses of brain regions corresponding to CC400 atlas were obtained by averaging voxel-wise fMRI intensities within regions. This yielded 351 regional time courses per subject. From the 351 regional time courses, a 351 x 351 correlation matrix describing the strength of the functional connectivity between region pairs was computed. Therefore, 61.425 features per subject were obtained by vectorising the correlation matrix and removing redundant elements. Additionally, the samples that did not pass the quality control of the Neuro Bureau, were removed. After this, data from 922 patients remained (555 Typically Developing, 204 ADHD-Combined, 12 ADHD-Hyperactive/Impulsive, and 127 ADHD-Inattentive). Finally, we also eliminated 12 ADHD-hyperactive/impulsive cases from this study, as the number of subjects in this group was not sufficient for meaningful classification.

3.2.2 Baseline or state-of-art methods

In this section we present the baselines included in the experimental study to be compared with the proposed RB-CCA algorithm. We decided to use linear models because they perform better than non-linear in high dimensional scenarios with a reduced number of samples (Muller et al., 2003). For this reason, we decided to use the following models as baselines:

- SVM: Linear Support Vector Machine (SVM) is a widely used binary classifier that can be integrated into a one-vs.-all configuration for multiclasss classification problems. Besides, we can use its balanced version to assign different weights to the regularisation parameter depending on the number of samples per class.
- Standard CCA with a SVM (CCA+SVM): This second approach consists in the use of standard CCA, described in Section 2.1.2, to find a projection of the data into a low dimensional space to later train a linear SVM classifier.
- SVM significance map with a SVM (p-map+SVM): The p-map+SVM method was presented by Abdulkadir et al. (2014) in the CADDementia challenge (Bron et al., 2015) for multiclass classification approach following a one-vs.-all framework, subsequently providing a class-wise FS. In this approach, the original features are fed into the SVM p-map, which calculates the significance of the features for the SVM classifier by means of a permutation test Ojala and Garriga (2010). With the p-map approach we set a threshold to select the features to be used that are suitable for further classification using a linear SVM.

For the ADHD database, we also decided to include as reference model the Extreme Learning Machine (ELM) approach used by Qureshi et al. (2016) for multiclass classification. We actually used three implementations:

- ELM with all the original features (ELM). ELM are feedforward neural networks with a single hidden layer where the parameters of the network do not need to be tuned. We use them as classifiers for the problem.
- ELM combined with a FS learnt through Random Feature Elimination (ELM+RFE). RFE iteratively eliminates the least relevant feature which provides a lower accuracy, until there is none left, finally obtaining a feature-wise relevance ranking. Hence, we need to cross validate the number of selected features.
- A hierarchical ELM with RFE (HELM+RFE), equivalent to the previous approach but using the Hierarchical version of ELM (HELM). HELM combines two steps: first an ELM-based unsupervised sparse multilayer autoencoder to encode the features and second a supervised ELM used to classify the samples.

To compare these baselines with the proposed approach, we use RB-CCA to select a subset of relevant features and, later, determine the eigen-MRIs/summary components, low dimensional representations of the selected features. However, we also combined the proposed RB-CCA with a linear SVM with class balancing to classify the projected data in order to measure the performance of the method.

3.2.3 Experimental set-up

In order to estimate the model performance and validate the hyperparameters, we decided to calculate the results using a nested 10-fold cross-validation. While the outer CV divides the data into train and test partitions, the inner CV focuses on validation, dividing the training partition into training and validation subsets.

Among the different methods evaluated in this section, there are several parameters that are susceptible to validation. For the proposed RB-CCA we have the bagging subsampling rate, the regularisation parameter, λ, and the number of extracted (latent) features. Besides, for the output SVM we have the parameter C which weights the regularisation term.

To avoid needing to cross validate so many parameters, we have analysed their influence, fixing those that are not critical for the final model performance. After some experiments, we concluded that the subsampling rate barely influences the final performance of the proposed method. For this reason, we set the subsampling rate to 50%. Regarding the number of extracted (latent) features, we decided to set it to the maximum ($\mathrm{D}_2 - 1$) as less features leads to a more oppressive bottle-neck that can deteriorate the performance. On the other hand, we found that the regularisation hyperparameter λ was indeed critical for the performance, so we cross-validated it. To do so, we explored 17 values in logarithmic scale from from $[10^{-4}$ to $10^{3}]$. We initially validated the value of hyperparameter C of the one-vs.-all SVM using all voxels as input, finding that the optimum value was rather small ($C = 0.035$). After evaluating this hyperparameter on the remaining analysed baselines and confirming they provided good performance, we decided to fix it to $C = 0.035$ for all the methods under study. When cross validating the number of selected features, validation accuracy curves tend to have a maximum value when all features are used. Thus, we decided to work with the point of the curve where the accuracy saturation begins, CV Stability Point (CV-SP), obtaining good performance while reducing the number of features.

To evaluate the model performances, we decided to use balanced classification accuracy, mean class-wise accuracy, as the score to compare the performance of the different models as well as to adjust the hyperparameters. By using the balanced version of the accuracy, we are able to fairly evaluate the performance on low-populated classes. Furthermore, we also used the balanced multiclass Area Under the Curve (AUC) calculated as the AUC of class c with respect to the rest of the classes weighted by the class population ratio. This way, we also showed the performance in the class-wise classification as well as the general model performance in the multiclass classification problem.

3.2.4 Performance compared to baseline methods

In this first performance analysis, we compare the performance of the stated baselines with the proposed balanced version of the RB-CCA-ST model (RB-CCA using the statistical test for FS). Here, we only include this version, however a further study of the different versions of the proposed model will be compared in next section.

In Table 3.3, we first include the results in the ADNI database. Here, we present the mean values and the standard deviation over the 10 CV test partitions. These results prove that the proposed approach is capable of outperforming the reference baselines in terms of balanced accuracy and multiclass AUC. Furthermore, the obtained results use only one third of the original features. Comparing it with the p-map + SVM method, which also performs FS, we can see that RB-CCA-ST outperforms it in terms of accuracy while further reducing the number of relevant features with a more consistent selection (note that the standard deviations of the results are clearly lower).

Method	**#feat.**	**Balanced**		**class AUC**			**AUC**
		accuracy	NC	sMCI	pMCI	AD	
SVM	29.852	56,47 ±5,97%	0,892	0,694	0,792	0,860	0,831 ±0,024
CCA + SVM	29.852	52,15 ±5,17%	0,854	0,603	0,751	0,825	0,786 ±0,029
p-map + SVM	19.637 ±6.959	55,98 ±5,95%	0,886	0,694	0,795	0,862	0,830 ±0,024
RB-CCA-ST + SVM	13.222 ±1.004	**62,91** **±4,64%**	**0,915**	**0,766**	**0,830**	**0,882**	**0,864** **±0,024**

Table 3.3: ADNI - Performance results in terms of balanced accuracy and AUC with the proposed method compared with the baselines. The results are averaged over the 10-fold CV test partitions and include the mean and standard deviation over these folds.

Regarding the ADHD database, Table 3.4 presents the results obtained with the baselines as well as the proposed method. Note that this is a complicated problem where some methods perform worse than a random classifier, most of the studied methods barely improve the classification balanced accuracy obtained by chance (33,3% if we randomly assign one of the three classes to a subject). In particular, the proposed method, compared to the ELM variants, is capable of outperforming them both in terms of balanced accuracy and multiclass AUC. If we analyse the performance obtained by the SVM, we see that, although the mean accuracy seems to be better than RB-CCA-ST, there are not sufficient statistical difference (SVM has considerably higher standard deviation). Furthermore, RB-CCA-ST outperforms SVM in terms of AUC, so both models tend to provide similar classification performances. Nevertheless, the main advantages obtained with this database are in terms of interpretability, since RB-CCA-ST is capable of reducing the original input features to one fifth from the original and extracting only two features to codify the data while maintaining a good classification score.

3.2.5 Analysis of the different versions of RB-CCA

Given that the proposed model combines different strategies, the FS approach and the FE method, here we will analyse the effect of each step. Concretely, we will combine the different strategies of the algorithm to understand their impact on the final model performance. So, on the one hand, the analysed FS techniques are:

Method	**#feat.**	**Balanced accuracy**	**class AUC**			**AUC**
			TDC	ADHD-I	ADHD-C	
SVM	61.425	**39,40 ±14,66%**	0,592	0,502	**0,649**	0,582 ±0,057
CCA + SVM	61.425	36,94 ±8,40%	0,597	0,500	0,632	0,581 ±0,053
p-map + SVM	969 ±61	36,54 ±6,33%	0,568	0,484	0,635	0,568 ±0,032
ELM	61.425	24,26 ±10,05%	0,533	0,504	0,567	0,537 ±0,065
ELM + RFE	7.025 ±2.376	22,64 ±5,26%	0,507	0,511	0,534	0,514 ±0,060
HELM + RFE	7.025 ±2.376	28,99 ±8,56%	0,525	0,505	0,564	0,531 ±0,057
RB-CCA-ST + SVM	18.295 ±4.393	**38,48 ±8,18%**	**0,600**	**0,530**	0,644	**0,600 ±0,058**

Table 3.4: ADHD - Performance results in terms of balanced accuracy and AUC with the proposed method compared with the reference models. The results are averaged over the 10 fold CV test partitions and include the mean and standard deviation over these folds. Note that, in this case, the balanced accuracy value obtained by chance would be 33,3%.

- No FS: uses the original features as input for either the feature extractor or the output classifier.

- Bagged CCA with CV to choose the number of selected features (BagCCA-CV): uses the feature relevance (3.10) determined through bagging combined with CV to determine the number of features to use as input for either the feature extractor or the classifier. Note that, we use the CV Stability Point (CV-SP) as the criteria to determine the number of features to select.

- Bagged CCA with the statistical test (BagCCA-ST): uses the feature relevance determined through bagging combined with the proposed statistical test to determine which features are used as input for either the feature extractor or the final classifier.

On the other hand, the explored FE methods are:

- No FE: there is no projection to a latent space. The input features are then used for the final classifier.

- Standard CCA: uses the classic CCA formulation to project the features to a low dimensional latent space.

- Regularised CCA: uses the proposed regularised version of CCA to project the features to a low dimensional latent space.

As in the previous experiments, we will always use a balanced linear SVM to classify the output of the different experiments. We used balanced version of RB-CCA to calculate these results.

FE method	**FS method**	**#feat.**	**Balanced**		**class AUC**			**AUC**
			accuracy	NC	sMCI	pMCI	AD	
No	None	29.852	56, 47 ±5, 97%	0,892	0,694	0,792	0,860	0, 831 ±0, 024
	BagCCA-CV	11.642 ±5.373	60, 55 ±6, 48%	0,908	0,737	0,825	0,883	0, 857 ±0, 026
	BagCCA-ST	13.222 ±1.004	61, 90 ±9, 65%	0,912	0,739	0,835	0,884	0, 861 ±0, 024
Standard CCA	None	29.852	52, 09 ±5, 16%	0,854	0,593	0,747	0,825	0, 783 ±0, 030
	BagCCA-CV	15.075 ±5.602	52, 12 ±5, 89%	0,850	0,598	0,745	0,831	0, 783 ±0, 030
	BagCCA-ST	13.222 ±1.004	52, 07 ±5, 95%	0,856	0,635	0,754	0,828	0, 792 ±0, 028
Regularised CCA	None	29.852	58, 19 ±6, 42%	0,911	0,718	0,816	0,870	0, 849 ±0, 019
	BagCCA-CV	6.862 ±5.135	62, 04 ±5, 50%	0,909	**0,785**	**0,835**	**0,887**	**0, 868** **±0, 025**
	BagCCA-ST	13.222 ±1.004	**62, 91** **±4, 64%**	**0,915**	0,766	0,830	0,882	0, 864 ±0, 024

Table 3.5: ADNI - Performance analysis of the different versions of the proposed method. This table shows the performance in terms of multiclass AUC and multiclass balanced accuracy for different combinations of FE and FS methods. The results are given with the mean and standard deviation over the test 10-fold CV.

Firstly, Table 3.5 contains the results using the different combinations of the proposed model functionalities with the ADNI database. Regarding FS, we can see that its inclusion provides an improvement in the performance in all the analysed scenarios, having gains of over 0, 03 in terms of AUC in the case without FE. Regarding the type of FS, the statistical test proves to be able to either outperform or get similar results to CV in terms of accuracy and AUC. Furthermore, these results stand out considering the computational cost of cross validating the number of selected features in comparison to calculating this through a statistical test. If we consider the FE, we can see that the inclusion of this feature is the most beneficial strategy for the problem only when it is combined with the proposed regularisation term. This can be seen comparing the results obtained with FE without regularisation (AUC is around 0,78-0,79) and with the proposed regularisation, where AUC reaches values around 0,864.

Secondly, Table 3.6 includes the equivalent results obtained with the ADHD database. The results imply that the inclusion of FS improves the performance of the model. Between the two proposed selection methods, the statistical test proves to be more reliable as it eliminates one CV loop. Besides, by using the statistical test we considerably reduce the standard deviation of the number of selected features, having a more stable selection. The FE step proves to be the

FE method	**FS method**	**#feat.**	**Balanced**		**class AUC**		**AUC**
			accuracy	TDC	ADHD-I	ADHD-C	
No	None	61.425	**39, 40** **±14, 66%**	0,592	0,502	**0,649**	0, 582 ±0, 057
	BagCCA-CV	21.806 ±19.202	36, 11 ±10, 52%	0,564	0,501	0,634	0, 571 ±0, 045
	BagCCA-ST	18.295 ±4.393	38, 65 ±12, 23%	0,589	0,516	0,634	0, 588 ±0, 044
Standard CCA	None	61.425	34, 53 ±8, 78%	0,593	0,506	0,629	0, 589 ±0, 057
	BagCCA-CV	16.277 ±11.978	37, 18 ±8, 72%	0,582	0,513	0,619	0, 580 ±0, 051
	BagCCA-ST	18.295 ±4.393	36, 18 ±8, 96%	0,598	0,528	0,624	0, 594 ±0, 039
Regularised CCA	None	61.425	37, 10 ±8, 73%	0,597	0,492	0,627	0, 589 ±0, 055
	BagCCA-CV	24.877 ±17.879	37, 40 ±6, 96%	0,590	0,518	0,625	0, 587 ±0, 050
	BagCCA-ST	18.295 ±4.393	**38, 48** **±8, 18%**	**0,600**	**0,530**	0,644	**0, 600** **±0, 058**

Table 3.6: ADHD - Performance results with the different versions of the method, considering the usage of the proposed selection and extraction methods in their balanced version. This table shows the performance in terms of multiclass AUC and multiclass balanced accuracy for different combinations of FE and FS methods. The results are given with the mean and standard deviation over the test 10-fold CV.

only approach capable of including FS and reducing the features to only two projections while maintaining the performance of the original SVM.

Finally, jointly analysing the results obtained in both databases we can conclude that while both FS methods provide similar results, using the statistical test step considerably reduces the computational time without degrading the performance (and even improving it in some cases). Moreover, the inclusion of the regularisation in CCA does pose a considerable improvement in the performance, especially when combined with the proposed FS. Note that we are improving the classification results obtained by a linear SVM using only three and two features (latent factors) with respect to using the 29.852 and 61.425 original features, respectively.

On another note, we can also analyse the effect of the proposed balanced version of RB-CCA-ST on the results. Specifically, we compare the classification performance of the model with and without the proposed balancing techniques combined with a balanced or imbalanced SVM, respectively. In Tables 3.7 and 3.8 we can see that the regularisation term is now giving more adequate weights to each class which, in turn, enhances the classification AUC in the low populated classes, i.e. sMCI and pMCI and ADHD-I. Furthermore, the improvement on the less populated classes does not negatively affect the more populated, on the contrary, assigning the corresponding weight to each class enhances the classification performance for the whole dataset.

Method	**Version**	**#feat.**	**Accuracy**	**class AUC**				**AUC**
				NC	sMCI	pMCI	AD	
RB-CCA-ST	Balanced	13.222 ±1.004	**62,91** **±4,64%**	**0,915**	**0,766**	**0,830**	**0,882**	**0,864** **±0,024**
RB-CCA-ST	Unbalanced	13.222 ±1.004	61,05 ±6,24%	0,911	0,755	0,822	0,878	0,858 ±0,022

Table 3.7: ADNI - Performance results with the RB-CCA-ST approach, considering the usage of the proposed selection and extraction methods in their balanced and unbalanced versions.

Method	**Version**	**#feat.**	**Accuracy**	**class AUC**			**AUC**
				TDC	ADHD-I	ADHD-C	
RB-CCA-ST	Balanced	18.295 ±4.393	**38,48** **±8,18%**	**0,600**	**0,530**	**0,644**	**0,600** **±0,058**
RB-CCA-ST	Unbalanced	18.295 ±4.393	**38,48** **±10,74%**	0,595	0,518	0,627	0,591 ±0,048

Table 3.8: ADHD - Performance results with the RB-CCA-ST model, considering the usage of the proposed FS and FE methods in their balanced and unbalanced versions.

3.2.6 Selected features and eigen-MRI

In this section we will study the interpretability of the learnt models. We will analyse which features the model considers to be relevant and how they influence in the construction each extracted factor. Finally, we will interpret the learnt projections to see how they are able to separate each data depending on their predicted label. Specifically, we will refer to the learnt projection matrices as eigen-MRIs.

Let's start by analysing the relevant voxels found by the FS in the ADNI database. Figure 3.4 depicts the class-wise selection obtained by RB-CCA-ST. For this images we included some axial slices which include areas that have been previously found to be relevant for the determination of AD, namely temporal, frontal and parietal areas, hypothalamus, cingulate gyrus and hippocampus (Frisoni et al., 2010; Weiner et al., 2017). We conclude that the selection obtained by the model coincides with the locations specified in the literature.

Equivalently, Figure 3.5 includes the sign consistency parameters b_{jc} over 10 CV-folds learnt by the model through the bagging process. Note that $b_{jc} \in [0, 1]$, where 0 corresponds to variables that change their sign in half of the bagging iteration and 1 to those that are sign consistent. The results obtained for this image provide information of the class-wise and the complete relevance of each selected voxel. Here, we can see that class NC has higher relevance for the areas in charge of the motor skills: putamen and the globus pallidus (which comprises the lentiform nucleus). For the case of the selection for sMCI, we can clearly see the selection of both lateral ventricles (first and second) as well as the third ventricle which, according to Carmichael et al. (2007), have

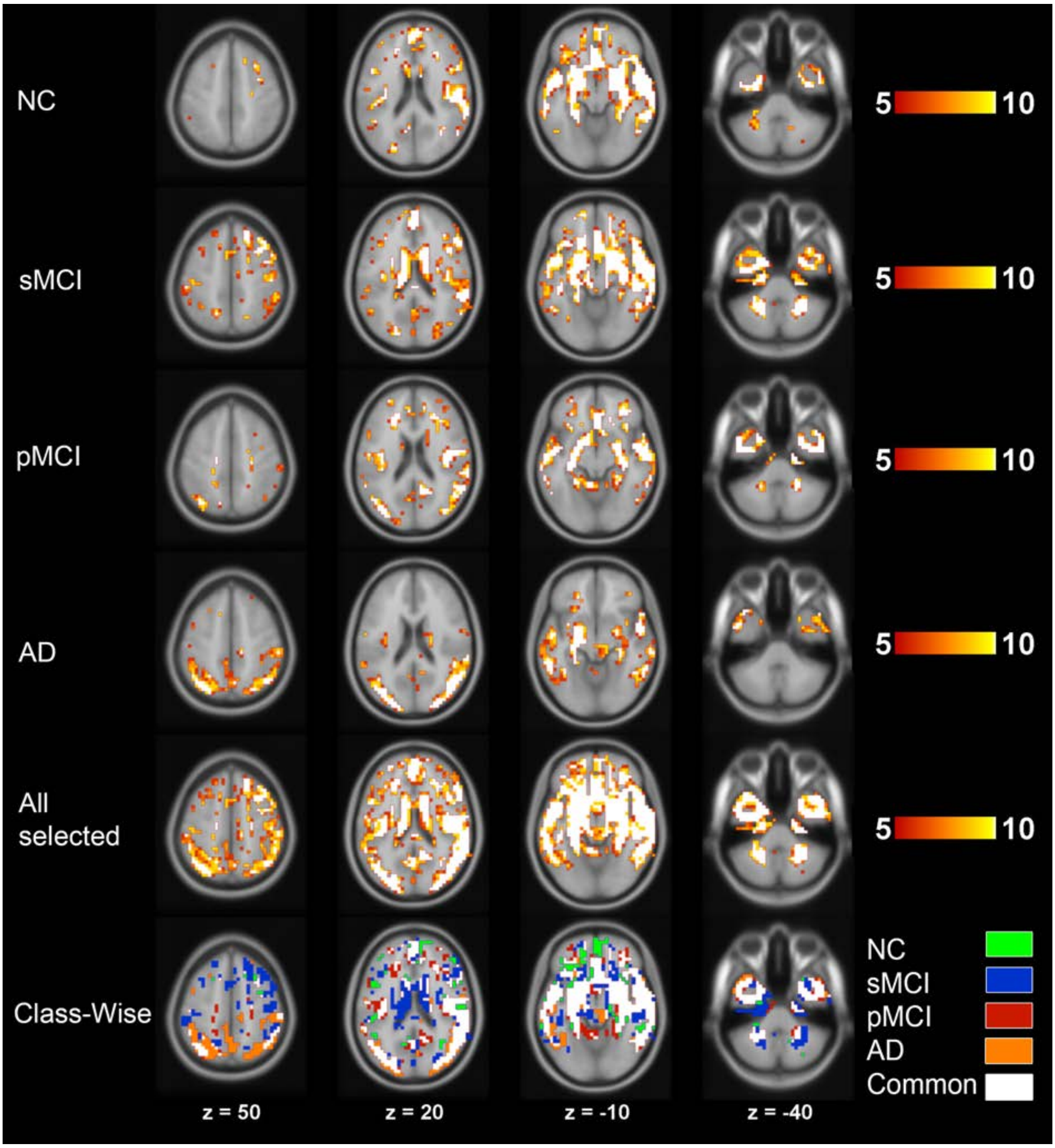


Figure 3.4: Locations of the most frequently selected voxels using RB-CCA-ST with ADNI data. The intensity of the overlay gives the number of times a voxel has been selected during the 10-fold CV. We have set a threshold of 5 folds to show only voxels selected in at least half of the folds. Four axial slices are shown, at $z = 50mm, 20mm - 10mm, -40mm$ of MNI space. The first 4 rows show the selected features for each class. The fifth row shows the union of the previous rows, which corresponds to the classifier input data, $\mathbf{X}_{:,\mathbf{S}}$. The bottom row shows the same union, specifying to which class the selected voxels correspond.

a close relation to the development of AD. Although the model does not consider as relevant the hippocampus for the classification of sMCI, the pMCI selection focuses on the hippocampus (in charge of creating new memories), thalamus (which processes the sensory information), the fusiform gyrus (involved in face and word recognition). Finally, for AD, the learnt relevance

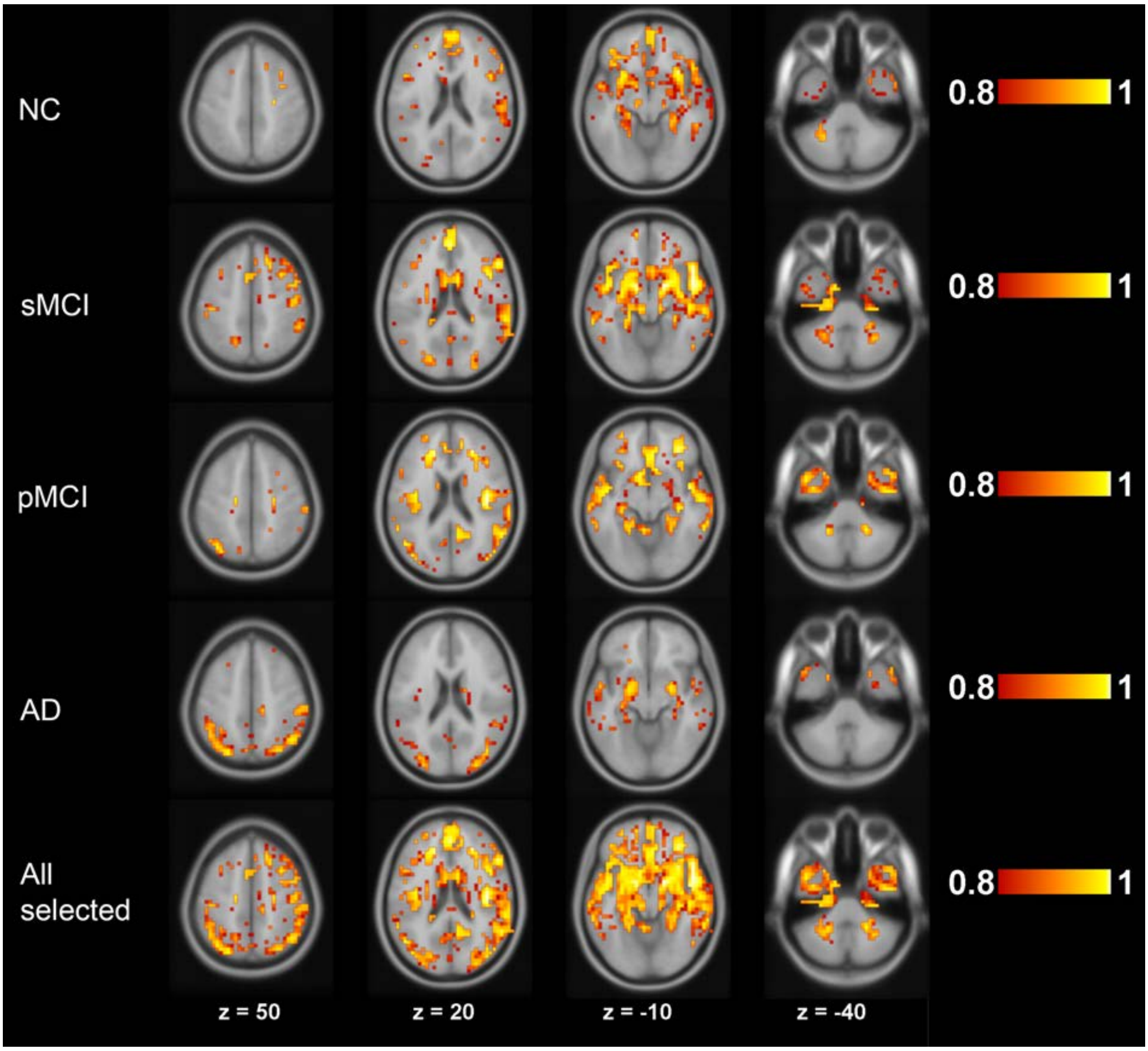


Figure 3.5: Variable relevance in ADNI data using RB-CCA-ST. Only relevances of voxels which have been selected at least in 5 of 10 CV-folds are shown. Four axial slices are shown, at $z = 50mm, 20mm - 10mm, -40mm$ of the MNI space. Four top rows show the class-wise relevances ($\mathbf{b}_{:,\mathrm{c}}$ where c $= 1, \ldots, 4$) of the voxels and the bottom row shows the relevance of all the selected voxels.

focuses on the grey matter of the occipital lobe (recognises and analyses the visual information) and parietal lobe (mainly focuses on spatial and touch sense).

CCA learns some summary components or eigen-MRIs which determine a projection of the original data into an orthogonal space (Equation (3.13)). In particular, in the ADNI database we calculated three eigen-MRIs which combine the information of the sign and magnitude consistency for each voxel to later use it to classify the data. In our case, the learnt eigen-MRIs are the projection matrix $\mathbf{U}$ (2.33), which relates the input feature space and the low-dimensional orthogonal space. We can visualise the mean value of the learnt eigenvectors over the CV folds in Figure 3.6, where the number associated to each feature specify their influence in the determination of the extracted eigen-MRIs. As it would be too extensive to analyse every single area described in this image, let's focus on two particularly relevant ones: the thalamus and

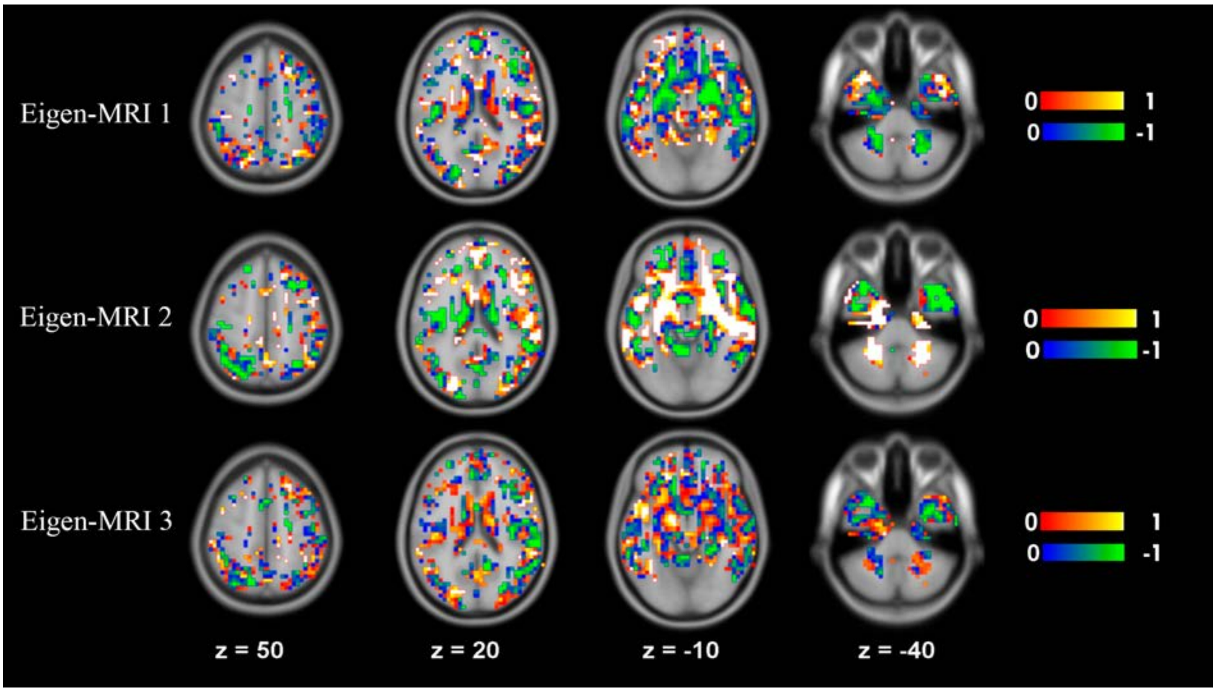
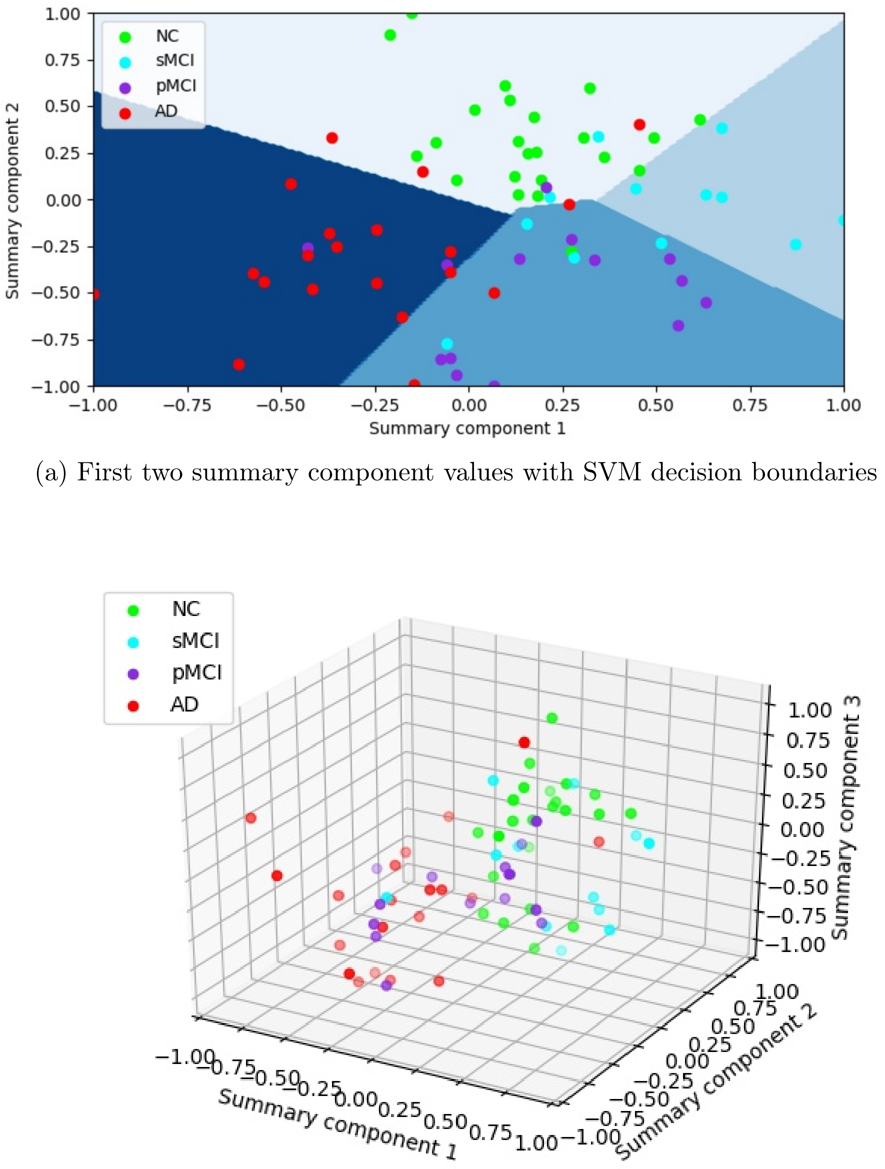


Figure 3.6: Normalised mean values of the generated eigen-MRIs using RB-CCA-ST with ADNI data. This has been combined with the FS learnt through bagging, here presenting the effect on the eigen-MRI of the most selected voxels in the 10-fold CV.

the lentiform nucleus. The first eigen-MRI is generated using a highly negative weight for the lentiform nucleus and a mostly negative weight for the thalamus. This contrasts with the second eigen-MRI, where the lentiform nucleus is highly positive while most areas corresponding to the thalamus have a highly negative associated weight. Finally, looking at the last eigen-MRI, we can see the thalamus is now mildly positive, while the lentiform nucleus has different sign and associated weights depending on the area. These results prove that the model provides a cohesive measure of not only the importance of each brain area for the task, but also the correlation between different brain areas. This allows the model to exploit these correlations to have a wide expressiveness, combining different signs and weights for correlated areas so that with only 3 eigen-MRIs we are capable of creating a sufficiently informative low dimensional dataset.

Conversely, we can exploit the projection of the input space into a low dimensional space and observe the data relation in the orthogonal space, as seen in Figure 3.7. Here, we included the learnt projection of the original data, summary components, for a representative fold. Figure 3.7(a) represents the two most representative summary components, as well as the SVM boundaries between each pair of classes learnt at training for one CV fold. We can observe that, thanks to the decorrelation and orthogonality between them, the first one separates AD subjects from the rest, while the second one separates the rest. The obtained overlapping between classes is expected as pMCI subjects are more similar to AD and sMCI to NC subjects. Furthermore, Figure 3.7(b) shows a normalised 3D representation of the three learnt summary components. This projection does not provide an easily interpretable insight on the structure of the data, implying that the third component is less helpful for the separation of the different classes.

We followed the same procedure for the ADHD database, obtaining the equivalent images for each analysis. The database is composed by fMRIs, which uses relations between regions, this



(a) First two summary component values with SVM decision boundaries

(b) Plot of the three summary component values

Figure 3.7: Normalised latent factor values for one representative fold with the SVM's decision boundaries using ADNI data. The proposed method finds a projection of the data capable of representing the dataset with 3 values. The first two summary components are the most informative and the ones that have the biggest role in the projection of the data. The first eigen-MRI is capable of discriminating between sMCI and AD. The second summary component separates NC from pMCI.

implies that working with relations instead of voxels does not allow to represent the eigen-MRIs, the previous brain sections. Unfortunately, the fact that this database is composed by fMRI makes the interpretability of the results less intuitive, hindering the analysis of the figures. For this reason we decided not to include here the images with the selected features and the learnt functional eigen-MRIs. However, the analysis of the correlation matrices used as input as well as

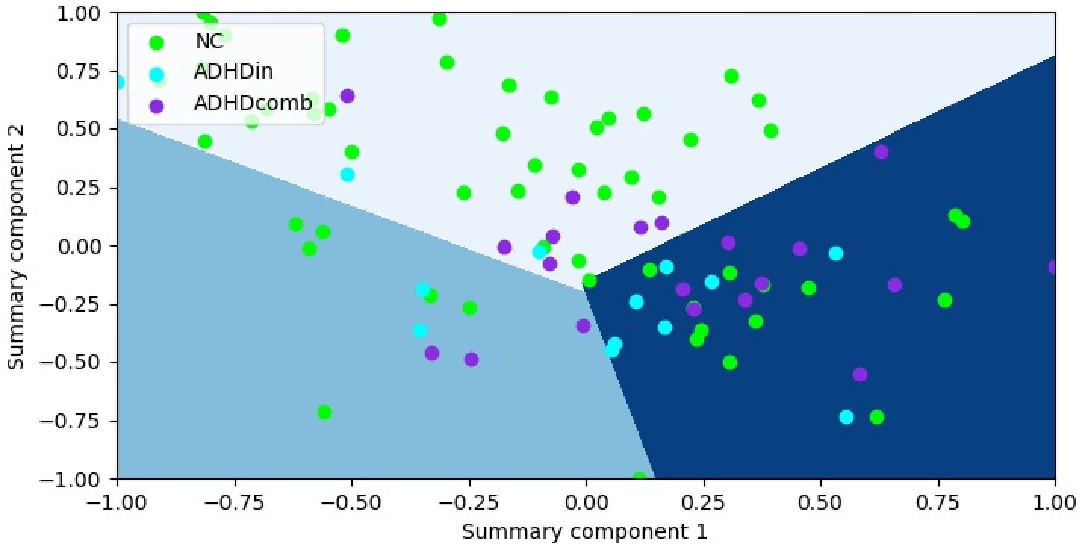


Figure 3.8: ADHD - Normalised summary components for one representative fold with the SVM's boundaries. The proposed method finds a projection of the data capable of representing the dataset with 2 values.

their associated learnt eigen-MRIs are available in the supplementary material in Sevilla-Salcedo et al. (2020a).

Nevertheless, what we can easily represent is the two summary components in a 2D plot as depicted in Figure 3.8. For this representation we include the results of a representative fold and show the SVM boundaries between each pair of classes learnt at training. As happened with the ADNI database, the main problem with this database is the discrimination of the minority class. This can be seen in the representation of the summary components, where the less populated classes are relatively separated but do not lead to any relevant result.

Chapter 4

Sparse Semi-supervised Heterogeneous Interbattery Bayesian Analysis

In the previous chapter, we proposed a model to work as a FE method using a reformulation of the classical CCA. In this chapter, we want to study this problem from a new point of view. Here we present a novel Bayesian FE model based on the BIBFA formulation reviewed in Section 2.2.2.3 which provides a different insight into the latent factors and the different functionalities that can be combined with them.

Throughout this chapter, we will present the proposed model: Sparse Semi-supervised Heterogeneous Interbattery Bayesian Analysis (SSHIBA), which generalises BIBFA by adding different extensions. This model combines the latent variables of the formulation with their automatic selection. It also includes FS capabilities, facilitates modelling heterogeneous data and works with semi-supervised learning, kernel extensions and multi-view data. These adaptations ease the work with neuroimaging datasets where the large number of features relative to the number of samples makes the use of feature selection and dual kernel-based formulations essential. Furthermore, semi-supervised and heterogeneous multiview extensions address another major problem of neuroimaging datasets, as they allow handling different data sources that complement neuroimaging data, e.g. demographics, genetics, tests,... and dealing with missing values, which are common in this type of problem. The model and part of the results of this chapter are published in Sevilla-Salcedo et al. (2021, 2020b).

4.1 Methodology

Starting from the formulation developed in Section 2.2.2.3, we present here the different extensions that comprise the SSHIBA model. This model enhances BIBFA by including several functionalities that improve its adaptability to neuroimaging scenarios. In particular, the main extensions and modifications are included for:

1) **Interpretability**: with the proposed FS, the model is capable of simultaneously inducing sparsity in the latent and in the feature space by including a double ARD prior in the latent matrix $\mathbf{W}^{(\mathrm{m})}$ for each view. In this way, we can get a measure of the relevance of

each input feature and a better understanding of the effect of the features on the problem. Furthermore, the sparsity in the latent space provides a better insight on the relation between each view and allows the model to automatically learn latent representations by combining different views.

2) **Combining the available information**: the proposed heterogeneous formulation allows SSHIBA to adequately model different types of input data in different views. This allows the model to combine the main input data, e.g. the MRI, with some additional information that can improve the performance of the model, such as patient demographic information or different test results. In particular, the model is now able to treat two new types of data: binary (multi-label) data, which works with data for which more than one label can simultaneously be positive for the same patient, e.g. resistance to different types of antibiotics; and categorical data, which allows to appropriately model information that corresponds to labels that are number independent, e.g. the hospital where the data was collected.

3) **Treating missing data**: while the predictive formulation allows the model to estimate some output using test data, the semi-supervised learning prevents SSHIBA from relying on the predictive formulation to estimate the unobserved data. In particular, this allows the model to use all the available data to train the model, further increasing the number of samples used to estimate the model parameters. Furthermore, this extension proves to be specially powerful in neuroimaging scenarios where the need of expensive experiments increases the number of missing values of the problem.

4) **Working in high dimensional problems**: we can make use of the model formulation to use kernels instead of the observed variables as input for the proposed model. The reasoning behind this is twofold. First, this provides the means to effectively work with high-dimensional problems, by working in the dual space, where the number of parameters depends on the number of samples instead of features. This is specially critical in neuroimaging problems where the number of features is considerably greater than the number of samples, which makes working with, at least, linear kernels critical to train the model adequately. Besides, including the kernel formulation allows to include non-linearities by means of a non-linear kernel, such as a Gaussian kernel, or combine different types of kernels in different views, having a multiple kernel learning framework.

These proposed extensions can be added interchangeably and combined with the basic formulation. As described in Figure 4.1, this formulation starts from a common latent space, given by the r.v. $\mathbf{Z}$, which is combined with different projection matrices of different views $\left\{\mathbf{W}^{(\mathrm{m})}\right\}_{\mathrm{m}=1}^{\mathrm{M}}$, Gaussian noise and a bias term are added to generate the corresponding data views $\left\{\mathbf{W}^{(\mathrm{m})}\right\}_{\mathrm{m}=1}^{\mathrm{M}}$, added Gaussian noise and a bias term to generate the corresponding data views $\left\{\mathbf{X}^{(\mathrm{m})}\right\}_{\mathrm{m}=1}^{\mathrm{M}}$. We decided to include a bias term in the formulation to deal with any kind of normalisation problems in the data, avoiding possible inconsistencies in the weight learning. Thus, we added a new variable $\mathbf{b}^{(\mathrm{m})}$ to the observations $\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}$ which models any bias information in the data. Therefore,

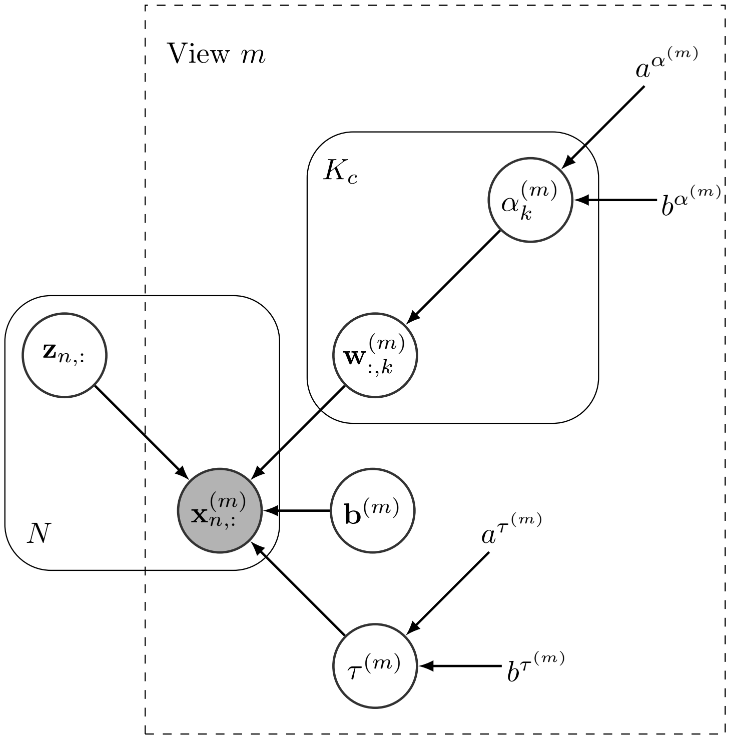


Figure 4.1: SSHIBA's graphical model.

the joint probability distribution of the model variables is

$$\mathbf{z}_{\mathrm{n},:} \sim \mathcal{N}(0, \mathbf{I}_{\mathrm{K_c}}) \tag{4.1}$$

$$\mathbf{w}_{:,\mathrm{k}}^{(\mathrm{m})} \sim \mathcal{N}\left(0, \left(\alpha_{\mathrm{k}}^{(\mathrm{m})}\right)^{-1} \mathbf{I}_{\mathrm{K_c}}\right) \tag{4.2}$$

$$\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})} \,|\, \mathbf{z}_{\mathrm{n},:} \sim \mathcal{N}(\mathbf{z}_{\mathrm{n},:} \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}} + \mathbf{b}^{(\mathrm{m})}, \tau^{(\mathrm{m})^{-1}} \mathbf{I}_{\mathrm{D_m}}) \tag{4.3}$$

$$\mathbf{b}^{(\mathrm{m})} \sim \mathcal{N}(0, \mathbf{I}_{\mathrm{D_m}}) \tag{4.4}$$

$$\alpha_{\mathrm{k}}^{(\mathrm{m})} \sim \Gamma\left(a^{\boldsymbol{\alpha}^{(\mathrm{m})}}, b^{\boldsymbol{\alpha}^{(\mathrm{m})}}\right) \tag{4.5}$$

$$\tau^{(\mathrm{m})} \sim \Gamma\left(a^{\tau^{(\mathrm{m})}}, b^{\tau^{(\mathrm{m})}}\right) \tag{4.6}$$

One of the main highlights of this formulation is its modular structure, which allows to combine different extensions in the same framework, e.g. one view can model multi-label data and missing values, another view can work with real data and sparsity in the input features and a third view can deal with categorical data. Throughout the following sections we develop the different proposed extensions that compose SSHIBA. To enhance the reading experience, the detailed explanations of the proposed algorithms, as well as the inclusion of the bias term, are included in Appendix A.

4.1.1 Feature selection with SSHIBA

One of the novel functionalities presented is the inclusion of FS capabilities to the model. In this section, we present an adaptation for the distribution of the projection matrix $\mathbf{W}^{(\mathrm{m})}$ that allows the model to simultaneously select the relevant latent factors and the useful input features.

Generative model

For this new tool we propose the inclusion of a double ARD prior along with a new variable, $\boldsymbol{\gamma}^{(\mathrm{m})}$, in charge of the sparsity in the input features:

$$\begin{aligned} \mathrm{w}_{\mathrm{d,k}}^{(\mathrm{m})} &\sim \mathcal{N}\left(0, \left(\gamma_{\mathrm{d}}^{(\mathrm{m})}\, \alpha_{\mathrm{k}}^{(\mathrm{m})}\right)^{-1}\right) && (4.7) \\ \gamma_{\mathrm{d}}^{(\mathrm{m})} &\sim \Gamma\left(a^{\gamma^{(\mathrm{m})}}, b^{\gamma^{(\mathrm{m})}}\right) && (4.8) \end{aligned}$$

where the variance of $\mathrm{w}_{\mathrm{d,k}}^{(\mathrm{m})}$ is now calculated as the product of two r.v.: one that affects the rows, $\alpha_{\mathrm{k}}^{(\mathrm{m})}$, defined in equation (2.77) and in charge of the selection of latent factors; and another r.v. that affects the columns, $\gamma_{\mathrm{d}}^{(\mathrm{m})}$, which induces the sparsity on the data features. This way, the model can automatically learn a sparse projection matrix in, both, rows and columns, leading to changes in the structure of the matrix and, subsequently, to an interpretability improvement. Specifically, the row-wise sparsity provides a measure of the relevance of each input feature and the column-wise sparsity the relation between the different views, namely, which latent factors are common or private. In Figure 4.2 we included the graphical model associated to this extension in a scenario with real value observations.

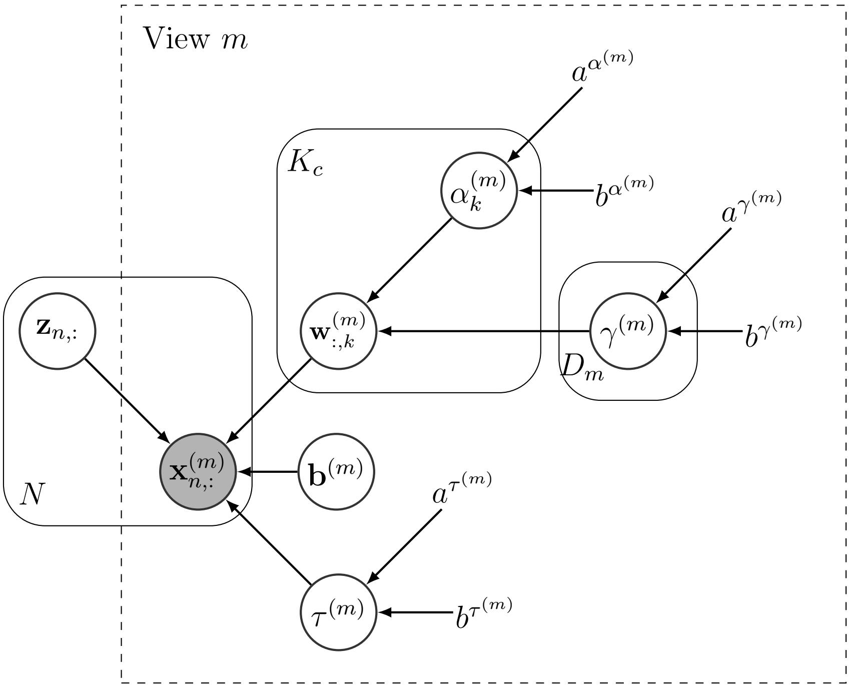


Figure 4.2: SSHIBA's feature selection graphical model.

Variational inference

As we have added a new term, $\boldsymbol{\gamma}^{(\mathrm{m})}$, to the model and modified the prior distribution of the projection matrix $\mathbf{W}^{(\mathrm{m})}$, we have to equivalently update the mean-field posterior distribution in

(2.79) as

$$
\begin{aligned}
&p(\Theta|\,\mathbf{X}^{\{\mathcal{M}\}}) \approx q(\Theta)\\
&= \prod_{\mathrm{m}=1}^{\mathrm{M}} \left(q\left(\mathbf{W}^{(\mathrm{m})}\right) q\left(\mathbf{b}^{(\mathrm{m})}\right) q\left(\tau^{(\mathrm{m})}\right) \prod_{\mathrm{k}=1}^{\mathrm{K_c}} q\left(\alpha_{\mathrm{k}}^{(\mathrm{m})}\right) \prod_{\mathrm{d}=1}^{\mathrm{D_m}} q\left(\gamma_{\mathrm{d}}^{(\mathrm{m})}\right) \right) \prod_{\mathrm{n}=1}^{\mathrm{N}} q(\mathbf{z}_{\mathrm{n},:}).
\end{aligned} \tag{4.9}
$$

Using this distribution we obtain the new update rules included in Table 4.1. A detailed development of these results is available in Appendix A.

Variable	q^* **distribution**	**Parameters**
$\mathbf{z}_{\mathrm{n},:}$	$\mathcal{N}(\mathbf{z}_{\mathrm{n},:} \mid \langle \mathbf{z}_{\mathrm{n},:} \rangle, \Sigma_{\mathbf{Z}})$	$\langle \mathbf{z}_{\mathrm{n},:} \rangle = \sum_{\mathrm{m}=1}^{\mathrm{M}} \left(\langle \tau^{(\mathrm{m})} \rangle \left(\mathbf{X}^{(\mathrm{m})} - \mathbb{1}_N \langle \mathbf{b}^{(\mathrm{m})} \rangle \right) \langle \mathbf{W}^{(\mathrm{m})} \rangle \right) \Sigma_{\mathbf{Z}}$ $\Sigma_{\mathbf{Z}}^{-1} = I_{\mathrm{K_c}} + \sum_{\mathrm{m}=1}^{\mathrm{M}} \langle \tau^{(\mathrm{m})} \rangle \langle \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}} \mathbf{W}^{(\mathrm{m})} \rangle$
$\mathbf{W}^{(\mathrm{m})}$	$\prod_{\mathrm{d}=1}^{\mathrm{D_m}} \mathcal{N}\left(\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})} \mid \langle \mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})} \rangle, \Sigma_{\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})}} \right)$	$\langle \mathbf{W}^{(\mathrm{m})} \rangle = \langle \tau^{(\mathrm{m})} \rangle \left(\mathbf{X}^{(\mathrm{m})} - \mathbb{1}_N \langle \mathbf{b}^{(\mathrm{m})} \rangle \right)^T \langle \mathbf{Z} \rangle \Sigma_{\mathbf{W}^{(\mathrm{m})}}$ $\Sigma_{\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})}}^{-1} = \mathrm{diag}(\langle \boldsymbol{\alpha}^{(\mathrm{m})} \rangle) \langle \gamma_{\mathrm{d}}^{(\mathrm{m})} \rangle + \langle \tau^{(\mathrm{m})} \rangle \langle \mathbf{Z}^{\mathrm{T}} \mathbf{Z} \rangle$
$\mathbf{b}^{(\mathrm{m})}$	$\mathcal{N}\left(\mathbf{b}^{(\mathrm{m})} \mid \langle \mathbf{b}^{(\mathrm{m})} \rangle, \Sigma_{\mathbf{b}^{(\mathrm{m})}} \right)$	$\langle \mathbf{b}^{(\mathrm{m})} \rangle = \langle \tau^{(\mathrm{m})} \rangle \sum_{\mathrm{n}=1}^{\mathrm{N}} \left(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})} - \langle \mathbf{z}_{\mathrm{n},:} \rangle \langle \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}} \rangle \right) \Sigma_{\mathbf{b}^{(\mathrm{m})}}$ $\Sigma_{\mathbf{b}^{(\mathrm{m})}}^{-1} = \left(N \langle \tau^{(\mathrm{m})} \rangle + 1 \right) I_{\mathrm{D_m}}$
$\boldsymbol{\alpha}^{(\mathrm{m})}$	$\prod_{\mathrm{k}=1}^{\mathrm{K_c}} \Gamma\left(\alpha_{\mathrm{k}}^{(\mathrm{m})} \mid a_{\alpha_{\mathrm{k}}^{(\mathrm{m})}}, b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}} \right)$	$a_{\alpha_{\mathrm{k}}^{(\mathrm{m})}} = \frac{\mathrm{D_m}}{2} + a^{\boldsymbol{\alpha}^{(\mathrm{m})}}$ $b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}} = b^{\boldsymbol{\alpha}^{(\mathrm{m})}} + \frac{1}{2} \sum_{\mathrm{d}=1}^{\mathrm{D_m}} \langle \gamma_{\mathrm{d}}^{(\mathrm{m})} \rangle \langle \mathrm{w}_{\mathrm{d,k}}^{(\mathrm{m})} \mathrm{w}_{\mathrm{d,k}}^{(\mathrm{m})} \rangle$
$\tau^{(\mathrm{m})}$	$\Gamma\left(\tau^{(\mathrm{m})} \mid a_{\tau^{(\mathrm{m})}}, b_{\tau^{(\mathrm{m})}} \right)$	$a_{\tau^{(\mathrm{m})}} = \frac{\mathrm{D_m}\,\mathrm{N}}{2} + a^{\tau^{(\mathrm{m})}}$ $b_{\tau^{(\mathrm{m})}} = b^{\tau^{(\mathrm{m})}} + \frac{1}{2} \sum_{\mathrm{n}=1}^{\mathrm{N}} \sum_{\mathrm{d}=1}^{\mathrm{D_m}} \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})^2}$ $- \mathrm{Tr}\left\{ \langle \mathbf{W}^{(\mathrm{m})} \rangle \langle \mathbf{Z}^{\mathrm{T}} \rangle \mathbf{X}^{(\mathrm{m})} \right\} + \frac{1}{2} \mathrm{Tr}\left\{ \langle \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}} \mathbf{W}^{(\mathrm{m})} \rangle \langle \mathbf{Z}^{\mathrm{T}} \mathbf{Z} \rangle \right\}$ $- \sum_{\mathrm{n}=1}^{\mathrm{N}} \mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})} \langle \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} \rangle + \sum_{\mathrm{n}=1}^{\mathrm{N}} \langle \mathbf{z}_{\mathrm{n},:} \rangle \langle \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}} \rangle \langle \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} \rangle + \frac{N}{2} \langle \mathbf{b}^{(\mathrm{m})} \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} \rangle$
$\boldsymbol{\gamma}^{(\mathrm{m})}$	$\prod_{\mathrm{d}=1}^{\mathrm{D_m}} \Gamma\left(\gamma_{\mathrm{d}}^{(\mathrm{m})} \mid a_{\gamma_{\mathrm{d}}^{(\mathrm{m})}}, b_{\gamma_{\mathrm{d}}^{(\mathrm{m})}} \right)$	$a_{\gamma_{\mathrm{d}}^{(\mathrm{m})}} = \frac{\mathrm{K_c}}{2} + a^{\boldsymbol{\gamma}^{(\mathrm{m})}}$ $b_{\gamma_{\mathrm{d}}^{(\mathrm{m})}} = b^{\boldsymbol{\gamma}^{(\mathrm{m})}} + \frac{1}{2} \sum_{\mathrm{k}=1}^{\mathrm{K_c}} \langle \alpha_{\mathrm{k}}^{(\mathrm{m})} \rangle \langle \mathrm{w}_{\mathrm{d,k}}^{(\mathrm{m})} \mathrm{w}_{\mathrm{d,k}}^{(\mathrm{m})} \rangle$

Table 4.1: Distribution q of the different r.v. of the graphical model for feature selection together with the different distribution parameters. Where $\mathbb{1}_N$ is a row vector of ones of dimension N. See Appendix A for further details.

Once the model parameters are learnt, we can analyse $\boldsymbol{\gamma}^{(\mathrm{m})}$ to determine the relevance of each variable. Namely, when $\boldsymbol{\gamma}^{(\mathrm{m})}$ takes lower values the feature is more relevant and when it takes high values the feature is less relevant, providing an online feature ranking for any specified view. This has a straightforward adaptation to FS by setting a threshold to the learnt relevance. Although we will usually set a restrictive prior distribution over $\gamma_{\mathrm{d}}^{(\mathrm{m})}$ that should automatically bring down to 0 the irrelevant features, in certain problems the relevance might have a more uniform value over all variables and $\boldsymbol{\gamma}^{(\mathrm{m})}$ only provides a variable ranking, needing to set a threshold to obtain a real FS.

Finally, note that the predictive model is not affected by the inclusion of these new terms, having the same predictive formulation explained in Section 2.2.2.3.

4.1.2 Heterogeneous data: Multidimensional binary views

Here we introduce a SSHIBA extension to work with heterogeneous data. In this instance, this formulation allows SSHIBA to model multidimensional binary data. The applications of this extension are numerous as, beyond using multi-label data in a classification problem, it also allows other data sources such as demographic data or genetic information to be properly modelled. This way, we can construct a model combining, for example, multi-label data in one view, demographic in another, genetic information in a different one and psychological tests or other type of evaluation in a separate view.

Generative model

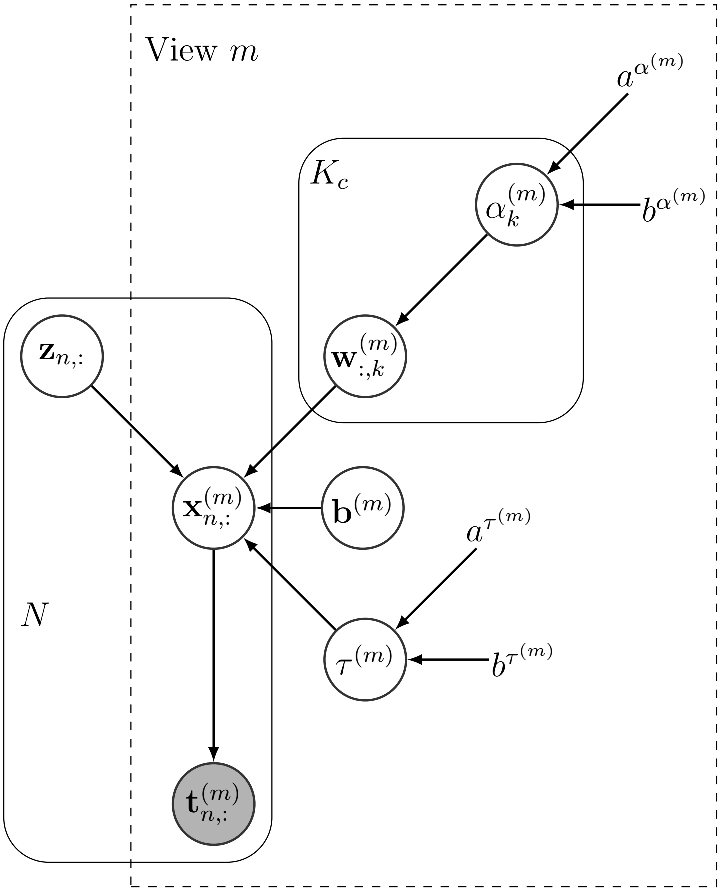


Figure 4.3: SSHIBA graphical model for multi-dimensional binary views.

For this extension, we draw on the Bayesian logistic regression presented in Jaakkola and Jordan (1997) to adapt the model to work with binary data. Thus, we obtain the graphical model in Figure 4.3, where we now consider the m-th view to be multi-labelled, so that the binary data $\mathbf{T}^{(\mathrm{m})}$ is observed and $\mathbf{X}^{(\mathrm{m})}$ is now an unobserved variable. $\mathbf{X}^{(\mathrm{m})}$ is generated from the latent variables $\mathbf{Z}$ combined with the projection matrix $\mathbf{W}^{(\mathrm{m})}$ with some Gaussian noise and a bias term. Consequently, the new observed variable $\mathbf{T}^{(\mathrm{m})} \in \mathbb{R}^{\mathrm{N} \times \mathrm{D_m}}$ is related to $\mathbf{X}^{(\mathrm{m})}$ by

$$p\left(\mathbf{t}_{\mathrm{n},:}^{(\mathrm{m})} \,|\, \mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}\right) = \prod_{\mathrm{d}=1}^{\mathrm{D_m}} p\left(\mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m})} \,|\, \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})}\right) \tag{4.10}$$

$$p\left(\mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m})} \,|\, \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})}\right) = \sigma\left(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})}\right)^{\mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m})}} \left(1 - \sigma\left(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})}\right)\right)^{1-\mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m})}} = e^{\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})}\,\mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m})}} \sigma\left(-\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})}\right), \tag{4.11}$$

where $\sigma(a)$ is the sigmoid function $(1 + e^{-a})^{-1}$. As specified in Jaakkola and Jordan (1997), instead of directly working with the logistic regression conditional probability, we can set the following lower bound to this probability:

$$\begin{aligned} p\left(\mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m})} \,|\, \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})}\right) \ &= \ e^{\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})}\,\mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m})}} \sigma\left(-\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})}\right) \geq \\ & e^{\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})}\,\mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m})}} \sigma\left(\xi_{\mathrm{n,d}}^{(\mathrm{m})}\right) e^{-\frac{\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})} + \xi_{\mathrm{n,d}}^{(\mathrm{m})}}{2} - \lambda\left(\xi_{\mathrm{n,d}}^{(\mathrm{m})}\right)\left(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})^2} - \xi_{\mathrm{n,d}}^{(\mathrm{m})^2}\right)} \end{aligned} \tag{4.12}$$

where $\lambda(a) = \frac{1}{2a}\left(\sigma(a) - \frac{1}{2}\right)$, and $\xi_{\mathrm{n,d}}^{(\mathrm{m})}$ is a variational parameter which can be optimised by maximising the evidence lower bound in (A.39).

Variational inference

By the inclusion of the changes in the r.v. previously described, the mean-field variational family can be approximated as

$$\begin{aligned} & p\left(\Theta |\, \mathbf{T}^{\{\mathcal{M}_t\}}, \mathbf{X}^{\{\mathcal{M}_r\}}\right) \approx q(\Theta) \\ &= q(\mathbf{Z}) \prod_{\mathrm{m_t} \in \mathcal{M}_\mathrm{t}} \left(\prod_{\mathrm{n}=1}^{\mathrm{N}} q\left(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\right)\right) \prod_{\mathrm{m}=1}^{\mathrm{M}} q\left(\mathbf{W}^{(\mathrm{m})}\right) q\left(\mathbf{b}^{(\mathrm{m})}\right) q\left(\boldsymbol{\alpha}^{(\mathrm{m})}\right) q\left(\tau^{(\mathrm{m})}\right) q\left(\boldsymbol{\gamma}^{(\mathrm{m})}\right), \end{aligned} \tag{4.13}$$

where $\mathcal{M}_t$ is the set of views with multidimensional binary data, $\mathrm{m_t}$ is one of the views belonging to $\mathcal{M}_t$ and $\mathcal{M}_r$ is the set of views with real-valued data. This formulation leads to a change in $q\left(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\right)$, included in Table 4.2. Due to the way the model is formulated, the modifications here proposed only require to replace in Tables 2.3 and 4.1 $\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}$ (or the stacked data matrix $\mathbf{X}^{(\mathrm{m_t})}$) by its mean $\langle \mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})} \rangle$ ($\langle \mathbf{X}^{(\mathrm{m_t})} \rangle$) for each data point in view $\mathrm{m_t}, \forall \mathrm{m_t} \in \mathcal{M}_\mathrm{t}$. Thus, we have a variational update-rule equivalent to the previous one with the exception of $q\left(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\right)$.

Variable	q distribution	Parameters
$\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}$	$\mathcal{N}\left(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})} \,|\langle \mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})} \rangle, \Sigma_{\mathbf{X}^{(\mathrm{m_t})}}\right)$	$\langle \mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})} \rangle = \left(\mathrm{t}_{\mathrm{n},:}^{(\mathrm{m_t})} - \frac{1}{2} + \langle \tau^{(\mathrm{m_t})} \rangle \langle \mathbf{z}_{\mathrm{n},:} \rangle \langle \mathbf{W}^{(\mathrm{m_t})^\mathrm{T}} \rangle + \langle \mathbf{b}^{(\mathrm{m_t})} \rangle\right) \Sigma_{\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}}$ $\Sigma_{\mathbf{X}^{(\mathrm{m_t})}}^{-1} = \langle \tau^{(\mathrm{m_t})} \rangle I + 2\Lambda_{\boldsymbol{\xi}_{\mathrm{n},:}^{(\mathrm{m_t})}}$

Table 4.2: Mean-field update rule for the $q\left(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\right)$ distribution in (4.13), where $\Lambda_{\boldsymbol{\xi}_{n,:}}$ is a diagonal matrix for which the diagonal elements are $\lambda(\xi_{n,1}), \lambda(\xi_{n,2}), \ldots, \lambda(\xi_{n,\mathrm{D_m}})$. This distribution only affects views $\mathcal{M}_t$, which are modelled as multidimensional binary data. See Appendix A for further details.

Regarding the predictive distribution, this formulation requires approximate inference (e.g. variational inference or Monte Carlo) to estimate the posterior latent distribution in (2.80) w.r.t.

the observed data. Using the approximation suggested in Bishop (2006), the resulting predictive distribution is:

$$p\left(\mathrm{t}_{*,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}} = 1 | \, \mathrm{x}_{*,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{in}}\}}\right) \;=\; \sigma\left(\frac{\langle \mathrm{x}_{*,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}}\rangle}{\left(1+\frac{\pi}{8}\Sigma_{\mathbf{x}_{:,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}}}\right)^{1/2}}\right) \tag{4.14}$$

A full description of the predictive distribution for multidimensional binary data is available in Appendix A. Nevertheless, this can be substituted by the semi-supervised version, which will be presented in Section 4.1.4.

4.1.3 Heterogeneous data: Categorical observations

To complete the heterogeneous data functionality, in this section we present an extension to work with categorical observations.

Generative model

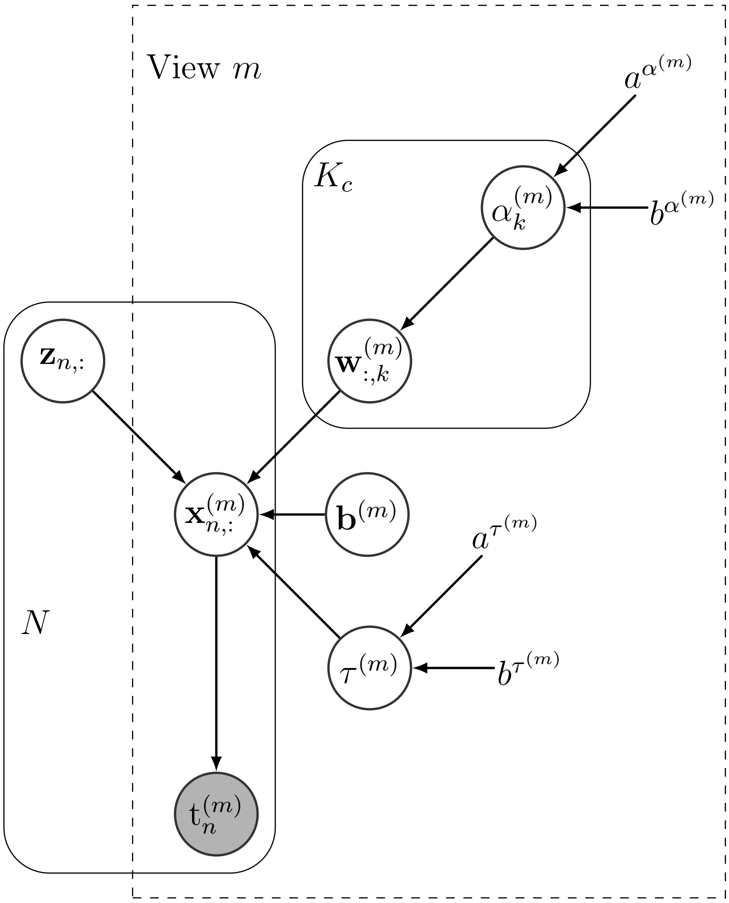


Figure 4.4: SSHIBA graphical model for categorical views.

The graphical model, shown in Figure 4.4, is equivalent to the one that handles multidimensional binary data, except that the observed data is now an array instead of a matrix. This means that $\mathrm{t}_\mathrm{n}^{(\mathrm{m})}$ is an integer with values in the set of $\{0,\ldots,\mathrm{D_m}-1\}$, where, in this scenario,

$\mathrm{D_m}$ corresponds to the number of categories in the m-th view. We combine the original model with a multinomial probit on the variable $\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}$, as proposed in Girolami and Rogers (2006), having that the relation between this variable and $\mathrm{t}_{\mathrm{n}}^{(\mathrm{m})}$ is

$$\mathrm{t}_{\mathrm{n}}^{(\mathrm{m})} = \mathrm{i} \qquad \text{if} \qquad \mathrm{x}_{\mathrm{n,i}}^{(\mathrm{m})} = \max_{\mathrm{d}=0,\ldots,\mathrm{D_m}-1} \left\{ \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})} \right\}, \tag{4.15}$$

Furthermore, Girolami and Rogers (2006) indicate that, by setting the noise variable $\tau^{(\mathrm{m})}$ to 1, we can state

$$p\left(\mathrm{t}_{\mathrm{n}}^{(\mathrm{m})} = \mathrm{i} |\, \mathbf{z}_{\mathrm{n},:}, \mathbf{W}^{(\mathrm{m})}\right) = \mathbb{E}_{p(u)}\left[\prod_{j\neq i}\left(\Phi\left(u + \mathrm{y}_{\mathrm{n,i}}^{(\mathrm{m})} - \mathrm{y}_{\mathrm{n,j}}^{(\mathrm{m})}\right)\right)\right], \tag{4.16}$$

where $\mathbf{y}_{\mathrm{n},:}^{(\mathrm{m})} = \mathbf{z}_{\mathrm{n},:} \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}$, $p(u) \sim \mathcal{N}(0,1)$, and $\Phi(\cdot)$ is the standard Gaussian cumulative distribution function. The expectation defined can then be approximated using Monte Carlo by sampling from an uni-dimensional standard Gaussian distribution.

Variational inference

Variable	q distribution	Parameters
$\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}$	$\frac{1}{\boldsymbol{\xi}_{n,:}}\mathcal{N}\left(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})} \,\vert \langle \mathbf{y}_{\mathrm{n},:}^{(\mathrm{m_t})}\rangle, \mathbf{I}\right)\times$ $\delta\left(\mathrm{x}_{\mathrm{n,i}}^{(\mathrm{m_t})} > \mathrm{x}_{\mathrm{n,j}}^{(\mathrm{m_t})}\, \forall \mathrm{i} \neq \mathrm{j}\right)$	$\langle \mathrm{x}_{\mathrm{n,i}}^{(\mathrm{m_t})}\rangle = \langle \mathrm{y}_{\mathrm{n,i}}^{(\mathrm{m_t})}\rangle + \sum_{\mathrm{j}\neq\mathrm{i}}\left(\langle \mathrm{y}_{\mathrm{n,j}}^{(\mathrm{m_t})}\rangle - \langle \mathrm{x}_{\mathrm{n,j}}^{(\mathrm{m_t})}\rangle\right)$ $\langle \mathrm{x}_{\mathrm{n,j}}^{(\mathrm{m_t})}\rangle = \langle \mathrm{y}_{\mathrm{n,j}}^{(\mathrm{m_t})}\rangle - \frac{1}{\boldsymbol{\xi}_{n,:}}\mathbb{E}_{p(u)}\left[\mathcal{N}_u\left(\langle \mathrm{y}_{\mathrm{n,j}}^{(\mathrm{m_t})}\rangle - \langle \mathrm{y}_{\mathrm{n,i}}^{(\mathrm{m_t})}\rangle, 1\right)\right.$ $\left.\prod_{\mathrm{k}\neq\mathrm{i}\neq\mathrm{j}}\left(\Phi\left(u + \langle \mathrm{y}_{\mathrm{n,i}}^{(\mathrm{m_t})}\rangle - \langle \mathrm{y}_{\mathrm{n,k}}^{(\mathrm{m_t})}\rangle\right)\right)\right]$

Table 4.3: q distribution of $\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}$, where $\langle \mathbf{y}_{\mathrm{n},:}^{(\mathrm{m_t})}\rangle = \langle \mathbf{z}_{\mathrm{n},:}\rangle\langle \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\rangle + \langle \mathbf{b}^{(\mathrm{m})}\rangle$ and $\boldsymbol{\xi}_{n,:} = \mathbb{E}_{p(u)}\left[\prod_{\mathrm{j}\neq\mathrm{i}}\left(\Phi\left(u + \langle \mathrm{y}_{\mathrm{n,i}}^{(\mathrm{m_t})}\rangle - \langle \mathrm{y}_{\mathrm{n,j}}^{(\mathrm{m_t})}\rangle\right)\right)\right]$ assuming i corresponds to the index of the true category, $\mathrm{t}_{\mathrm{n}}^{(\mathrm{m})} = \mathrm{i}$. This distribution only affects the views modelled as categorical data. See Appendix A for further details.

The mean-field variational family is equivalent to (4.13), if we now consider that $\mathcal{M}_t$ corresponds to the categorical observations. With this, we obtain the mean-field update rules described in Table 4.3, where we included the only new term with respect to the standard mean-field updates in Table 2.3. As we can see in this table, given $\mathrm{t}_{\mathrm{n}}^{(\mathrm{m_t})}$, $q\left(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\right)$ corresponds to a truncated Gaussian.

As happened with the multidimensional binary extension, we can use a point estimate to determine the predictive formulation. This way the predicted categorical variable, $\mathrm{t}_{*}^{(\mathrm{m_t})}$, can be calculated as the argument that maximises $\langle \mathbf{x}_{*,:}^{(\mathrm{m_t})}\rangle$:

$$\mathrm{t}_{*}^{(\mathrm{m_t})} = \mathrm{i} \qquad \text{if} \qquad \langle \mathbf{x}_{*,\mathrm{i}}^{(\mathrm{m_t})}\rangle = \max_{\mathrm{d}=0,\ldots,\mathrm{D_{m_t}}-1} \left\{ \langle \mathbf{x}_{*,\mathrm{d}}^{(\mathrm{m_t})}\rangle \right\} \tag{4.17}$$

A full description of the predictive formulation for categorical data is included in Appendix A. However, as stated in the previous subsection, this can be replaced by the semi-supervised version presented below.

4.1.4 Semi-supervised SSHIBA

Here we propose an alternative algorithm formulation to work in a semi-supervised way. The proposed change in the formulation provides a basic tool to work with most neuroimaging scenarios: imputation of missing values. To do that, the model learns the predictive distribution of the unobserved input data using the r.v. learnt through the variational inference iterations. Besides, this implies that the model is capable of combining labelled and unlabelled data (for classification) while training, being able to impute the model output on the test data, and, consequently, not needing a predictive distribution. This, in combination with the multi-view framework, allows the model to combine the so-called input data in one or more views and the output data in another view, where the output view will be defined as a stack of the output training data and the missing values for the values corresponding to the test data.

Generative model

Suppose the m-th view corresponds to a real variable, then we define $\tilde{\mathbf{X}}^{(\mathrm{m})}$ as the set of data points with missing values. On the other hand, if the m-th view corresponds to a multi-dimensional binary variable or a categorical variable, we equivalently define $\tilde{\mathbf{T}}^{(\mathrm{m})}$ as the set of data points with missing values[4]. For this functionality, the graphical model is the same as seen before with the exception of the observed variable, which should have a white circle instead of a grey one for the unobserved samples.

Variational inference

Now, considering that some samples of some views are missing, to approximate the joint posterior distribution of the r.v. we need to include the r.v. associated to the missing data points ($\tilde{\mathbf{X}}^{(\mathrm{m})}$ or $\tilde{\mathbf{T}}^{(\mathrm{m})}$). Therefore, the mean-field variational family is

$$\begin{aligned} & p(\Theta, \tilde{\mathbf{T}}^{\{\mathcal{M}_\mathrm{t}\}}, \tilde{\mathbf{X}}^{\{\mathcal{M}_\mathrm{r}\}} | \, \mathbf{T}^{\{\mathcal{M}_\mathrm{t}\}}, \mathbf{X}^{\{\mathcal{M}_\mathrm{r}\}}) \approx q(\Theta) \\ = \; & q(\mathbf{Z}) \prod_{m_t \in \mathcal{M}_t} \left(q\left(\tilde{\mathbf{T}}^{(\mathrm{m_t})}\right) \prod_{\mathrm{n=1}}^{\mathrm{N}} q\left(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\right) \right) \prod_{m_r \in \mathcal{M}_r} q\left(\tilde{\mathbf{X}}^{(\mathrm{m_r})}\right) \\ & \prod_{\mathrm{m=1}}^{\mathrm{M}} \left(q\left(\mathbf{W}^{(\mathrm{m})}\right) q\left(\boldsymbol{\alpha}^{(\mathrm{m})}\right) q\left(\tau^{(\mathrm{m})}\right) q\left(\boldsymbol{\gamma}^{(\mathrm{m})}\right) \right). \end{aligned} \tag{4.18}$$

where $\tilde{\mathbf{X}}^{(\mathrm{m})}$ and $\tilde{\mathbf{T}}^{(\mathrm{m})}$ only need to be included for the missing samples. As happened before, Table 4.4 has the new distributions for all factors on (4.18) and a detailed explanation of the mean-field update rules is available in Appendix A.

4.1.5 Kernelised SSHIBA

To complete the proposed model, here we present a generative formulation for the kernel representation of the observed data, in a similar way to kernel MVA methods. This extension is

[4] For the sake of simplicity, we consider in this case that the categorical data is a column matrix rather than a vector.

Version	Variable	q distribution	Parameters
Regression	$\tilde{\mathbf{X}}^{(\mathrm{m})}$	$\prod_{\mathrm{n}=1}^{\mathrm{N}} \mathcal{N}\left(\tilde{\mathbf{x}}_{\mathrm{n},:}^{(\mathrm{m})} \lvert \langle \tilde{\mathbf{x}}_{\mathrm{n},:}^{(\mathrm{m})} \rangle, \Sigma_{\tilde{\mathbf{X}}^{(\mathrm{m})}}\right)$	$\langle \tilde{\mathbf{X}}^{(\mathrm{m})} \rangle = \langle \tilde{\mathbf{Z}} \rangle \langle \mathbf{W}^{(\mathrm{m})} \rangle^T$ $\Sigma_{\tilde{\mathbf{X}}^{(\mathrm{m})}} = \langle \tau^{(\mathrm{m})} \rangle^{-1} I_{\mathrm{D_m}}$
Multidimensional	$\tilde{\mathbf{T}}^{(\mathrm{m})}$	$\prod_{\mathrm{n}=1}^{\mathrm{N}} \mathcal{N}\left(\tilde{\mathbf{t}}_{\mathrm{n},:}^{(\mathrm{m})} \lvert \langle \tilde{\mathbf{t}}_{\mathrm{n},:}^{(\mathrm{m})} \rangle, \Sigma_{\tilde{\mathbf{T}}^{(\mathrm{m})}}\right)$	$\langle \tilde{\mathbf{T}}^{(\mathrm{m})} \rangle = \sigma\left(\langle \tilde{\mathbf{X}}^{(\mathrm{m})} \rangle\right)$ $\Sigma_{\tilde{\mathbf{T}}^{(\mathrm{m})}} = \frac{e^{\langle \tilde{\mathbf{X}}^{(\mathrm{m})} \rangle}}{\left(1+e^{\langle \tilde{\mathbf{X}}^{(\mathrm{m})} \rangle}\right)^2}$
Categorical	$\tilde{\mathbf{t}}^{(\mathrm{m})}$	$\prod_{\mathrm{n}=1}^{\mathrm{N}} \mathcal{N}\left(\tilde{\mathrm{t}}_{\mathrm{n}}^{(\mathrm{m})} \lvert \langle \tilde{\mathrm{t}}_{\mathrm{n}}^{(\mathrm{m})} \rangle, \Sigma_{\tilde{\mathbf{t}}^{(\mathrm{m})}}\right)$	$\langle \tilde{\mathrm{t}}_{\mathrm{n}}^{(\mathrm{m})} \rangle = \langle \tilde{\mathrm{y}}_{\mathrm{n,j}}^{(\mathrm{m_t})} \rangle - \frac{1}{\boldsymbol{\xi}_{n,:}} \mathbb{E}_{p(u)} \left[\mathcal{N}_u \left(\langle \tilde{\mathrm{y}}_{\mathrm{n,j}}^{(\mathrm{m_t})} \rangle - \langle \tilde{\mathrm{y}}_{\mathrm{n,i}}^{(\mathrm{m_t})} \rangle, 1 \right)\right.$ $\left.\prod_{k \neq i \neq j} \left(\Phi \left(u + \langle \tilde{\mathrm{y}}_{\mathrm{n,i}}^{(\mathrm{m_t})} \rangle - \langle \tilde{\mathrm{y}}_{\mathrm{n,k}}^{(\mathrm{m_t})} \rangle \right)\right)\right]$

Table 4.4: q distribution of the r.v. of the graphical model on the semi-supervised scheme for the three heterogeneous data types. The table shows the r.v. distributions along with their parameters, where $\langle \tilde{\mathbf{y}}_{\mathrm{n},:}^{(\mathrm{m_t})} \rangle = \langle \tilde{\mathbf{z}}_{\mathrm{n},:} \rangle \langle \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}} \rangle + \langle \mathbf{b}^{(\mathrm{m})} \rangle$. See Appendix A for further details.

specially useful for neuroimaging problems where we work with high dimensional data, as it allows to work in the dual space, where the number of parameters to learn depends on the number of samples. However, although working with kernels also allows to include non-linearities in the model, we will use linear kernels for neuroimaing problems to conserve the linear variable relations and the model interpretability.

Generative model

Let us start by revisiting the generative model, defining the latent variables $\mathbf{z}_{\mathrm{n},:} \sim \mathcal{N}(0, \mathbf{I}_{\mathrm{K_c}})$ which are combined with some dual variables $\mathbf{A}^{(\mathrm{m})} \in \mathbb{R}^{\mathrm{N} \times \mathrm{K_c}}$ and some Gaussian noise with precision $\tau^{(\mathrm{m})} \sim \Gamma\left(a^{\tau^{(\mathrm{m})}}, b^{\tau^{(\mathrm{m})}}\right)$ to generate a kernelised representation of each data sample, $\mathbf{k}_{\mathrm{n},:}^{(\mathrm{m})}$. That is, each data sample $\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}$ will be represented by a vector $\mathbf{k}_{\mathrm{n},:}^{(\mathrm{m})}$ consisting of the kernel between $\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}$ and the training data, such that $\mathbf{k}_{\mathrm{n},:}^{(\mathrm{m})} = [K(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}, \mathbf{x}_{1,:}^{(\mathrm{m})}), \ldots, K(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}, \mathbf{x}_{\mathrm{N},:}^{(\mathrm{m})})]$ is the kernel function induced by the mapping function $\phi(\cdot)$, where $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}')$. However, it may be desirable to have a compact kernelised representation of $\mathbf{k}_{\mathrm{n},:}^{(\mathrm{m})}$ using only a subset of the training data (called Relevance Vectors, RV) reducing the model complexity. Although these vectors can be selected randomly, another extension of the model to perform automatic RV selection will be presented in the next section.

Figure 4.5 presents the graphical model of Kernelised SSHIBA (KSSHIBA), combining kernelised and standard observations in different views. This way, the views that are defined using the observed data (views in the primal space) follow the same equations previously explained in this chapter and in the Bayesian FA, Section 2.2. Following this graphic model, matrix $\mathbf{A}^{(\mathrm{m})}$ defines the linear projections learnt by the model and is combined with the same ARD prior defined in equation (2.69) to force zeroes in the latent factors

$$\mathbf{a}_{:,\mathrm{k}}^{(\mathrm{m})} \sim \mathcal{N}\left(0, \left(\alpha_{\mathrm{k}}^{(\mathrm{m})}\right)^{-1} I_{K_c}\right) \tag{4.19}$$

$$\alpha_k^{(m)} \sim \Gamma\left(a^{\alpha^{(m)}}, b^{\alpha^{(m)}}\right), \tag{4.20}$$

in a way that, when combined with the latent projection $\mathbf{z}_{\mathrm{n},:}$, it induces sparsity in the latent factors and, thus, their automatic selection (Tipping, 2001).

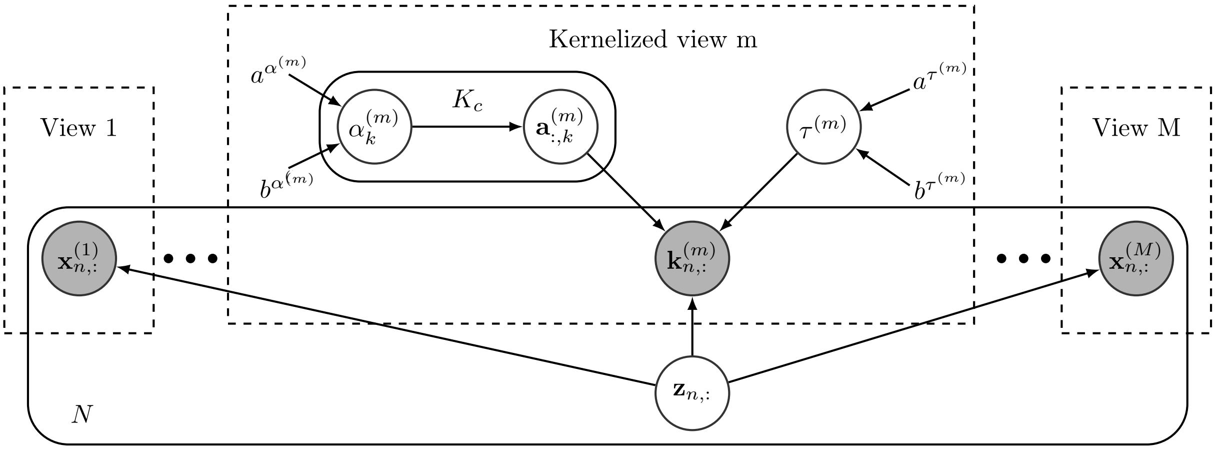
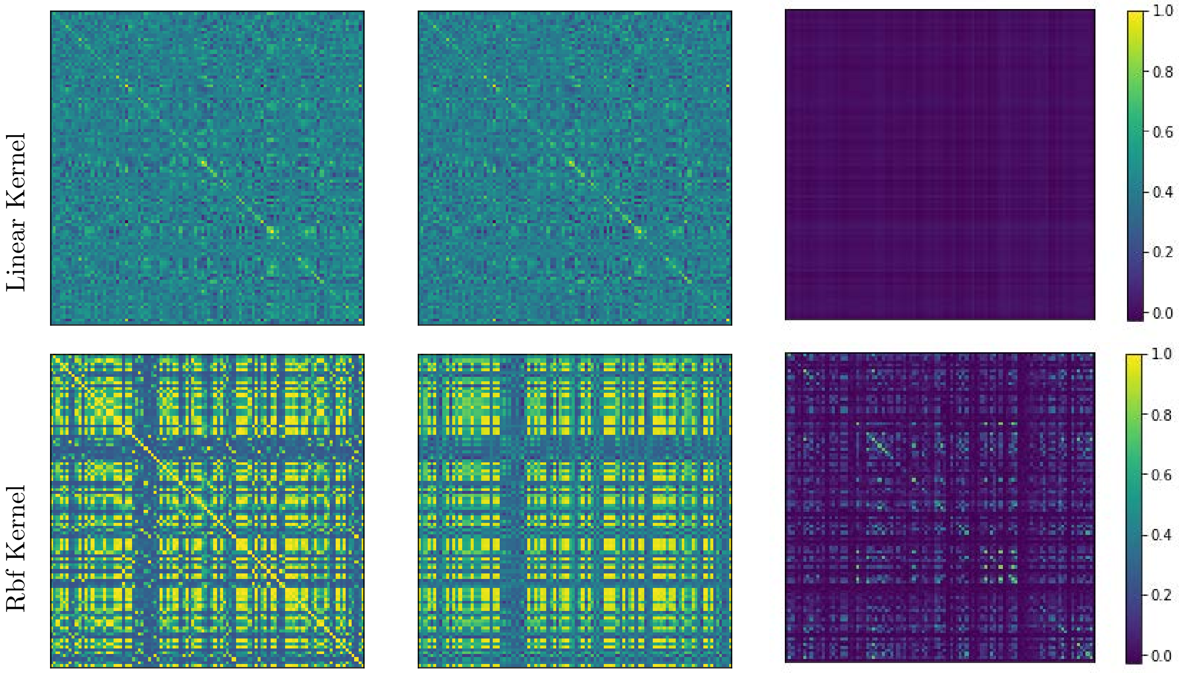


Figure 4.5: Graphic model of Kernelised SSHIBA (KSSHIBA).

The advantages of adding this kernelised formulation are two-fold: (1) It exploits some non-linear relationships between the data. (2) It reduces the number of parameters that need to be estimated in problems with $\mathrm{D_m} >> \mathrm{N}$. This way, we can combine in a latent projection $\mathbf{z}_{\mathrm{n},:}$ the information related to the primal and the dual views, as well as their linear and non-linear relations. Furthermore, we can add kernelised views to the model, while maintaining the rest of the functionalities previously discussed.



(a) Original kernel matrix (b) Reconstructed matrix (c) Reconstruction error

Figure 4.6: Example of the generative properties of KSSHIBA to reconstruct a complete kernel matrix.

On another note, the fact that we are treating the kernel matrix as an observation imply that the objective of the generative model is to optimise the reconstruction error of the kernel matrix instead of the original data. However, this does not unequivocally ensure a valid positive semi-definite kernel matrix. One can see that, in the proposed model, the prior distribution over the kernel representations is independent over n, which is not the most appropriate when each $\mathbf{k}_{\mathrm{n},:}^{(\mathrm{m})}$ models the relationships between $\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}$ and all the data. A more representative model would have to include this dependence over the distribution of $\mathbf{K}^{(\mathrm{m})}$. To do so, we have analysed different formulations to model non-independent noise; by defining the noise as an Inverse-Wishart by having a full rank covariance (Wang, 2007; Klami and Kaski, 2007) or modelling the covariance as a product of two low rank matrices (Murphy, 2012). However, these formulations are more complex and limit the rest of the properties and extensions of SSHIBA. Whereas the proposal might not seem the most appropriate, it is the simplest and, in fact, accurately reconstructs the kernel data representation. Figure 4.6 proves this with a toy example; specifically, we defined both linear and Radial Basis Function (RBF) kernel matrices and used them as observations to train the model. After that, we obtained the reconstructed matrix making use of mean of the posterior distribution of $\mathbf{z}_{\mathrm{n},:}$, proving that they are accurately reconstructed. This implies that, despite the approximation, the model is capable of inherently model the data dependencies.

Variational inference

At this point, we can define the fully factorised variational family chosen to approximate the posterior distribution

$$\begin{aligned}
&p(\Theta|\,\mathbf{K}^{\{\mathcal{M}_\mathrm{k}\}},\mathbf{X}^{\{\mathcal{M}_\mathrm{x}\}}) \approx q(\Theta)\\
&= \prod_{m_k\in\mathcal{M}_k}\left(q\left(\mathbf{A}^{(\mathrm{m_k})}\right)\right)\prod_{m_x\in\mathcal{M}_x}\left(q\left(\mathbf{W}^{(\mathrm{m_x})}\right)\right)\prod_{\mathrm{m=1}}^{\mathrm{M}}\left(q\left(\tau^{(\mathrm{m})}\right)\prod_{\mathrm{k=1}}^{\mathrm{K_c}}q\left(\alpha_\mathrm{k}^{(\mathrm{m})}\right)\right)\prod_{\mathrm{n=1}}^{\mathrm{N}}q(\mathbf{z}_{\mathrm{n},:})
\end{aligned}\tag{4.21}$$

where $\mathcal{M}_k$ represents the set of kernelised views, $\mathcal{M}_x$ is the set of views with the original data and $\mathbf{K}^{\{\mathcal{M}_\mathrm{k}\}}$ contains a stacked version of each sample $\mathbf{k_{n,:}}^{\{\mathcal{M}\}}$ of dimension $\mathrm{N}\times\mathrm{D_m}$, with all the kernel representations.

The fact that we are redefining the observations with kernel representations imply that the SSHIBA distributions are mostly equivalent to those presented in Table 4.1, having to reinterpret matrix $\mathbf{W}^{(\mathrm{m})}$ and change the observation matrices. By using the mean-field approximation, we obtain the updates included in Table 4.5. The posterior distributions of all model parameters and latent projections are approximated using variational inference with a fully factorised posterior.

The following subsections present different functionalities that are specific for kernelised views. These can be combined either with some previous extensions, such as semi-supervised formulation, or between them depending on the problem needs due to their modular structure.

4.1.5.1 Automatic Bayesian relevance vector selection

Equivalently to the SSHIBA extension that allows to include FS (Section 4.1.1), here we can include a double ARD prior to automatically reduce the number of latent factors but, in this scenario, instead of selecting variables, to select a training subset to obtain a compact kernelised representation, RVs. This way, we have that we can redefine the dual variable $\mathbf{A}^{(\mathrm{m})}$,

$$\mathrm{a}_{\mathrm{n,k}}^{(\mathrm{m})} \sim \mathcal{N}\left(0,\left(\gamma_\mathrm{n}^{(\mathrm{m})}\,\alpha_\mathrm{k}^{(\mathrm{m})}\right)^{-1}\right),\tag{4.22}$$

Variable	q^* **distribution**	**Parameters**
$\mathbf{z}_{\mathrm{n},:}$	$\mathcal{N}(\mathbf{z}_{\mathrm{n},:} \lvert \langle \mathbf{z}_{\mathrm{n},:} \rangle, \Sigma_{\mathbf{Z}})$	$\langle \mathbf{z}_{\mathrm{n},:} \rangle = \sum_{\mathrm{m}=1}^{\mathrm{M}} \left(\langle \tau^{(\mathrm{m})} \rangle \, \mathbf{K}^{(\mathrm{m})} \langle \mathbf{A}^{(\mathrm{m})} \rangle \right) \Sigma_{\mathbf{Z}}$ $\Sigma_{\mathbf{Z}}^{-1} = \left(I + \sum_{\mathrm{m}=1}^{\mathrm{M}} \left(\langle \tau^{(\mathrm{m})} \rangle \langle \mathbf{A}^{(\mathrm{m})^{\mathrm{T}}} \mathbf{A}^{(\mathrm{m})} \rangle \right) \right)$
$\mathbf{A}^{(\mathrm{m})}$	$\prod_{\mathrm{n}=1}^{\mathrm{N}} \left(\mathcal{N} \left(\mathbf{a}_{\mathrm{n},:}^{(\mathrm{m})} \lvert \langle \mathbf{a}_{\mathrm{n},:}^{(\mathrm{m})} \rangle, \Sigma_{\mathbf{A}^{(\mathrm{m})}} \right) \right)$	$\langle \mathbf{a}_{\mathrm{n},:}^{(\mathrm{m})} \rangle = \langle \tau^{(\mathrm{m})} \rangle \, \mathbf{K}^{(\mathrm{m})^{\mathrm{T}}} \langle \mathbf{Z} \rangle \Sigma_{\mathbf{A}^{(\mathrm{m})}}$ $\Sigma_{\mathbf{A}^{(\mathrm{m})}}^{-1} = \left(\mathrm{diag}(\langle \boldsymbol{\alpha}^{(\mathrm{m})} \rangle) + \langle \tau^{(\mathrm{m})} \rangle \langle \mathbf{Z}^{\mathrm{T}} \, \mathbf{Z} \rangle \right)$
$\alpha_{\mathrm{k}}^{(\mathrm{m})}$	$\Gamma \left(\alpha_{\mathrm{k}}^{(\mathrm{m})} \lvert a_{\alpha_{\mathrm{k}}^{(\mathrm{m})}}, b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}} \right)$	$a_{\alpha_{\mathrm{k}}^{(\mathrm{m})}} = \frac{D_m}{2} + a^{\boldsymbol{\alpha}^{(\mathrm{m})}}$ $b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}} = b^{\boldsymbol{\alpha}^{(\mathrm{m})}} + \frac{1}{2} \langle \mathbf{A}^{(\mathrm{m})^{\mathrm{T}}} \mathbf{A}^{(\mathrm{m})} \rangle_{k,k}$
$\tau^{(\mathrm{m})}$	$\Gamma \left(\tau^{(\mathrm{m})} \lvert a_{\tau^{(\mathrm{m})}}, b_{\tau^{(\mathrm{m})}} \right)$	$a_{\tau^{(\mathrm{m})}} = \frac{D_m N}{2} + a^{\tau^{(\mathrm{m})}}$ $b_{\tau^{(\mathrm{m})}} = b^{\tau^{(\mathrm{m})}} + \frac{1}{2} \Bigg(\sum_{\mathrm{n}=1}^{\mathrm{N}} \sum_{\mathrm{n'}=1}^{\mathrm{N}} \mathrm{k}_{\mathrm{n,n'}}^{(\mathrm{m})^2}$ $-2 \, \mathrm{Tr} \left\{ \langle \mathbf{A}^{(\mathrm{m})} \rangle \langle \mathbf{Z}^{\mathrm{T}} \rangle \, \mathbf{K}^{(\mathrm{m})} \right\}$ $+ \, \mathrm{Tr} \left\{ \langle \mathbf{A}^{(\mathrm{m})^{\mathrm{T}}} \mathbf{A}^{(\mathrm{m})} \rangle \langle \mathbf{Z}^{\mathrm{T}} \, \mathbf{Z} \rangle \right\} \Bigg)$

Table 4.5: Updated rules of q distribution obtained by a mean-field approximation for the different variables of KSSHIBA model.

so that $\alpha_{\mathrm{k}}^{(\mathrm{m})}$ keeps inducing sparsity in the rows to automatically select the latent factors, while

$$\gamma_{\mathrm{n}}^{(\mathrm{m})} \sim \Gamma \left(a^{\gamma^{(\mathrm{m})}}, b^{\gamma^{(\mathrm{m})}} \right) \tag{4.23}$$

forces the column sparsity, to automatically select the RVs that are needed for an accurate data representation. Find the updated mean-field distributions in Table 4.6, where the only change (with respect to Table 4.5) corresponds to the variables in the double ARD prior, $\boldsymbol{\gamma}^{(\mathrm{m})}$ and $\mathbf{A}^{(\mathrm{m})}$.

Variable	q^* **distribution**	**Parameters**
$\mathbf{A}^{(\mathrm{m})}$	$\prod_{\mathrm{n}=1}^{\mathrm{N}} \mathcal{N} \left(\mathbf{a}_{\mathrm{n},:}^{(\mathrm{m})} \lvert \langle \mathbf{a}_{\mathrm{n},:}^{(\mathrm{m})} \rangle, \Sigma_{\mathbf{a}_{\mathrm{n},:}^{(\mathrm{m})}} \right)$	$\langle \mathbf{A}^{(\mathrm{m})} \rangle = \langle \tau^{(\mathrm{m})} \rangle \, \mathbf{X}^{(\mathrm{m})^{\mathrm{T}}} \langle \mathbf{Z} \rangle \Sigma_{\mathbf{W}^{(\mathrm{m})}}$ $\Sigma_{\mathbf{a}_{\mathrm{n},:}^{(\mathrm{m})}}^{-1} = \mathrm{diag}(\langle \boldsymbol{\alpha}^{(\mathrm{m})} \rangle) \langle \gamma_{\mathrm{n}}^{(\mathrm{m})} \rangle + \langle \tau^{(\mathrm{m})} \rangle \langle \mathbf{Z}^{\mathrm{T}} \, \mathbf{Z} \rangle$
$\boldsymbol{\gamma}^{(\mathrm{m})}$	$\prod_{\mathrm{n}=1}^{\mathrm{N}} \Gamma \left(\gamma_{\mathrm{n}}^{(\mathrm{m})} \lvert a_{\gamma_{\mathrm{n}}^{(\mathrm{m})}}, b_{\gamma_{\mathrm{n}}^{(\mathrm{m})}} \right)$	$a_{\gamma_{\mathrm{n}}^{(\mathrm{m})}} = \frac{K_c}{2} + a^{\boldsymbol{\gamma}^{(\mathrm{m})}}$ $b_{\gamma_{\mathrm{n}}^{(\mathrm{m})}} = b^{\boldsymbol{\gamma}^{(\mathrm{m})}} + \frac{1}{2} \sum_{\mathrm{k}=1}^{\mathrm{K_c}} \langle \alpha_{\mathrm{k}}^{(\mathrm{m})} \rangle \langle \mathrm{a}_{\mathrm{n,k}}^{(\mathrm{m})} \, \mathrm{a}_{\mathrm{n,k}}^{(\mathrm{m})} \rangle$

Table 4.6: Updated q distribution for the automatic RV selection extension.

4.1.5.2 Automatic feature-relevance determination

Note that by using kernelised observations we lose the feature selection capabilities, but we can modify the kernel definition to automatically determine the feature relevance. To do so, we can include an ARD over the kernel, $\mathbf{K}^{(\mathrm{m})}$, to find the feature importance. Suppose the kernel we are using is a standard RBF, then

$$\mathrm{k}_{\mathrm{n,n'}}^{(\mathrm{m})} \;=\; \exp\left(-\sum_{\mathrm{d}=1}^{\mathrm{D_m}}\left(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})}-\mathrm{x}_{\mathrm{n',d}}^{(\mathrm{m})}\right)^2\lambda_{\mathrm{d}}^{(\mathrm{m})}\right), \tag{4.24}$$

where, for each input feature, we have included a term $\lambda_{\mathrm{d}}^{(\mathrm{m})}$ that learns the relevance corresponding to each feature. Conversely, we can optimise $\boldsymbol{\lambda}^{(\mathrm{m})} = [\lambda_1^{(\mathrm{m})}, \ldots, \lambda_{\mathrm{D_m}}^{(\mathrm{m})}]$ by maximising the variational lower bound obtained in the mean-field update. To calculate the optimum lower bound we first determine the only terms that depend on the ARD for feature relevance determination, $\mathbb{E}_q\left[\ln\left(p\left(\mathbf{K}^{(\mathrm{m})}\,|\Theta\right)\right)\right]$, which leaves the lower bound as

$$LB \;=\; -\frac{\langle\tau^{(\mathrm{m})}\rangle}{2}\sum_{\mathrm{n}=1}^{\mathrm{N}}\sum_{\mathrm{n'}=1}^{\mathrm{N}}\left(\mathrm{k}_{\mathrm{n,n'}}^{(\mathrm{m})\,2} - 2\,\mathrm{k}_{\mathrm{n,n'}}^{(\mathrm{m})}\langle\mathbf{a}_{\mathrm{n',:}}^{(\mathrm{m})}\rangle\langle\mathbf{z}_{\mathrm{n,:}}^{\mathrm{T}}\rangle + \langle\mathbf{a}_{\mathrm{n',:}}^{(\mathrm{m})^{\mathrm{T}}}, \mathbf{a}_{\mathrm{n',:}}^{(\mathrm{m})}\rangle\langle\mathbf{z}_{\mathrm{n,:}}^{\mathrm{T}}, \mathbf{z}_{\mathrm{n,:}}\rangle\right) \tag{4.25}$$

After this, we can alternate between the variational updates and a gradient ascend algorithm to calculate the maximum of (4.25) w.r.t. $\boldsymbol{\lambda}^{(\mathrm{m})}$. Besides, we can combine the learnt variables, $\boldsymbol{\lambda}^{(\mathrm{m})}$, with a threshold to automatically eliminate the features that are not relevant for the problem.

4.1.5.3 Multiple kernel learning

An additional functionality developed through the inclusion of kernelised observations is the ability to automatically learn a multiple kernel learning (MKL SSHIBA) formulation. This is obtained by including different kernels in different views. Let us analyse the update rule of the latent variables in this scenario

$$\langle\mathbf{z}_{\mathrm{n,:}}\rangle = \sum_{\mathrm{m}=1}^{\mathrm{M}}\langle\tau^{(\mathrm{m})}\rangle\,\mathbf{K}^{(\mathrm{m})}\langle\mathbf{A}^{(\mathrm{m})}\rangle\Sigma_{\mathbf{Z}}. \tag{4.26}$$

This equation shows that, when we have different kernels in different views, the latent factors are calculated as a linear combination of each of these kernels. Furthermore, the Bayesian nature of the model implies that the mixing parameters of the MKL scheme (in our case, $\tau^{(\mathrm{m})}$, $\mathbf{A}^{(\mathrm{m})}$ and $\Sigma_{\mathbf{Z}}$) are automatically learnt in the inference process.

A great number of MKL models either rely on two step optimisation processes (Fung et al., 2004) or on heuristics (Qiu and Lane, 2008; de Diego et al., 2010) to determine the kernel parameters. However, our proposed extension calculates these parameters in the variational inference process, while conserving the functionalities of the SSHIBA formulation, i.e. semi-supervised learning, feature selection and RV selection.

In the particular case of neuroimaging problems, we could use the multi-view structure of the model to combine different data with either the dual or primal formulation. This allows to combine demographic data that tends to have less features with brain imaging data without needing to compromise to either formulation. Moreover, we can also model different Regions Of Interest (ROIs) in separate views and use the dual or primal formulation depending on the number of features in each region, as we will see in the experimental results.

4.2 Results

In this section we will explore the performance of SSHIBA in different scenarios to demonstrate the ability of SSHIBA to adapt to the different particularities of real data sets. Specifically, we will divide the experiments in two distinctive subsections: Benchmark datasets and Neuroimaging datasets. The objective is to firstly study the performance of the model on well-known databases where we can independently explore each model extension in comparison to state-of-the-art methods, and later explore these functionalities on neuroimaging problems where we will use specific SSHIBA extensions depending on the problem needs.

The project has been developed using *Python* and most baselines were accessible in packages from *Scikit-learn* Pedregosa et al. (2011). You can find exemplary notebooks openly available in these GitHub repositories: SSHIBA and KSSHIBA, with the complete code of both the SSHIBA model and its modifications to work with kernels, as well as exemplary results to show how to use the models. The version that works with kernelised observations also uses *Pytorch* and *Adam* to calculate the kernel ARD.

For all the experiments presented with SSHIBA, we decided to use the automatic latent factor selection, or pruning. This implies that we set a criteria during inference process for which we remove the k-th column of $\mathbf{W}^{(\mathrm{m})}$, $\forall m$, if all the elements of $\mathbf{w}^{(\mathrm{m})}_{:,\mathrm{k}}$, across all views, are lower than the pruning threshold, set to 10^{-6}. Furthermore, we determine the convergence of the model using the evolution of the lower bound. In particular, the utilised criteria is $\frac{1}{100}\sum([LB[-101]...LB[-2]]) > LB[-1](1-10^{-8})$ where $LB[-1]$ is the variational lower bound value at the last iteration and $\frac{1}{100}\sum([LB[-100]...LB[-2]])$ is the mean value of the lower bound from the past 100 iterations prior to the last. We also set that if there is no convergence after 50.000 iterations the algorithm stops. As advised by Klami et al. (2013), we initialise the model by sampling from the priors, train the model until convergence and then choose the solution with the best lower bound. In our case we used 10 initialisations.

4.2.1 Benchmark databases

The objective of this first set of experiments is to present the different functionalities and extensions of SSHIBA. For that purpose, we include results on datasets from different domains and tasks with the main purpose of analysing the performance of the model on standard databases before studying its applications to neuroimaging.

4.2.1.1 Database description

Due to the heterogeneous nature of the proposed model, we decided to include different databases from various contexts (number of features, domains, types of variables,...) to prove its potential. Specifically, we used databases which the different model functionalities we wanted to analyse can exploit. The scenarios are:

- **Multi-label datasets**: We decided to use some multi-label datasets from the Mulan repository (Tsoumakas et al., 2011), comprising three multi-label problems: *yeast* (Elisseeff and Weston, 2002), *scene* (Boutell et al., 2004) and *birds* (Briggs et al., 2013). For these problems, we decided to stuck the different labels and include them in a single view and the rest of the features are included in another view. These databases are used to test the multi-label and semi-supervised extensions of the model.

- **Categorical data**: To analyse categorical data we used *AVIRIS* database (Baumgardner et al., 1992). This database is used to test the categorical and semi-supervised extensions of the model.
- **Multi-dimensional regression**: Regarding regression problems, we used 9 multi-dimensional regression datasets also available in the Mulan repository (Karalič and Bratko, 1997; Džeroski et al., 2000; Spyromitros-Xioufis et al., 2016). These databases are used to test the kernelised extension together with the automatic selection of RV extensions of the model.
- **Image classification**: In order to test the properties of the model in an image classification framework, we also included the Labeled Faces in the Wild (LFW) database (Huang et al., 2007). We have preprocessed these images by cropping and resizing, this way, we finally obtained 60 × 40 pixels images. Furthermore, we used two different configurations of the database to test feature selection extension of the model:
  - **Face recognition**: The objective is to correctly identify the images of 7 different people. We used the 7 most populated classes in the database. This version will be referred as LFW.
  - **Multi-label attributes**: Here we need to predict the attributes of the same database specified in Kumar et al. (2009). These attributes consist of physical descriptions of the images, such as gender or hair colour. This version will be referred as LFWA.

  Furthermore, we also incorporated the results obtained by KSSHIBA on the *AR10P* database, also composed by 60 × 40 pixels images, which is available in the Feature Selection Repository. The results on these databases are used for the interpretability analysis of both the learnt latent space and the FS extension.
- **Multiple view learning**: Finally, to study the effect of multiple view learning with kernel representations we used three categorical databases: *Arrhythmia* and *Landsat* from the UCI repository (Dua and Graff, 2017) and *Fashion MNIST* (Xiao et al., 2017). We decided to include each feature type in a different view for the first two databases and different types of kernels in different views for *Fashion MNIST* to test the MKL model extension.

You can find the summarised information of the presented databases in Table 4.7.

For the multi-label databases from the Mulan repository the data is already partitioned into training and test datasets, so we have used these previously defined partitions. For *AVIRIS* and *LFW* we have split the data into 70% train / 30% test partitions. For the remaining datasets, the regression databases from the Mulan repository and the databases used for MKL, we decided to apply a 10-fold train and test CV for these databases.

4.2.1.2 Baseline or state-of-art methods

As this section aims to present the different functionalities of SSHIBA, we will use different state-of-the-art algorithms as baselines to compare the obtained results. We decided to use:

Scenario	Database	Samples	Features		Labels/Tasks
	yeast	2,417	103		14
Multi-label	*scene*	2,407	294		6
	birds	645	260		19
Categorical	*AVIRIS*	21,025	220		16
	LFW	1,277	2,400		7
Image	*LFWA*	22,343	2,400		73
	AR10P	130	2,400		10
	at1pd	337	411		6
	at7pd	296	411		6
	oes97	334	263		16
	oes10	403	298		16
Multi-regression	*edm*	154	16		2
	jura	359	15		3
	wq	1,060	16		14
	enb	768	8		2
	slump	103	7		3
		Kernel 1	Kernel 2		
	Arrhythmia	452	15	264	2
MKL	*Landsat*	6,435	4	4	6
	Fashion MNIST	1,000	784	784	10

Table 4.7: Summary of the characteristic of the analysed databases used in this work. The first 7 correspond to the multi-label and categorical databases, the next 9 to the regression problems and the last 3 to the MKL problems. Kernels 1 and 2 are constructed either with different data or different transformations combined as input data.

- Support Vector Regression (**SVR**) and Linear Regression (**LR**) for regression tasks. Both of these algorithms are widely used for common regression problems. We use their linear or kernel version depending on whether we are comparing to SSHIBA or KSSHIBA.

- Equivalently, for classification problems, we used Logistic Regression (**LogR**) and Support Vector Machines (**SVM**) as reference methods.

- Canonical Correlation Analysis (**CCA**) as a reference MVA supervised algorithm used for FE. We use it due to its relations in properties and equations to the proposed model. In this case, due to its supervised nature, CCA is used both as a predictor, as well as a FE method combined with a regressor or classifier.

- Principal Component Analysis (**PCA**) as another reference FE MVA algorithm. Unlike CCA this algorithm is unsupervised, so it needs to be combined with a regressor or classifier to compute predictions.

- Multi-Layer Perceptron (**MLP**) with one hidden layer as a neural network that carries out feature extraction (in its hidden layer) as well as the prediction.

- Manifold Relevance Determination (**MRD**) for regression tasks. This algorithm, presented in (Damianou et al., 2012), is a shared GPLVM which has similar extensions to KSSHIBA, like working with kernels, using ARD for RVs selection, working with multiview data and finding latent representations.

- As final model reference, we have also included the original Bayesian Inter-Battery Factor Analysis (**BIBFA**) (Klami et al., 2013) formulation, to evaluate the improvement obtained through the proposed extensions. As this model is not adapted for classification in their paper, we have considered its regression model and included a threshold for classification problems.

We decided to validate the hyperparameters of the regressors and classifiers using a 10-fold CV, where we validate the regularisation parameter of logistic regression, the number of neurons in the MLP and the C and γ parameters of the SVM. Regarding the FE algorithms, we set the number of factors, $\mathrm{K_c}$ used by PCA to those that explain 95% of the variance, while CCA set it to $\mathrm{D_M} - 1$ and $\mathrm{D_M}$ for classification and regression respectively, where $\mathrm{D_M}$ is the number of classes of the output view. For the selection of the RVs in the kernel extension, we validate the optimum number of selected RVs using the following percentages: $1\%, 2\%, 3\%, 4\%, 5\%, 10\%, \ldots, 100\%$. For the MRD model, we used the available library in *Matlab* (Damianou et al., 2012), setting the number of latents to twice the number of tasks ($2 * \mathrm{D_M}$). We use the RBF kernel with ARD and set the number of model optimisation iterations to 100 due to the long computational time required to train. Finally, as scoring method we use R^2 for regression problems and balanced multiclass Area Under the Curve (AUC) for the classification problems.

4.2.1.3 Evaluation over heterogeneous data

One of the main advantages of the proposed model is its ability to model heterogeneous data. For this reason, here we present results on three types of databases. First, we use *yeast*, *scene*, *birds* (multi-label), and *AVIRIS* (categorical) to evaluate the prediction performance of modelling the output view as multi-label/categorical. In these scenarios we work with two views, an input view with $\mathrm{D_1}$ features and an output view with $\mathrm{D_2}$ features.

	SSHIBA	BIBFA	CCA	CCA+LR	PCA+LogR	MLP	LogR
yeast	0.66	**0.69**	0.61	0.66	0.68	0.61	0.67
	66	66	13	13	73	300	
scene	**0.92**	0.90	0.88	0.87	**0.92**	0.82	**0.92**
	137	119	5	5	121	900	
AVIRIS	**0.89**	**0.89**	0.88	**0.89**	0.81	0.85	**0.89**
	197	197	72	72	252	50	
birds	**0.83**	0.67	0.56	0.56	0.82	0.68	**0.83**
	75	10	18	18	87	100	

Table 4.8: Results of the predictive SSHIBA and the different methods under study on multi-label and categorical databases. Results include the performance in terms of AUC and the number $\mathrm{K_c}$ of latent factors.

Table 4.8 shows the performance as well as the number of used latent factors of the analysed algorithms in these classification tasks. The results show that SSHIBA provides a performance in terms of AUC equivalent to the rest of the baselines while carrying out a dimensionality reduction. Moreover, our initial experiments proved that by modelling the output data in *AVIRIS* database as categorical against binarising the data and treating them as multi-label leads to a slight AUC improvement.

	KSSHIBA	KSSHIBA $\mathrm{K_c = D_2}$	KCCA	KCCA + LR	KPCA + LR	MLP	MRD	SVR
at1pd	$\mathbf{0.79 \pm 0.09}$	0.78 ± 0.09	0.45 ± 0.05	0.75 ± 0.11	0.67 ± 0.12	0.77 ± 0.11	0.67 ± 0.07	0.01
	46 ± 6	6	6	6	22 ± 10	100	12	±0.05
at7pd	0.50 ± 0.18	0.52 ± 0.13	0.24 ± 0.05	$\mathbf{0.57 \pm 0.16}$	0.39 ± 0.19	0.35 ± 0.69	0.48 ± 0.12	0.01
	14 ± 6	6	6	6	21 ± 1	100	12	±0.03
oes97	$\mathbf{0.71 \pm 0.10}$	0.69 ± 0.10	0.30 ± 0.08	0.36 ± 0.09	0.45 ± 0.20	0.58 ± 0.21	0.34 ± 0.07	0.39
	17 ± 6	16	16	16	12 ± 7	100	32	±0.10
oes10	$\mathbf{0.82 \pm 0.05}$	0.80 ± 0.07	0.35 ± 0.17	0.43 ± 0.12	0.59 ± 0.15	0.76 ± 0.08	0.38 ± 0.07	0.48
	16 ± 5	16	16	16	14 ± 7	100	32	±0.12
edm	$\mathbf{0.51 \pm 0.18}$	0.21 ± 0.09	0.26 ± 0.18	0.18 ± 0.26	0.38 ± 0.19	0.26 ± 0.21	-0.17 ± 0.45	0.35
	30 ± 11	2	2	2	16 ± 5	100	6	±0.19
jura	$\mathbf{0.62 \pm 0.08}$	0.30 ± 0.10	0.11 ± 0.08	0.18 ± 0.15	0.38 ± 0.11	0.61 ± 0.06	0.57 ± 0.06	0.60
	21 ± 3	3	3	3	23 ± 1	100	6	±0.05
wq	$\mathbf{0.14 \pm 0.01}$	0.12 ± 0.01	-0.01 ± 0.01	-0.01 ± 0.01	0.09 ± 0.02	0.13 ± 0.03	-0.35 ± 0.08	0.08
	76 ± 9	14	14	14	29 ± 1	29 ± 1 100	28	±0.02
enb	$\mathbf{0.99 \pm 0.01}$	0.86 ± 0.02	0.96 ± 0.01	0.98 ± 0.01	0.86 ± 0.01	$\mathbf{0.99 \pm 0.08}$	0.91 ± 0.01	**0.99**
	118 ± 4	2	2	2	13 ± 1	100	4	**±0.01**

Table 4.9: Results on multitask databases of KSSHIBA and the baselines. The white subrow represents the mean and standard deviation of R^2 score and the gray subrow the number of effective latent factors found.

On the other hand, Table 4.9 includes the results obtained on the fourth type of studied problem, regression, which we analysed with the kernel version of SSHIBA in two different scenarios: (1) when the model automatically learns the number of latent factors $\mathrm{K_c}$ using the ARD prior; (2) when we set $\mathrm{K_c}$ to $\mathrm{D_2}$, which corresponds to the maximum number of latent factors CCA can have. Besides, we also included the performance obtained by the kernel version of the baselines, namely, KCCA, KPCA, MRD, SVR and MLP. By looking at the R^2 score obtained by the different methods, we can conclude that KSSHIBA consistently outperforms the baselines in most datasets, highlighting *edm* and *oes97* where the performance is substantially greater. Furthermore, this performance proves to be remarkable considering we are including a noticeable dimensionality reduction projecting in a latent space. Meanwhile, the results obtained by the version of KSSHIBA in which we set the number of latent factors to the number of output tasks shows that we could impose a more restrictive pruning criteria for the latent factors without considerably degrading the results[5].

[5] KSSHIBA with the number of latent factors set to the number of output tasks has a considerably worse performance in databases *edm, jura* and *enb*, where the number of latent factors is greatly reduced (2 or 3 latent factors).

4.2.1.4 Missing data imputation

Another common problem that can be found in real databases is the absence of some measurements for certain samples. These are usually referred as missing values. Specifically, neuroimaging problems often suffer from this lack of information due to the cost of certain medical tests or the fact that patients do not fully follow their study.

To test the performance of SSHIBA in this scenario, we have randomly eliminated 50% of the features from data samples and we used the semi-supervised formulation to impute random patterns of missing values on the data on the four databases analysed in Table 4.8. To fairly compare the imputation techniques, we decided to always use SSHIBA as a classifier and use different imputation methods to estimate the missing values with the feature mean, median, most frequent value or, with our approach, the mean of the predictive distribution learnt by SSHIBA.

As the semi-supervised version of SSHIBA can replace the predictive formulation, we also compared the results obtained using its predictive formulation. Although both prediction methods are supposed to provide similar results, the fact that the semi-supervised version uses all the available data to learn the model parameters may lead to some performance improvements.

Missing pattern in training data	Imputation method	SSHIBA	AUCs			
			yeast	*scene*	*AVIRIS*	*birds*
No missing values	–	Predictive	0.66	0.92	0.88	0.83
		SS	**0.68**	0.92	0.88	0.83
50% missing values	Semi-Supervised	SS	**0.64**	**0.89**	**0.87**	**0.79**
	Mean		0.61	0.87	0.78	0.77
	Median		0.55	0.70	0.78	0.75
	Most frequent value		0.48	0.52	0.77	0.74

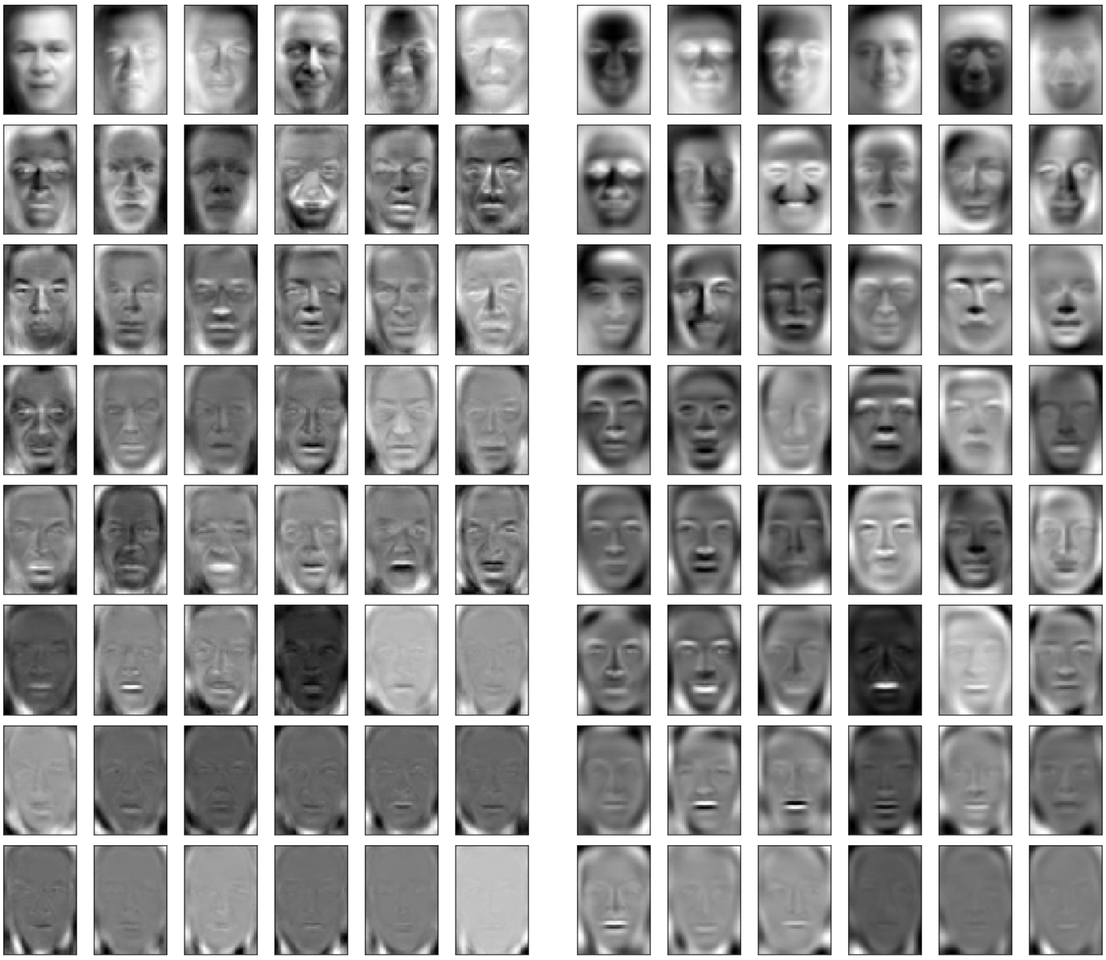
Table 4.10: Results on *yeast*, *scene*, *AVIRIS* and *birds* databases of the predictive and semi-supervised SSHIBA in comparison to different imputation techniques. Results include the AUC values with the complete dataset and when there is a 50% of missing input values.

All these results are included in Table 4.10, where we can see that by using the semi-supervised formulation without missing values, we maintain and, in some cases, improve the results obtained with the predictive model in Table 4.8. Besides, when we analyse the performance with missing values in the training data, we can see that using SSHIBA without any previous pre-processing imputation technique considerably improves the results obtained with the other analysed techniques. We can then conclude that the method in its semi-supervised version can learn hidden correlations in the input information and, consequently, improve the variable modelling and, subsequently, the final model performance.

4.2.1.5 Analysis of feature selection capabilities and model interpretability

Here we will utilise the *LFW* (categorical) and *LFWA* (multi-label) databases to train our model and study the learnt latent factors as well as the feature selection using SSHIBA. We have selected image based datasets for this analysis, since our objective is to visually represent the information learnt by the model and find structural patterns in the features.

The first thing we will represent is the input view's projection matrices $\mathbf{W}^{(1)}$ learnt by the model, commonly known for classic MVA techniques in these types of datasets as eigenfaces. Each face in Figure 4.7 depicts the k-th column of the projection matrix ($\mathbf{w}^{(1)}_{:,\mathrm{k}} \in \mathbb{R}^{\mathrm{D}_1 \times 1}$) rearranged as the original images. The learnt latent faces have been ordered using the corresponding value of $\boldsymbol{\alpha}^{(1)}$, which determines the relevance of each latent feature. This way, the first faces will be more relevant for the model while the last will be less relevant. When generating new data or reconstructing data, these faces will be combined using different weights. Furthermore, we can see that in both databases we reach a point where the images become more distorted and less informative, specially from the sixth row onward, which correspond to a value of $\boldsymbol{\alpha}^{(1)} \approx 0.3$. Therefore, we could set a threshold around that point to prune the less relevant latent factors.



(a) LFW database.

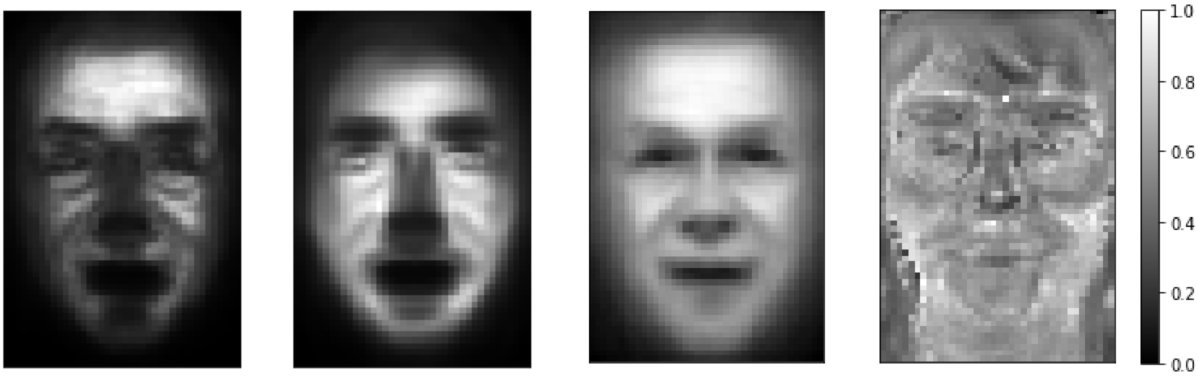
(b) LFWA database.

Figure 4.7: $\mathbf{W}^{(1)}$ matrix learnt by the sparse version of SSHIBA using two different databases. Each latent face is a column of matrix $\mathbf{W}^{(1)}$. The images include the latent faces learnt by the model and are ordered using the latent relevance variable $\boldsymbol{\alpha}^{(1)}$.

Note that, when analysing the results obtained in both databases, we can see that the model adapts itself to the learning task by modifying the relevance of different facial features. We can see this in Figure 4.7(a), where the model centres its attention on the identification of the different individuals (the first latent face is specialised on the label *George W. Bush* and the second one on *Hugo Chávez*). Equivalently, the latent factors learnt in Figure 4.7(b) are dedicated to finding and describing attributes like the forehead, the tooth or the eyes.

Moving on to the feature selection, we can represent the learnt relevance for each feature in an equivalent fashion to the latent factors, rearranging the feature relevance, $\boldsymbol{\gamma}^{(1)}$, as the original pixels. This way, we obtain the feature importance learnt by the model depicted in Figure 4.8. These results show not only that the model can reasonably find the relevant areas of the image for the classification problem, but also that it adapts to the classification task. In particular, we can see that 4.8(a) focuses on the under-eye bags as well as wrinkles and the forehead to discriminate between the 7 subjects. On the other hand, Figure 4.8(b) shows how when the objective is to find different attributes such as whether they wear glasses, the model focuses on different areas that are more relevant for the multi-label classification problem.

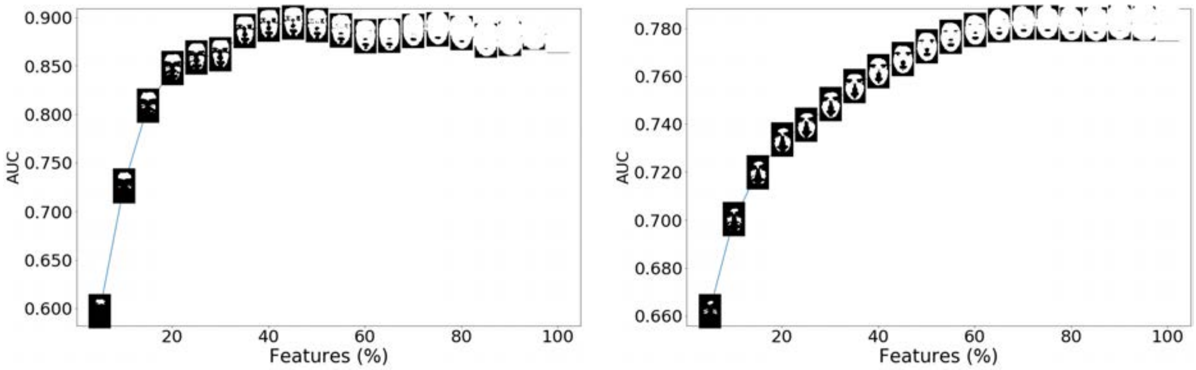
In this case, we have also included the selection obtained using a RBF kernel on the input images on two databases. Figure 4.8(c) corresponds to the classification problem LFW in which, in this case, the relevance mainly differs in its intensity from the results obtained on Figure 4.8(a). While SSHIBA uses an informative prior to impose sparsity and force zeroes on the input features, KSSHIBA uses an ARD over the kernel that learns each feature relevance, but does not force sparsity. We also obtained the relevance learnt by the model on the *AR10P* database, where we can see that the model is capable of learning different face features such as the chin, nose, hair and mouth. More results on the kernel version for FS are available at (Sevilla-Salcedo et al., 2020b).



(a) LFW (SSHIBA) (b) LFWA (SSHIBA) (c) LFW (KSSHIBA) (d) AR10P (KSSHIBA)

Figure 4.8: Gamma masks learnt by the sparse version of SSHIBA and KSSHIBA using three different databases. The masks represent the importance of each pixel: lighter colours imply the pixel is more relevant while darker ones represent the pixel is less relevant.

We finally included an analysis on the effect of the selection on the classification performance. We sorted the pixels by relevance, $\boldsymbol{\gamma}^{(1)}$, and trained the model using different percentages of the input pixels. The results are available in Figure 4.9, where we can see that using only 50% of the original features achieves a good classification performance. Concretely, the results on the *LFW* prove that the model is capable of outperforming the results with all the features using only 40% of them.



(a) LFW database.

(b) LFWA database.

Figure 4.9: AUC results on the *LFW* and *LFWA* databases using the sparse version of the method. These images show the AUC results using different percentages of the most relevant values in the learnt mask. Each face shows the mask with different numbers of features.

4.2.1.6 Evaluation of the kernel selection of relevance vectors

In this section we aim to explore the functionality of KSSHIBA to automatically select the relevant subset of training points, RVs, and reduce the dimensions of the data kernel with a more compact solution. To do so, we use the databases analysed in Table 4.9 only considering KPCA+LR and KCCA+LR as baselines. However, we use a Nyström (Williams and Seeger, 2001) subsampling technique to select a subset of RVs in KPCA and KCCA approaches. While KSSHIBA automatically selects the number of significant RVs, KPCA and KCCA were combined with a 10-fold CV to determine the optimum number of RVs.

The results obtained in this analyses are included in Table 4.11, where we can see that KSSHIBA with the RV selection mostly maintains the performance on Table 4.9 and, in the case of *oes97* and *edm*, even increases it. Furthermore, the model complexity is reduced not only by the compact representation of the kernelised data, but also due to the reduction in the number of latent factors. Comparing the results of KSSHIBA with the other two baselines, we can see that the proposed RV selection provides a more robust selection with, for most cases, a reduced number of RVs. The main reason behind this is that KSSHIBA learns the relevance of each RV while KPCA and KCCA need to do this by randomly selecting them with Nyström subsampling.

4.2.1.7 Multi-view learning

Finally, we will look at the multivariate nature of the presented model. Although the model can work with traditional classification and regression configurations, where there is one set of variables used as input and others as output, it can also combine information from variables of different nature in separate views. Here we will use this functionality to study the results obtained when including different types of kernels in each view. This way, we will have a linear combination of the kernels in the latent representation, working as a MKL approach. However, we could use the multi-view approach in any problem with data of different natures (e.g. categorical and multi-label) to adequately model each data source and later mix their information in the latent representation for the prediction problem.

	Sparse KSSHIBA		KPCA + LR		KCCA + LR	
	R^2 - K$_\mathrm{c}$	%RVs	R^2 - K$_\mathrm{c}$	%RVs	R^2 - K$_\mathrm{c}$	%RVs
at1pd	0.77 ± 0.09	18.4 ± 24.1	0.78 ± 0.09	69.7 ± 32.9	**0.80 ± 0.09**	84.8 ± 27.5
	41 ± 11		87 ± 35		6	
at7pd	0.55 ± 0.15	18.5 ± 26.3	0.56 ± 0.18	79.7 ± 31.7	**0.60 ± 0.12**	73.9 ± 34.1
	70 ± 27		90 ± 37		6	
oes97	**0.58 ± 0.15**	38.6 ± 24.5	0.52 ± 0.24	81.7 ± 27.8	0.42 ± 0.30	23.9 ± 27.8
	61 ± 7		124 ± 34		16	
oes10	**0.77 ± 0.11**	44.4 ± 38.4	0.71 ± 0.12	71.9 ± 11.6	0.66 ± 0.10	57.8 ± 35.2
	74 ± 6		132 ± 53		16	
edm	**0.42 ± 0.21**	53.8 ± 28.5	0.41 ± 0.26	52.5 ± 30.5	0.20 ± 0.14	22.7 ± 13.6
	13 ± 4		29 ± 14		2	
jura	**0.58 ± 0.14**	48.7 ± 38.4	0.57 ± 0.10	60.7 ± 28.9	0.36 ± 0.09	18.9 ± 7.5
	30 ± 4		59 ± 14		3	
wq	**0.12 ± 0.01**	58.1 ± 33.2	0.12 ± 0.02	22.9 ± 15.9	0.10 ± 0.01	5.9 ± 3.1
	21 ± 2		96 ± 49		14	
enb	**0.99 ± 0.01**	19.5 ± 12.8	0.91 ± 0.01	48.9 ± 32.9	0.97 ± 0.01	41.9 ± 12.2
	78 ± 8		28 ± 1		2	

Table 4.11: Results on the multitask databases for the automatic RV selection. The first subcolumn shows on the white subrow the mean and standard deviation of the R^2 score and on the gray subrow the number of effective latent factors (K$_\mathrm{c}$), the second subcolumn includes the percentage of selected RVs (%RVs).

We decided to use a SVM and CCA+SVM as the two reference baselines, for which we will use as input a linear combination of two kernels

$$K(X_{i,:}, X_{j,:}) = \mu K^1(X_{i,S}, X_{j,S}) + (1-\mu)K^2(X_{i,R}, X_{j,R}) \tag{4.27}$$

where K^1 represents the first kernel, K^2 the second kernel and $\mu \in [0,1]$ is a combination coefficient.

As indicated in Table 4.7 we will use two RBF kernels for each database where, for the *Arrhythmia* we used 15 features with demographic and general ECG information for the first kernel and 263 features with the width and amplitude of the channel for the second kernel. For the *Landsat* database 4 features with the spectral information are used in the first kernel and 4 features with the contextual information in the second kernel, as presented in (Amorós-López et al., 2011). Lastly, we decided to use the *Fashion MNIST* database to combine a polynomial and a RBF kernel, both kernels using the same 784 features. For this analysis, in the reference models, besides adjusting the SVM hyperparameters, we also validated the kernel combination coefficient μ[6].

[6] KSSHIBA uses the two kernels as two different inputs and internally determines the relevance of each kernel. Thus, unlike the baselines, it does not need to validate the kernel combination coefficient.

Database	MKL SSHIBA	MKL SSHIBA + RVS	SVM	CCA+SVM
Arrhythmia	0.809 ± 0.057	$\mathbf{0.818 \pm 0.046}$	0.774 ± 0.045	0.737 ± 0.059
Landsat	$\mathbf{0.983 \pm 0.002}$	$\mathbf{0.983 \pm 0.002}$	0.963 ± 0.042	0.980 ± 0.002
Fashion MNIST	0.951 ± 0.011	0.952 ± 0.013	$\mathbf{0.970 \pm 0.010}$	0.844 ± 0.012

Table 4.12: Multiple Kernel Learning analysis. The performance measure used is the multiclass AUC, which we calculated in a 10-fold cross-validation.

We used the three databases with two different kernels composed by specific features to analyse the performance of SSHIBA on a MKL scenario. Table 4.12 shows the results obtained in this setup where we use two different configurations of SSHIBA, with and without RVs Selection (RVS) and we used multiclass AUC to determine the performance of each algorithm. The results prove that SSHIBA obtains a relevant improvement in terms of AUC with respect to the different baselines using the MKL version. This improvement is particularly notable in the *Arrhythmia* database, where the improvement in terms of AUC is 0.044 higher than the result with the SVM. Besides, by including the automatic selection of the relevant RVs improves the robustness of the results. Regarding the *Fashion MNIST* database, we consider the gain obtained by the SVM is not high enough to justify the absence of feature extraction and the lack of interpretability of the model. On the other hand, if we consider the baseline that combines CCA with SVM, we can see that it obtains 0.100 lower AUC performance than SSHIBA along with needing to validate the combination coefficient, which justifies the usage of the proposed model.

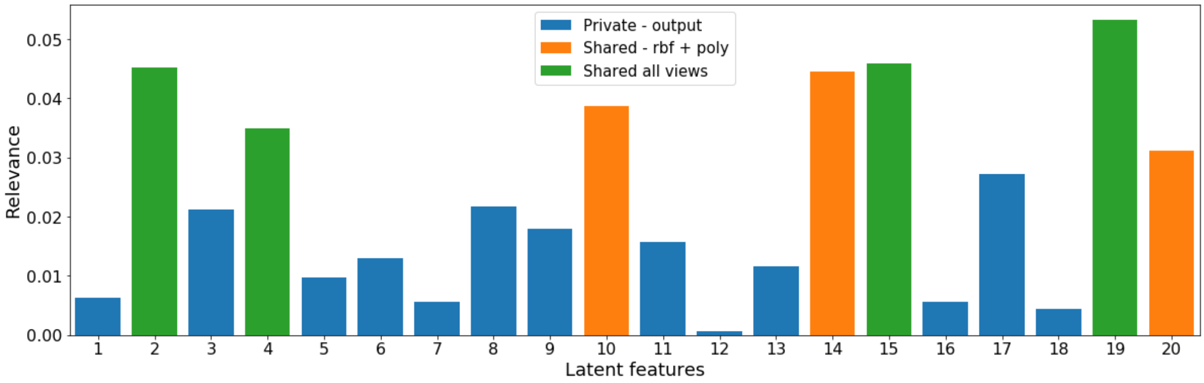


Figure 4.10: Relevance of the learnt latent factors. Measure of relevance on *Fashion MNIST* database combining kernels and labels on different views.

Furthermore, we used the *Fashion MNIST* database using three kernels to analyse the performance and the information learnt by the model. We combined a RBF kernel, a polynomial kernel and a linear kernel with the output labels in a fourth view. The performance in this experiment is the same as in the previous one, proving that the inclusion of a new kernel matrix does not deteriorate the results. We can now analyse the learnt relevance of each of the 100 latent factors. The results are included in Figure 4.10, where the RBF and the polynomial kernel share 3 latent factors and the output view only needs 17 latent factors, from which only 4 are shared with the input kernel views. In fact, we can see that the model can be used with only the 4 latent factors shared by all views without worsening the performance. Note that in the prediction stage we only use the information of the latent features related to the output view, both shared and private, while the rest of the latent factors learn how each view relate with each other.

4.2.2 Neuroimaging databases

In this subsection we will focus on the main objective of the thesis: working with neuroimaging databases. Specifically, here we will compare the Bayesian model, SSHIBA, with the MVA approach, RB-CCA, on the ADNI database. The aim of this first analysis is to evaluate the advantages of each approach in this neuroimaging multiclass classification problem.

In addition, we will analyse the performance of SSHIBA in an even more realistic neuroimaging scenario: with different data sources, missing values, heterogeneous data,... where we will be able to fully explore the versatility of the proposed model.

4.2.2.1 Alzheimer's disease classification

We firstly used the ADNI database, described in Section 3.2.1.1, to compare the results obtained with RB-CCA and SSHIBA in the same scenario. Here, in order to take advantage of the functionalities of the SSHIBA model, we included the following configuration:

- The patient sex and age included as additional views.
- A combination of the MRI voxel information in different views, having 67 views composed by the voxels corresponding to each brain area. These brain regions were obtained from the Harvard-Oxford probabilistic atlas (Makris et al., 2006), resampled to 4 mm spatial resolution and masked into 29.852 voxels to coincide with the ADNI database. This way, we expect to abuse the projection into a common latent space to capture inter- and intra-area correlations. More details on the regions can be found in Appendix B.
- As some areas consist on more features than samples, we treated these as linear kernelised observations to take advantage of the dual space representation.
- Sparsity by means of an ARD prior in both the primal, $\mathbf{X}^{(\mathrm{m})}$, and dual views, $\mathbf{K}^{(\mathrm{m})}$, to learn the relevance of both input features and RVs, respectively. This provides an automatic selection of latent features and, besides, FS in the primal views and RVs in the dual views. To learn the input feature relevance in the dual views, we include an ARD over the kernel.

We provide the results obtain with feature selection (SSHIBA+FS) and without it (SSHIBA) to compare the effect of removing the least relevant features. Moreover, as SSHIBA is a multi-source model, it can learn additional information in the combination of the available data as we will discuss below.

The results obtained are included in Table 4.13, where we combine the above results with the performance obtained by SSHIBA in this scenario. Here we can see that the standard version of SSHIBA (without FS), slightly outperforms RB-CCA without FS in terms of AUC. Furthermore, SSHIBA can largely outperform RB-CCA in classifying the least populated classes (sMCI and pMCI) without using any specific balancing tool as RB-CCA does. However, when we look at

Method	**#feat.**	**Balanced accuracy**	class AUC NC	sMCI	pMCI	AD	**AUC**
RB-CCA	29.852	58, 19 ±6, 42%	0,911	0,718	0,816	0,870	0, 849 ±0, 019
SSHIBA	29.852	57, 02 ±4, 64%	0,890	0,769	**0,833**	0,867	0, 852 ±0, 015
RB-CCA-ST	13.222 ±1.004	62, 91 ±4, 64%	**0,915**	0,766	0,830	**0,882**	**0, 864** **±0, 024**
SSHIBA+FS	22.712 ±571	**63.03** **±2.56%**	0.890	**0.777**	0.832	0.865	0.852 ±0.013

Table 4.13: Results obtained in the ADNI database comparing SSHIBA (with and without FS) with the methods analysed in Chapter 3. The performance results are provided in terms of balanced accuracy and multiclass AUC. These are averaged over the 10 fold CV test partitions, including the associated mean and standard deviation.

RB-CCA-ST, we can see that, by selecting some of the input features, the classification of the most populated classes now largely outperforms SSHIBA (without FS) and consequently provides a more robust solution on this database.

As for SSHIBA+FS, we can see that the model removes the variables that cause overfitting, which especially affect the minority class sMCI. Specifically, by removing these noisy variables, the model improves the classification of sMCI in terms of AUC. This change is most critical in terms of balanced accuracy, where the use of this performance measure leads to a 6% improvement between the two versions of SSHIBA. Also, using accuracy to compare the analysed methods, we observe that SSHIBA+FS slightly outperforms RB-CCA-ST and, at the same time, improves the classification of less populated classes.

However, it should be noted that RB-CCA-ST can automatically select relevant features by a statistical test, while SSHIBA needs either to CV the relevance threshold for selection or to find an ad-hoc threshold which will not be an optimal working point. Moreover, while SSHIBA can perform automatic FS while updating the model parameters with the primal formulation, the dual formulation learns this relevance but does not allow pruning the features while training, as it is limited by the initially defined kernel.

On the other hand, we also want to analyse the interpretability of the selection computed by the sparse version of SSHIBA on the primal views. The results are included in Figure 4.11 where we can see that the model focuses on removing highly redundant voxels that are not providing information for the classification problem. Looking at the learnt relevance, we can also state that the model is providing higher values to the central part of the different brain areas, while the peripheral sections have largely lower relevance. Furthermore, the model is learning to give a higher importance to the ventricles, especially in the frontal part, which are related to the development of AD (Carmichael et al., 2007). Also of high learnt relevance is the grey matter of the occipito-temporal lobe which, according to Carlson et al. (2009), is implicated in problems such as movement blindness and impaired written communication. Another prominent area is the corpus callosum, which connects the two cerebral hemispheres so that they work synchronously.

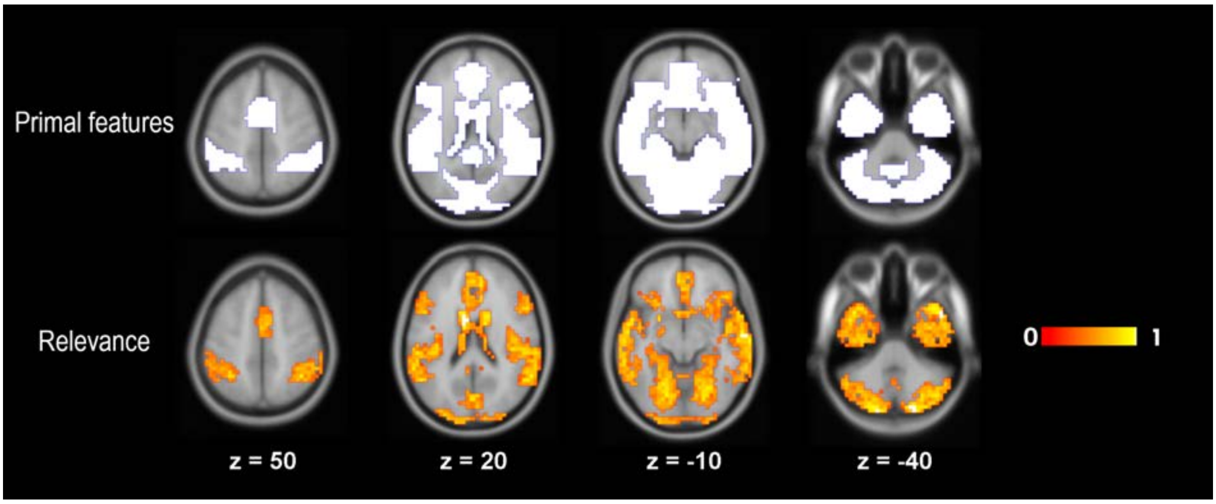


Figure 4.11: Normalised variable relevance on the ADNI database using SSHIBA+FS. These results correspond to the voxels that have been selected in more than 5 of the 10 CV folds. We included four axial slices, at $z = 50mm, 20mm, -10mm, -40mm$ of the MNI space.

Besides, we also want to analyse the learnt latent factors. As SSHIBA automatically calculates a sparse projection matrix and prunes the less relevant latent factors, the final result is composed of the latent factors that have information from a single view or a group of views. We include the relationship between views in Figure 4.12, where we represent the number of relevant latent factors for each view and how many of them are common between views. This matrix can be interpreted as a correlation matrix and shows that the final 138 latent factors, out of the original 1.000, are all shared with the output view, the diagnosis. This is interesting, as the model does not only focus on finding relationships between each pair of views, but finds the relationship between the views and the view we are using to predict. Furthermore, the lack of correlation found between sex and age and the rest of the brain areas could imply that they are not being considerably useful for the classification task, note that the brain image has been corrected for age as specified in chapter 3. Finally, it is also noteworthy that the model is learning that the Brain Stem (BS), in charge of alertness, awareness and cognition, and the Frontal Pole (FP), related to focus in tasks (Mansouri et al., 2015), share their information with almost all other views.

4.2.2.2 Longitudinal classification and regression on TADPOLE

To take full advantage of the potential of SSHIBA, with most of its extensions, we present here the results obtained with the TADPOLE database based on Alzheimer's disease prognosis. We decided to explore the results in this database because of its particularities among neuroimaging databases, as it is a complicated problem with a considerably high number of missing values and different types of data. In this context, the proposed Bayesian model, SSHIBA, adequately addresses the particularities of this scenario. First, we describe the characteristics of this database to understand the objectives of the problem and its different data sources. Then, we define how we adapt SSHIBA to work with each data source. Finally, we describe the baselines used to compare the performance of the model along with an interpretability study.

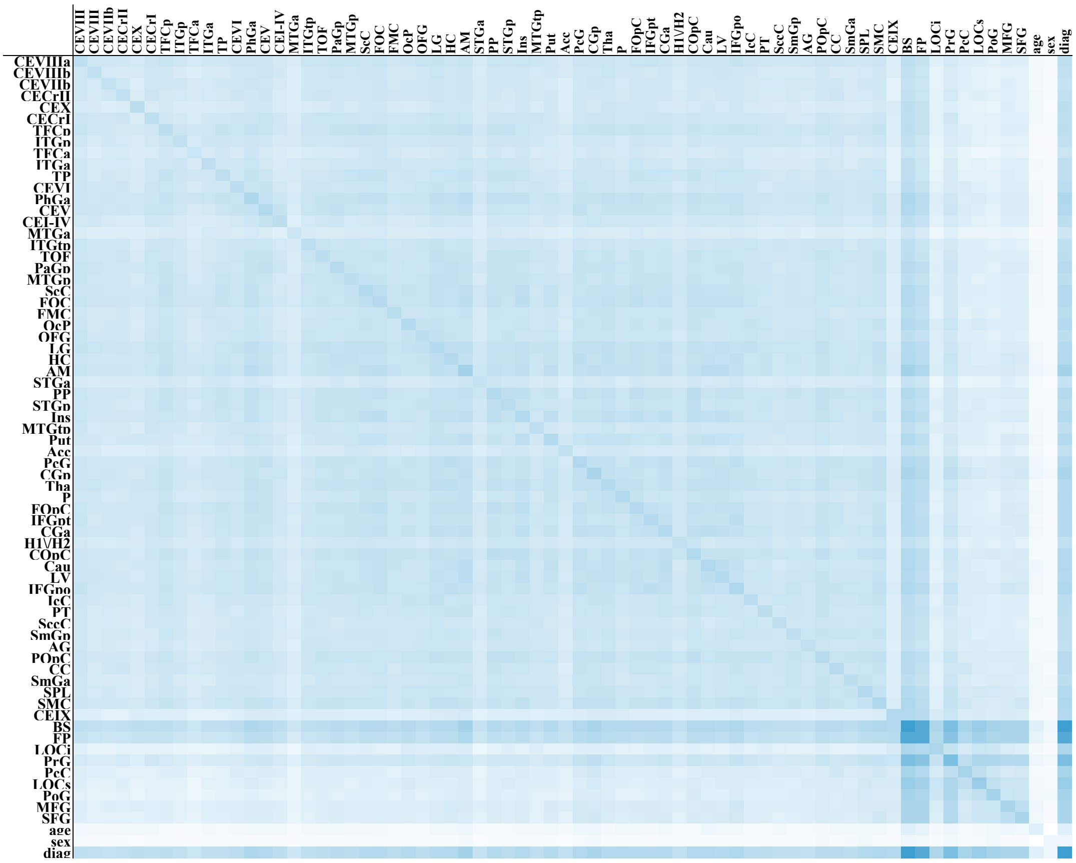


Figure 4.12: Number of latent features associated to each model view. The first 57 views correspond to the primal space views, the next 10 to the kernel views, then there are 2 demographic views and 1 composed by the diagnosis information. Each cell in the matrix represents the number of latent factors common between the two corresponding views. Cells in dark blue imply that the 138 factors are common and cells in white that none are. The diagonal represents the total number of latent factors associated to each view.

Database description

Here we analyse another version of the ADNI database. Specifically, we use the TADPOLE database, available at https://tadpole.grand-challenge.org/Data/. While the ADNI problem focuses on a classification problem, in this case, the TADPOLE database proposes three different predictions: the clinical diagnosis (classification between Cognitively Normal (CN), Mild Cognitive Impairment (MCI) and probable AD), Alzheimer's Disease Assessment Scale Cognitive Subdomain (ADAS-Cog13) score and the ventricle volume from MRI. For this purpose, this database provides a set of ADNI biomarkers to analyse the progression of dementia. The biomarkers focus on two main categories that have been found useful for the diagnosis of AD and MCI. The first is amyloid beta protein, which has been measured using CerebroSpinal Fluid (CSF) and Positron Emission Tomography (PET). The second is nerve cell damage. This has been measured by quantifying the fraction of tau protein in CSF (tau-PET), quantifying brain metabolism with FluoroDeoxyGlucose (FDG) PET and using MRI. To complete the dataset, these biomarkers are combined with some cognitive tests that are valuable for the analysis of Alzheimer's disease, as well as some demographic information and clinical diagnosis. The variable Diagnosis has been changed from the original TADPOLE database so that it now specifies the patient's current diagnosis rather than whether the patient's diagnosis has changed between measures. Some variables in the database have values given in the form of *'<1300'* and *'>120'* that have been set to their specified limit (i.e. 1300 and 120 respectively).

In addition, the data used from this database is composed of longitudinal data, which combines information from all tests performed on each subject over time. As the periodicity of testing may differ between patients, this also implies that there are many missing values. In fact, some variables have up to 89% of missing values, which makes it difficult to use these variables.

Although in the original database there is information up to 120 months after month 0, we have decided to use the data for the first 36 months. Firstly, because the further away from month 0, the more missing values we have, which could lead to inconclusive results. This also provides a more realistic framework in which we have information for up to 3 years. Furthermore, in this way we can train the model using the information for the first 2 years in 6-month intervals (months 0, 6, 12, 18 and 24) and determine a one-year forecast (month 36).

As the database includes some longitudinal information, we have first checked which variables depend on the timestamp and divided them into time dependent and time independent. As a result, we obtain a set of 33 variables described in Table 4.14.

SSHIBA configuration

In light of the nature of the problem we decided to use SSHIBA capabilities to adequately combine the information of the different heterogeneous data into a common latent space. Furthermore, the modular nature of SSHIBA allows us to combine heterogeneous data with multi-view learning, while having the ability to impose sparsity in the input space and automatically impute missing values. It is this last aspect which particularly fits the problem at hand, as there is a considerably high amount of missing values in the data (see Table 4.14).

For this experiment, we are considering three different variables to predict: the diagnosis (NC, MCI, AD) defined as D[t], the ventricle volume of the MRI data defined as V[t] and the ADAS13 score defined as A[t], where t represents the month. These variables are included in three different views of the model for the prediction (output views).

Taking advantage of the formulation, we decided to utilise the multi-view framework to combine the time independent information and the time-dependent variables in the training and test sets (described in Table 4.14). To do so, we decided to use one view to model all the time independent data and combine the time dependent data in different views, as detailed in Figure 4.13. Besides, as the diagnosis prediction corresponds to a multiclass classification problem, we also included the diagnosis for previous months in different input views.

Specifically, the aim of the framework is to be able to make predictions using information from up to 36 months in advance. Thus, we have the measures associated to the timestamps, from 6 to 24 months before the prediction, each modelled in one view, i.e. 5 views for the 5 timestamps. Therefore, to predict an outcome 30 months after month 0 we would use the information related to months 0, 6, 12, 18 and 24. However, to improve the performance of the model, we decided to use the missing values functionality of the model to increase the number of samples per patient

Variable	Description	Time dependent	Group	Missing
Age	Age at baseline	No	TI	0.1%
Education	Years of education	No	TI	0.1%
APOE4	Number of APOE4 alleles (genes)	No	TI	0.2%
Gender	Female or Male	No	TI	0.1%
AngularLeft	FDG PET derived measures	No	TI	5%
AngularRight	FDG PET derived measures	No	TI	5%
CingulumPostBilateral	FDG PET derived measures	No	TI	5%
TemporalLeft	FDG PET derived measures	No	TI	5%
TemporalRight	FDG PET derived measures	No	TI	5%
Diagnosis	NC, MCI or AD	Yes	D[t]	53%
ADAS13	ADAS-Cog13 score	Yes	A[t]	50%
MMSE	Neurophysclogical and behavioural tests	Yes	TD[t]	50%
RAVLT learning	Neurophysclogical and behavioural tests	Yes	TD[t]	50%
RAVLT immediate	Neurophysclogical and behavioural tests	Yes	TD[t]	50%
RAVLT perc forgetting	Neurophysclogical and behavioural tests	Yes	TD[t]	50%
FAQ	Neurophysclogical and behavioural tests	Yes	TD[t]	50%
Ventricle volume	MRI volumetry	Yes	V[t]	55%
Hippocampus	MRI volumetry	Yes	TD[t]	60%
WholeBrain	MRI volumetry	Yes	TD[t]	54%
Entorhinal	MRI volumetry	Yes	TD[t]	61%
Fusiform	MRI volumetry	Yes	TD[t]	61%
MidTemp	MRI volumetry	Yes	TD[t]	61%
ICV	MRI volumetry	Yes	TD[t]	53%
Cerebellum Grey Matter	AVF45 PET data	Yes	TD[t]	89%
Whole Cerebellum	AVF45 PET data	Yes	TD[t]	89%
Eroded Subcortical Wm	AVF45 PET data	Yes	TD[t]	89%
Frontal	AVF45 PET data	Yes	TD[t]	89%
Cingulate	AVF45 PET data	Yes	TD[t]	89%
Parietal	AVF45 PET data	Yes	TD[t]	89%
Temporal	AVF45 PET data	Yes	TD[t]	89%
ABETA	CSF values	Yes	TD[t]	84%
TAU	CSF values	Yes	TD[t]	84%
PTAU	CSF values	Yes	TD[t]	84%

Table 4.14: Description of the different variables in the TADPOLE database. Each variable is assigned to one group to facilitate the understanding of the framework: TI, Time Independent variables; TD[t], Time Dependent variables at month t; D[t], Diagnosis at month t; V[t], Ventricle volume at month t; A[t], ADAS13 score at month t. We also included the percentage of missing values associated to each variable, where the ones that are not time dependent have considerably less missing values.

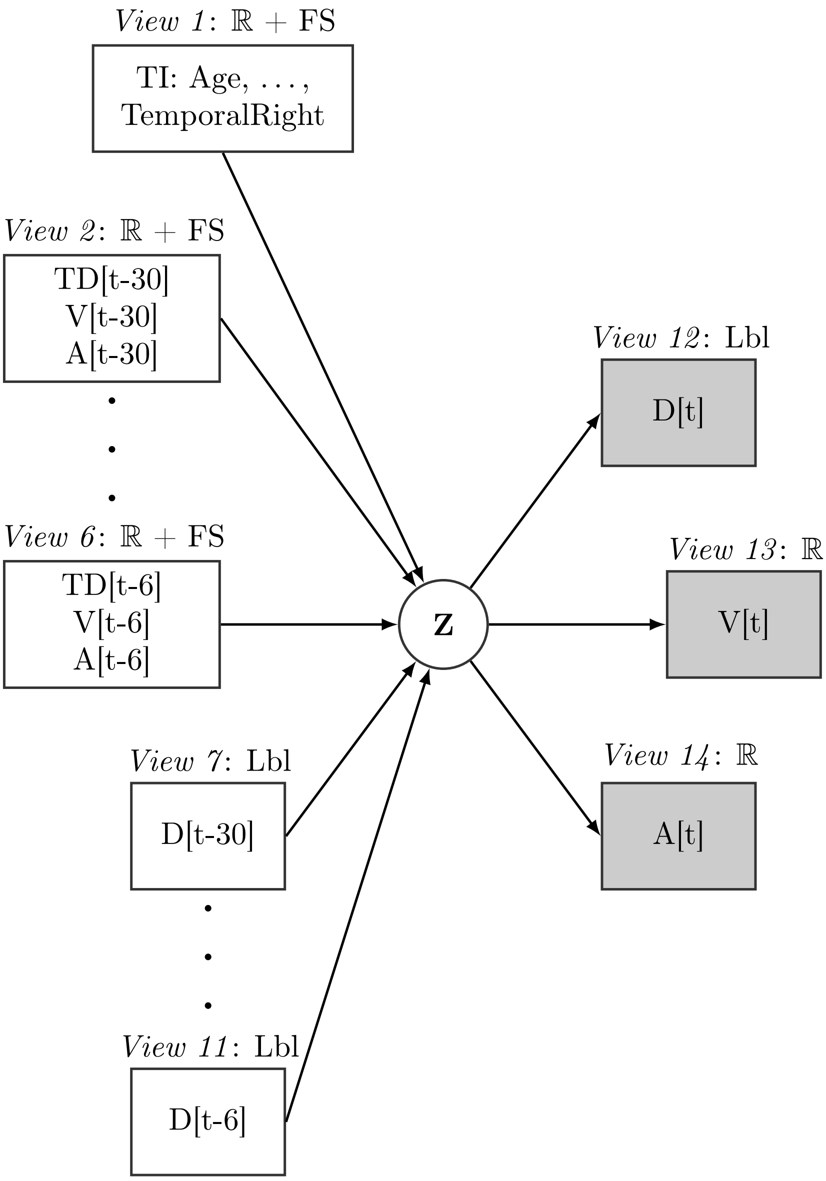


Figure 4.13: Plate diagram for the SSHIBA graphical model. White circles denote unobserved r.v., white squares denote input views and grey squares output views. TI comprises the Time Independent variables and TD[t] the Time Dependent variables not counting the output variables. [t-x] represent x months before month t. ℝ means the view treats the information in it as real, ℝ + FS that it also includes feature selection and Lbl that the data is treated as multi-label.

we have. For this purpose, we consider that there are missing values in the measurements prior to month 0 so that the model always uses 5 timestamps to make the prediction, in the training step. Therefore, we not only use the information related to months 0, 6, 12, 12, 18 and 24 to predict 30 as our training data, but also months -6, 0, 6, 6, 12 and 18 to predict 24 and so on, where the data corresponding to each negative month is treated as missing values. This scheme is summarised in the table 4.15, noting that the test set is always set to predict month 36 and always uses months 6 to 24 as input. This change in the way the information is handled allows the model to grow from 1.787 to 8.685 training samples, while keeping the original 1.787 test samples. See table 4.15 for more details.

	View	Samples for patient p					
		train 1	train 2	train 3	train 4	train 5	test 1
Input	1	TI	TI	TI	TI	TI	TI
	2	–	–	–	–	TD[0]	TD[6]
	3	–	–	–	TD[0]	TD[6]	TD[12]
	4	–	–	TD[0]	TD[6]	TD[12]	TD[18]
	5	–	TD[0]	TD[6]	TD[12]	TD[18]	TD[24]
	6	TD[0]	TD[6]	TD[12]	TD[18]	TD[24]	–
	7	–	–	–	–	D[0]	D[6]
	8	–	–	–	D[0]	D[6]	D[12]
	9	–	–	D[0]	D[6]	D[12]	D[18]
	10	–	D[0]	D[6]	D[12]	D[18]	D[24]
	11	D[0]	D[6]	D[12]	D[18]	D[24]	–
Output	12	D[6]	D[12]	D[18]	D[24]	–	D[36]
	13	V[6]	V[12]	V[18]	V[24]	–	V[36]
	14	A[6]	A[12]	A[18]	A[24]	–	A[36]

Table 4.15: Description of the training/test data configuration. D represents the diagnosis, V the ventricle volume, A the ADAS13 score, TI represents the Time Independent data and TD the Time Dependent data for month t including V and D. TDs are actually divided in another view only with the diagnosis variable (D) and treated as multi-label.

This way, we have that the first view includes the variables that are time independent; views 2-6 the time dependent variables corresponding to the 5 timestamps previous to the prediction's timestamp; views 7-11 are equivalent to the previous five but only have the diagnosis information; and views 12-14 have the three output variables.

We used the heterogeneity of SSHIBA to model each view differently depending on the data nature. In particular, the set of views $\mathcal{M}_i = \{1, 2, \ldots, 6\}$ work with real data and also include sparsity in the features space in order to automatically learn the relevance associated to each feature in this view. Besides, this extension also provides a measure of the effect of each feature in each evaluated month. The set of views $\mathcal{M}_m = \{7, \ldots, 12\}$ are modelled as multi-label data and the set of views $\mathcal{M}_o = \{13, 14\}$ work with real data without sparsity in the feature space.

Performance analysis

This section presents the results obtained using SSHIBA to predict the clinical diagnosis (D), ventricle volume of the MRI (V) and the ADAS13 score (A) at month 36 in comparison to some state-of-the-art baselines. To configure these baselines, due to the nature of the problem, we combined the regressor or classifier with different imputation strategies to work around the missing values. In particular, as regressor we decided to use Ridge Regression (RR) and as classifier we used Logistic Regression (LogR). To impute the missing data we used 7 different strategies: substituting by zero, the mean, the median and the most frequent value, temporal imputation, the KNN Imputer (which substitutes the value by the mean of the K nearest neighbours) and

Iterative Imputer (which models the missing features as a function of the rest in a round-robin fashion). However, we finally included the results obtained with the first five, as we observed that KNN and Iterative Imputer always imputed the mean value. We also incorporate a temporal imputation of missing values by replacing them with the mean of the patient's previous existing values. If there are no previous existing value, we substitute it by the variable mean. The main limitation for the comparison with these baselines is the fact that they can neither use multi-output estimation nor work with heterogeneous data modelled in different views, so we had to include all input views as input variables for each model and train one model for each prediction problem. The regularisation parameter for RR was cross-validated using a 10-fold cross-validation for 11 values in log space between -20 and 2. We used the same pruning and convergence criteria in SSHIBA as in the previous section.

Regarding the performance evaluation, for the regression prediction of variables A[t] and V[t] (views 13 and 14), we use the coefficient of determination (R^2). For the classification problem of D[t] (view 12), we decided to use the multiclass AUC as in previous experiments.

We include the results on two different SSHIBA frameworks: (1) SSHIBA 1-output, where we train one model to predict each output variable resulting in 3 SSHIBA models; (2) SSHIBA 3-output, where we train a single model that simultaneously predict the three output variables taking advantage of the inter-output views relationships to construct the model. Table 4.16 collects these results using the respective scoring for each output variable. We can see that, the best baseline, RR + temporal imputation, obtained a R^2 of 0,701 in the prediction of ADAS13 score and 0,924 for ventricle volume while SSHIBA gets to improve to 0,839 and 0,971, respectively. Considering the diagnosis prediction AUC score, the results of the best baseline and SSHIBA in both frameworks are equivalent. These results prove that SSHIBA is capable of fully adapting to this complex scenario where all temporal variables having more than 50% missing values and provides competitive performance in terms of R^2 and AUC while providing interpretable results.

Imputation strategy	Regressor / Classifier	ADAS13	Ventricle	Diagnosis
		R^2	R^2	AUC
zero	RR / LogR	0.097	0.653	0.591
mean		0.609	0.653	0.738
median		0.569	0.621	0.738
most frequent		0.360	0.189	0.738
temporal		0.701	0.924	**0.904**
SSHIBA single output		0.838	0.969	0.902
SSHIBA multiple output		**0.839**	**0.971**	0.903

Table 4.16: Results obtained in the prediction of three different variables. Each model was trained independently for each task with the exception of SSHIBA multiple output, which simultaneously predicts the three variables. We used two different scores for this experiment, namely, R^2 for ADAS13 and Ventricle and multiclass AUC for Diagnosis.

Furthermore, SSHIBA's 3-output configuration proves to be slightly better than predicting a single variable at a time. This demonstrates that the model is able to adequately learn the information inherent in each data view and exploit their relationships, combining heterogeneous data and simultaneously predicting different data types, labels and real values. Both SSHIBA configurations impute missing values in the months prior to prediction, having to simultaneously predict relevant missing variables, which could be of clinical relevance. The following section will discuss the relationships of the variables and the outcome using the learnt latent factors, the learnt relevance associated with each variable and its relationship to missing values.

4.2.2.3 Interpretability analysis

Factor analysis algorithms provide interpretability to the results that may help identifying relations between variables and other information related to the data. In this section we want to analyse some of these characteristics learnt by the SSHIBA 3-output framework.

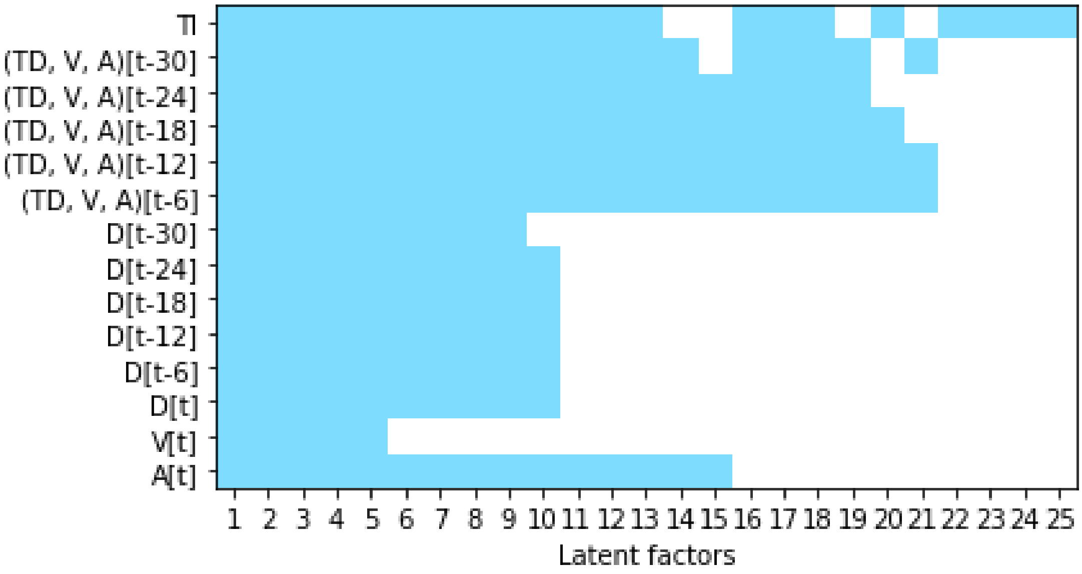
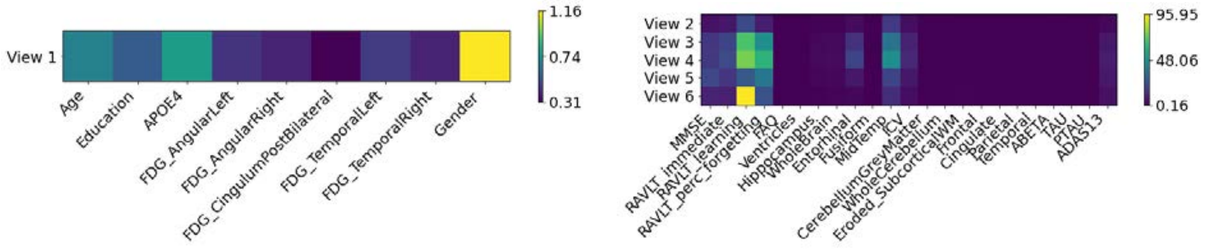


Figure 4.14: Learnt latent factors with multiple output prediction. Blue cells Correspond to latent variables relevant to the associated view and white cells to irrelevant ones.

The ARD prior on the $\mathbf{W}^{(\mathrm{m})}$ matrix induces zeros in the latent factors, leading to the elimination of those that are irrelevant to certain views. In particular, this pruning makes some latent features common only to some views, hence, learning relations between views. Figure 4.14 shows the latent factors learnt by the model, where the 14 views have been concatenated by rows and the factors have been reordered by columns to show how these factors relate to each view. This figure shows that the model learns 5 factors common to all views (factors 1-5), which combine information from all available data; the next 5 factors (6-10) combine all views but do not use information from the output ventricle and, in one case, the diagnosis at the first month (10). In addition, there are 5 other factors (11-15) that only use the output information from the ADAS13 score and do not use the diagnostic information, two of which do not use the TI variables (factors 14 and 15). Finally, there are 10 latent factors that are not related to either the outcome or the diagnosis in the previous months. These final latent factors focus on the combination of information from the TI and TD variables.

In order to analyse the relations between latent factors, we can study the implications of having them associated to just one view. Comparing these results to classic feature extraction algorithms, the last 4 latent factors, which only have information of the TI view, can be interpreted as a PCA where the information of view 1 is used to find a low dimensional projection space. Equivalently, the latent factors associated to the other views can be seen as a CCA, where factors 16 to 21 only combine input information and the rest use some or all the views to construct a latent space using both input and output variables. Even though some latent factors prove that every input data is relevant for the prediction problem, we can see that the output ADAS13 score (view 14) is mostly related to views 1-6. Equivalently, the prediction of ventricle volume (view 13) does not need any shared latent factor besides the ones that are common to every view. As expected, the output diagnosis view (view 12) is always related to the views composed by the diagnosis at previous timestamps.



(a) Relevance of TI variables.

(b) Month evolution of the relevance of TD variables.

Figure 4.15: Analysis of the relevance learnt by the model for each feature. Figure 4.15(a) represents all the time-independent variables of the model and figure 4.15(b) the different relevance of the time-dependent data for each view.

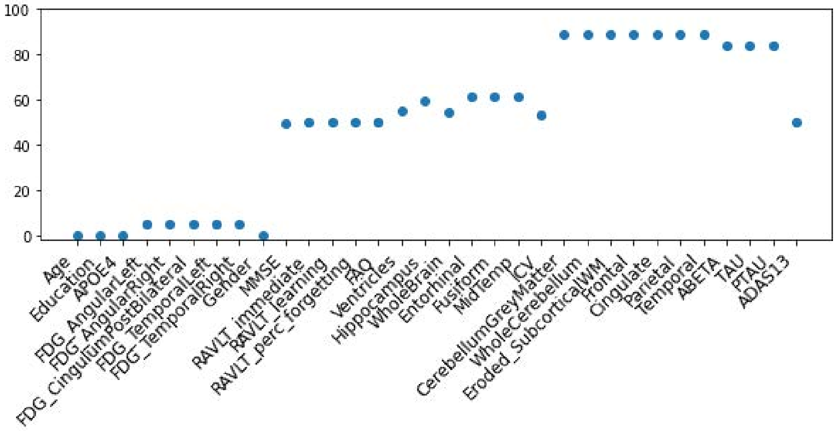
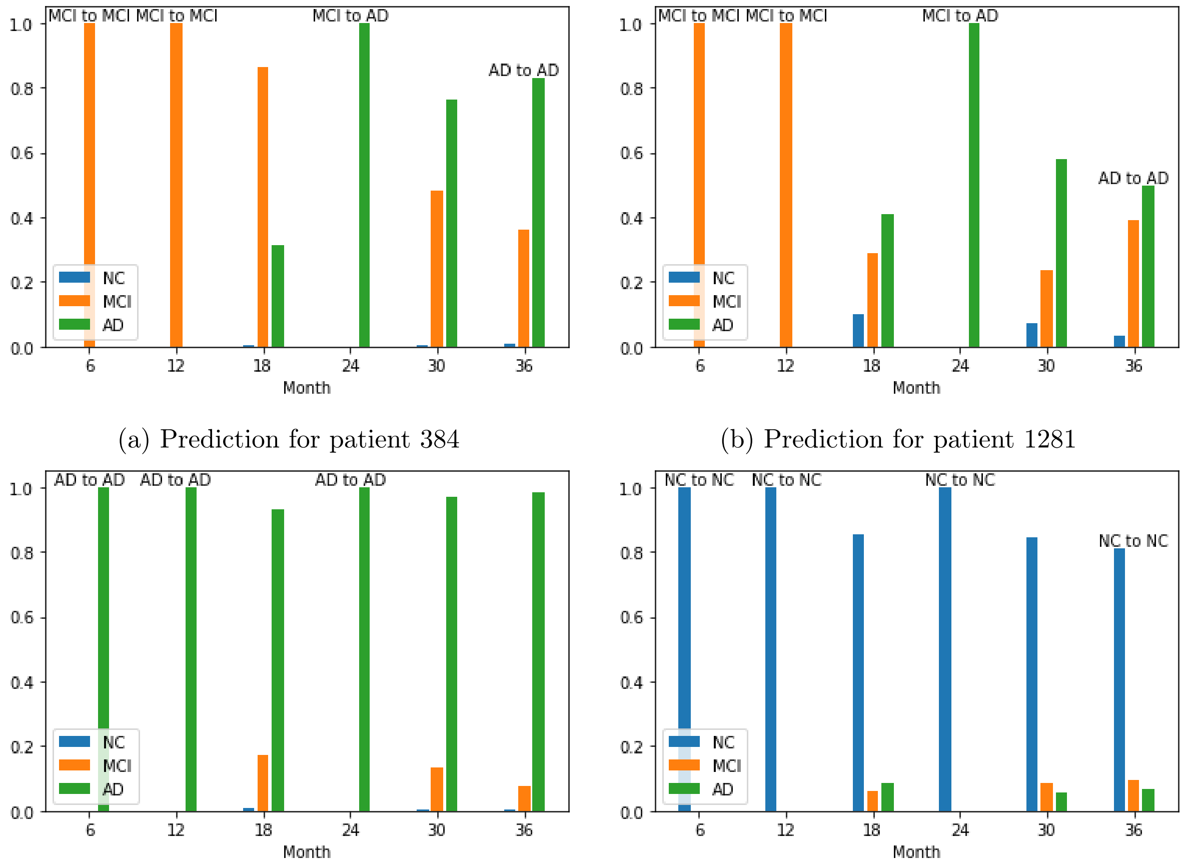


Figure 4.16: Percentage of missing values for each input variable. The values of the TD variables have been calculated as the average of the number of missing values for the analysed timestamps.

Another interesting functionality of the framework is its ability to impose sparsity on the feature space to provide a significance measure for each input variable of a given view. The relevance learnt by the model is presented in Figure 4.15, where we show the relevance associated to TI variables (Figure 4.15(a)) and TD variables for different time stamps before prediction (Figure 4.15(b)). These results show the importance of the neuropsychological and behavioural tests as well as the MRI data with respect to the rest of the variables. Furthermore, the difference in scales between the two figures shows that the level of importance of the TD variables makes the relevance of the TI variables insignificant.

However, these results are highly correlated with the number of missing values in each input variable. Figure 4.16 shows the average percentage of missing values at each timestamp for each variable, where we concatenate the variables TI and TD. This image illustrates the high number of missing values, especially in the values of AVF45 and CSF, which could lead to a lower relevance of the features learnt by the model.



(a) Prediction for patient 384

(b) Prediction for patient 1281

(c) Prediction for patient 1

(d) Prediction for patient 649

Figure 4.17: Exemplary diagnosis prediction for some relevant patients for month 36 with associated learnt prediction of missing values in previous months. Each bar represents the probability of each class in the corresponding month. If there is only one bar with probability 1 before month 36, the change in patient diagnosis is available and included in the text above the bar. Otherwise, the bars correspond to the predicted estimate determined by the model. Regardless of whether labels are available, we always include the prediction of the model in the output of the test set, month 36.

On the other hand, the Bayesian nature of the formulation provides more information in the prediction of the output diagnostic labels (NC, MCI and AD). Figure 4.17 shows some exemplary prediction results obtained with the 3-output SSHIBA framework. In particular, we decided to use the SSHIBA multi-label scheme, making the model able to find two labels for the same patient in an attempt to capture information related to the change in patient diagnosis. Although these results were obtained for the prediction of the output variables at month 36, the model is able to learn any other missing values in the data as well. This implies that, for example, the model can also learn the diagnosis of the patient in a certain unavailable month. In the results shown in Figure 4.17, we included the diagnosis label above the bar when available and their associated predictive estimation learnt by the model.

Looking at figures 4.17(a) and 4.17(b) we see that there was a change in diagnosis between month 0 and 36 which, if we observe the predictive distribution at months 18 and 30, this change is captured by the model. We can see how the probability associated with each label changes with the shift in clinical diagnosis. This type of soft diagnoses can be particularly interesting in patients who are developing a change in the stage of their disease. Another representative behaviour in diagnosis prediction is seen in figures 4.17(c) and 4.17(d), where the patient is clearly associated with a label and the label distribution learns the highest probability associated with that class.

Chapter 5

Conclusions

In this thesis we have analysed different alternatives for characterising mental diseases. In particular, we have focused on different options to work with high-dimensional neuroimaging data using various feature reduction techniques. In this context, we decided to combine FS and FE/FA techniques to build a more robust and interpretable model. Furthermore, we also explored solutions to work with other features often present in neuroimaging data, such as the imbalance of certain classes, the availability of heterogeneous data or the existence of missing values. To make efficient use of the available information, we proposed two models based on FE techniques.

The first model, RB-CCA, is able to extract summary components for neuroimaging data based on the classical MVA formulation. This model starts with a bagging procedure, which uses the learnt projection matrices to determine their feature-wise sign consistency and magnitude. These values dictate the importance of each input feature and are subsequently used to select the most relevant features, as well as to guide the next FE step through a regularised CCA. Although the number of selected features could be determined using CV, we decided to define an approach based on a hypothesis test, which considerably reduces the computational time and provides a more stable working point, which in turn reduces the variance of the number of selected features.

Moreover, we provide the model with the ability to calculate class-wise FS using the bagging procedure which, together with the hypothesis test, is able to automatically select the most relevant features for each class. This considerably improves the selection of the most relevant features, as well as providing more interpretable results. Additionally, we define the model to run in dual space, which copes with the intractable computational time of neuroimaging problems, and we included a balanced formulation for heavily imbalanced scenarios.

In terms of performance, it is important to note that the results show the considerable improvement of RB-CCA-ST compared to the different baselines in terms of both accuracy and AUC, especially in the ADNI database. Specifically, this is obtained while simultaneously determining two or three latent factors and class-wise feature selection. Notably, we have seen that the areas selected by the algorithm match the literature, being relevant areas of AD-related brain atrophy such as the thalamus, hippocampus and parietal lobe.

Next, the second proposed model, SSHIBA, brings a Bayesian formulation of FA capable of combining different functionalities in a single framework. In particular, it allows to impute missing values, perform FS, deal with heterogeneous data, work in dual space using kernels in its formulation and select kernel RVs. Furthermore, the model combines all these extensions in different views to obtain a common low-dimensional latent space that extracts both inter- and intra-view information.

First, the results testing the different functionalities show that the model performance is similar to or improves on state-of-the-art algorithms, while finding a latent space of reduced dimensionality, with fewer latent features than classical MVA methods. Besides, we show that the different extensions can be combined without deteriorating the results and even improving them, obtaining more compact models with more interpretable results. Furthermore, the dual formulation proves to model a good approximation of the kernel with competent results, allowing to include non-linearities in the model. By combining these extensions in different views, the model can also learn the relevance associated with each of them, which correspond to the MKL combination coefficients for the kernel version of SSHIBA.

Subsequently, in the experiments focusing on neuroimaging problems, SSHIBA proved to be an efficient model. Performance results obtained in comparison with RB-CCA on the ADNI database showed that both models perform similarly in terms of balanced accuracy and multiclass AUC. Surprisingly, in this experiment, the classification AUC of less populated classes is better using SSHIBA than RB-CCA, despite not using any specific balancing tool. The longitudinal neuroimaging database TADPOLE is a more realistic database, where the number of missing values is considerably high for certain variables. In this scenario, SSHIBA demonstrated that its ability to combine different extensions allows it to adequately model real-life problems. In particular, the results far outperform all baselines analysed when three output variables are predicted simultaneously. Besides, the model is able to provide several representations that further improve the interpretability of the results.

5.1 Future work

The original idea of developing SSHIBA came from a concern to develop RB-CCA towards a Bayesian framework. The initial perspective was to adapt it to preserve the FS capabilities while using probabilistic formulations to improve its performance. Subsequently, the different extensions and adaptations presented here emerged as needs to adapt the formulation to the characterisation of mental diseases. For this reason, future adaptations and proposals focus on SSHIBA and not on RB-CCA.

Concretely, one of the main extensions of the model that we will include is the modelling of time series. Although SSHIBA proved to perform well working with longitudinal data, we want to adequately model this type of data. The current formulation is based on combining different views in a latent space which, in turn, we use to include data from different timestamps in different views. The nature of the model allows us to capture the correlations of the views and to learn the time dependence in the linear relationships between them. However, we want to further exploit this framework to explicitly specify the relationships between some views. In this way, we expect to improve performance with time-dependent data and maintain the modular essence of the model. This can be done by having a temporal model for each timestamp and relating the latent variables in a manner equivalent to Hidden Markov Models. In this way, the latent factors can simultaneously model inter- and intra-view information and their time evolution.

As specified in Chapter 4, the presented kernel adaptation of SSHIBA does not intuitively fit into a generative model. Although the kernel reconstruction error shown ensures that the model is correctly learning the internal relationships and structure of the kernel, from a generative point of view, it is not the most sensible approach to the problem. Moreover, the current model considers the samples to be uncorrelated, which in most cases is an erroneous assumption, as the kernel relevance vectors are usually a subset or all of the training samples. For this reason, we want to include an alternative solution to this formulation that guarantees to have a generative model that works in dual space but has as observed variables the input data instead of the kernel. Although the adaptation is not complicated, our aim is to preserve the available extensions of the model and to be able to combine them freely for most scenarios. In particular, this adaptation will use a prior ARD on the primal projection matrix to impose sparsity on the feature space for immediate feature selection, whereas we have seen that the current ARD on the kernel provides a measure of feature importance, but is not reliable enough to have a consistent FS.

Finally, we can integrate neural networks to model different data types frequently used for neuroimaging problems and combine them in the common latent space learnt by SSHIBA. Besides, using neural networks allow to find non-linear representations of the original data using non-linear layers in their architecture which, in turn, improves the expressiveness of the model. In fact, using Variational AutoEncoders (VAE), we can intuitively integrate the neural network into SSHIBA probabilistic formulation. Particularly, we can obtain an image feature map corresponding to an input fMRI to exploit its spatial structure using one convolutional neural network as an encoder and another as a decoder to recover the image space. Another possible application is to increase the SSHIBA heterogeneity modelling ordinal data with a neural network; this can be useful, for example, to measure the stage on the disease or to represent psychological tests.

Bibliography

Abdulkadir, A., Peter, J., Ronneberger, O., Brox, T., Klöppel, S., 2014. Voxel-based multi-class classification of ad, mci, and elderly controls. Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2014 CADDementia Challenge 2014, 1–8.

Ahmad, F., Dar, W.M., 2018. Classification of alzheimer's disease stages: An approach using pca-based algorithm. American Journal of Alzheimer's Disease & Other Dementias® 33, 433–439.

Amorós-López, J., Gómez-Chova, L., Guanter, L., Alonso, L., Moreno, J., Camps-Valls, G., 2011. Multitemporal fusion of landsat and meris images, in: 2011 6th International Workshop on the Analysis of Multi-temporal Remote Sensing Images (Multi-Temp), IEEE. pp. 81–84.

Bach, F.R., Jordan, M.I., 2005. A probabilistic interpretation of canonical correlation analysis. Technical Report. University of California, Berkeley.

Bahg, G., Evans, D.G., Galdo, M., Turner, B.M., 2020. Gaussian process linking functions for mind, brain, and behavior. Proceedings of the National Academy of Sciences 117, 29398–29406.

Baumgardner, M., Biehl, L., Landgrebe, D., 1992. 220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3. purdue university research repository. 2015.

Bellec, P., Chu, C., Chouinard-Decorte, F., Benhajali, Y., Margulies, D.S., Craddock, R.C., 2017. The neuro bureau adhd-200 preprocessed repository. Neuroimage 144, 275–286.

Bereman, M.S., Beri, J., Enders, J.R., Nash, T., 2018. Machine learning reveals protein signatures in csf and plasma fluids of clinical value for als. Scientific reports 8, 1–14.

Bi, J., Bennett, K., Embrechts, M., Breneman, C., Song, M., 2003. Dimensionality reduction via sparse support vector machines. Journal of Machine Learning Research 3, 1229–1243.

Bishop, C.M., 1999. Bayesian pca. Advances in neural information processing systems , 382–388.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg.

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

Boutell, M.R., Luo, J., Shen, X., Brown, C.M., 2004. Learning multi-label scene classification. Pattern recognition 37, 1757–1771.

Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

Briggs, F., Huang, Y., Raich, R., Eftaxias, K., Lei, Z., Cukierski, W., Hadley, S.F., Hadley, A., Betts, M., Fern, X.Z., et al., 2013. The 9th annual mlsp competition: new methods for acoustic classification of multiple simultaneous bird species in a noisy environment, in: 2013 IEEE international workshop on machine learning for signal processing (MLSP), IEEE. pp. 1–8.

Bron, E.E., Smits, M., Van Der Flier, W.M., Vrenken, H., Barkhof, F., Scheltens, P., Papma, J.M., Steketee, R.M., Orellana, C.M., Meijboom, R., et al., 2015. Standardized evaluation of algorithms for computer-aided diagnosis of dementia based on structural mri: the caddementia challenge. NeuroImage 111, 562–579.

Bzdok, D., Nichols, T.E., Smith, S.M., 2019. Towards algorithmic analytics for large-scale datasets. Nature machine intelligence 1, 296–306.

Carlson, N.R., Heth, D., Miller, H., Donahoe, J., Martin, G.N., 2009. Psychology: the science of behavior. Pearson.

Carmichael, O.T., Kuller, L.H., Lopez, O.L., Thompson, P.M., Dutton, R.A., Lu, A., Lee, S.E., Lee, J.Y., Aizenstein, H.J., Meltzer, C.C., et al., 2007. Ventricular volume and dementia progression in the cardiovascular health study. Neurobiology of aging 28, 389–397.

Castro, E., Martínez-Ramón, M., Pearlson, G., Sui, J., Calhoun, V.D., 2011. Characterization of groups using composite kernels and multi-source fmri analysis data: application to schizophrenia. Neuroimage 58, 526–536.

Chang, K.C., Hsieh, P.H., Wu, M.Y., Wang, Y.C., Chen, J.Y., Tsai, F.J., Shih, E.S., Hwang, M.J., Huang, T.C., 2021. Usefulness of machine learning-based detection and classification of cardiac arrhythmias with 12-lead electrocardiograms. Canadian Journal of Cardiology 37, 94–104.

Chen, J., Du, L., He, H., Guo, Y., 2019. Convolutional factor analysis model with application to radar automatic target recognition. Pattern Recognition 87, 140–156.

Cheng, B., Liu, M., Shen, D., Li, Z., Zhang, D., Initiative, A.D.N., et al., 2017. Multi-domain transfer learning for early diagnosis of alzheimer's disease. Neuroinformatics 15, 115–132.

Connor, P., Hollensen, P., Krigolson, O., Trappenberg, T., 2015. A biological mechanism for bayesian feature selection: Weight decay and raising the lasso. Neural Networks 67, 121–130.

Damianou, A., Lawrence, N.D., Ek, C.H., 2016. Multi-view learning as a nonparametric nonlinear inter-battery factor analysis. arXiv preprint arXiv:1604.04939 .

Damianou, A.C., Ek, C.H., Titsias, M.K., Lawrence, N.D., 2012. Manifold relevance determination, in: Proceedings of the 29th International Coference on International Conference on Machine Learning, pp. 531–538.

de Diego, I.M., Muñoz, A., Moguerza, J.M., 2010. Methods for the combination of kernel matrices within a support vector framework. Machine learning 78, 137.

Ding, X., Yang, Y., Stein, E.A., Ross, T.J., 2017. Combining multiple resting-state fmri features during classification: optimized frameworks and their application to nicotine addiction. Frontiers in human neuroscience 11, 362.

Donini, M., Monteiro, J.M., Pontil, M., Hahn, T., Fallgatter, A.J., Shawe-Taylor, J., Mourão-Miranda, J., Initiative, A.D.N., et al., 2019. Combining heterogeneous data sources for neuroimaging based diagnosis: re-weighting and selecting what is important. NeuroImage 195, 215–231.

Dua, D., Graff, C., 2017. UCI machine learning repository. URL: http://archive.ics.uci.edu/ml.

Dukart, J., Schroeter, M.L., Mueller, K., Initiative, A.D.N., et al., 2011. Age correction in dementia–matching to a healthy brain. PloS one 6, e22193.

Džeroski, S., Demšar, D., Grbović, J., 2000. Predicting chemical parameters of river water quality from bioindicator data. Applied Intelligence 13, 7–17.

Elisseeff, A., Weston, J., 2002. A kernel method for multi-labelled classification, in: Advances in neural information processing systems, pp. 681–687.

Eloyan, A., Li, S., Muschelli, J., Pekar, J.J., Mostofsky, S.H., Caffo, B.S., 2014. Analytic programming with fmri data: A quick-start guide for statisticians using r. PloS one 9, e89470.

Friedman, J., Hastie, T., Tibshirani, R., et al., 2001. The elements of statistical learning. volume 1. Springer series in statistics New York.

Frisoni, G.B., Fox, N.C., Jack Jr, C.R., Scheltens, P., Thompson, P.M., 2010. The clinical use of structural mri in alzheimer's disease. Nature Reviews Neurology 6, 67.

Fujiwara, Y., Miyawaki, Y., Kamitani, Y., 2009. Estimating image bases for visual image reconstruction from human brain activity. Advances in neural information processing systems 22, 576–584.

Fung, G., Dundar, M., Bi, J., Rao, B., 2004. A fast iterative algorithm for fisher discriminant using heterogeneous kernels, in: Proceedings of the twenty-first international conference on Machine learning, p. 40.

Gaonkar, B., Davatzikos, C., 2013. Analytic estimation of statistical significance maps for support vector machine based multi-variate image analysis and classification. Neuroimage 78, 270–283.

Garg, A.X., Adhikari, N.K., McDonald, H., Rosas-Arellano, M.P., Devereaux, P.J., Beyene, J., Sam, J., Haynes, R.B., 2005. Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review. Jama 293, 1223–1238.

Gaser, C., Franke, K., Klöppel, S., Koutsouleris, N., Sauer, H., Initiative, A.D.N., et al., 2013. Brainage in mild cognitive impaired patients: predicting the conversion to alzheimer's disease. PloS one 8, e67346.

Georgieva, P., De la Torre, F., 2013. Robust principal component analysis for brain imaging, in: International Conference on Artificial Neural Networks, Springer. pp. 288–295.

Ghahramani, Z., 2015. Probabilistic machine learning and artificial intelligence. Nature 521, 452–459.

Gholami, B., Norton, I., Tannenbaum, A.R., Agar, N.Y., 2012. Recursive feature elimination for brain tumor classification using desorption electrospray ionization mass spectrometry imaging, in: 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE. pp. 5258–5261.

Girolami, M., Rogers, S., 2006. Variational bayesian multinomial probit regression with gaussian process priors. Neural Computation 18, 1790–1817.

Gönen, M., 2012. Bayesian supervised multilabel learning with coupled embedding and classification, in: Proceedings of the 2012 SIAM International Conference on Data Mining, SIAM. pp. 367–378.

Grant, J.E., Chamberlain, S.R., 2018. Costs and benefits of neuroimaging research in obsessive-compulsive disorder: time to take stock. CNS Spectrums 23, 298–299.

Grosenick, L., Klingenberg, B., Katovich, K., Knutson, B., Taylor, J.E., 2013. Interpretable whole-brain prediction analysis with graphnet. NeuroImage 72, 304–321.

Gunawardena, S.R., He, F., Sarrigiannis, P., Blackburn, D.J., 2020. Nonlinear classification of eeg recordings from patients with alzheimer's disease using gaussian process latent variable model. medRxiv .

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of machine learning research 3, 1157–1182.

Hansen, L.K., Larsen, J., Nielsen, F.Å., Strother, S.C., Rostrup, E., Savoy, R., Lange, N., Sidtis, J., Svarer, C., Paulson, O.B., 1999. Generalizable patterns in neuroimaging: How many principal components? NeuroImage 9, 534–544.

Hanson, S.J., Halchenko, Y.O., 2008. Brain reading using full brain support vector machines for object recognition: there is no "face" identification area. Neural Computation 20, 486–503.

Hao, X., Bao, Y., Guo, Y., Yu, M., Zhang, D., Risacher, S.L., Saykin, A.J., Yao, X., Shen, L., Initiative, A.D.N., et al., 2020. Multi-modal neuroimaging feature selection with consistent metric constraint for diagnosis of alzheimer's disease. Medical image analysis 60, 101625.

He, H., Ma, Y., 2013. Imbalanced learning: foundations, algorithms, and applications. John Wiley & Sons.

Hemmelmann, C., Horn, M., Reiterer, S., Schack, B., Süsse, T., Weiss, S., 2004. Multivariate tests for the evaluation of high-dimensional eeg data. Journal of Neuroscience Methods 139, 111–120.

Hinrich, J.L., Nielsen, S.F.V., Madsen, K.H., Morup, M., 2016. Variational group-pca for intrinsic dimensionality determination in fmri data, in: 2016 International Workshop on Pattern Recognition in Neuroimaging (PRNI), IEEE. pp. 1–4.

Hinrichs, C., Singh, V., Xu, G., Johnson, S.C., Initiative, A.D.N., et al., 2011. Predictive markers for ad in a multi-modality framework: an analysis of mci progression in the ADNI population. Neuroimage 55, 574–589.

Hotelling, H., 1992. Relations between two sets of variates, in: Breakthroughs in statistics. Springer, pp. 162–190.

Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E., 2007. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical Report 07-49. University of Massachusetts, Amherst.

Huttunen, H., Manninen, T., Kauppi, J.P., Tohka, J., 2013. Mind reading with regularized multinomial logistic regression. Machine vision and applications 24, 1311–1325.

Inza, I., Larranaga, P., Blanco, R., Cerrolaza, A.J., 2004. Filter versus wrapper gene selection approaches in dna microarray domains. Artificial intelligence in medicine 31, 91–103.

Jaakkola, T., Jordan, M., 1997. A variational approach to bayesian logistic regression models and their extensions, in: Sixth International Workshop on Artificial Intelligence and Statistics, p. 4.

Kamronn, S., Poulsen, A.T., Hansen, L.K., 2015. Multiview bayesian correlated component analysis. Neural Computation 27, 2207–2230.

Karalič, A., Bratko, I., 1997. First order regression. Machine learning 26, 147–176.

Klami, A., Kaski, S., 2007. Local dependent components, in: Proceedings of the 24th international conference on Machine learning, pp. 425–432.

Klami, A., Virtanen, S., Kaski, S., 2013. Bayesian canonical correlation analysis. Journal of Machine Learning Research 14, 965–1003.

Klöppel, S., Stonnington, C.M., Chu, C., Draganski, B., Scahill, R.I., Rohrer, J.D., Fox, N.C., Jack Jr, C.R., Ashburner, J., Frackowiak, R.S., 2008. Automatic classification of mr scans in alzheimer's disease. Brain 131, 681–689.

Knorr, F.G., Neukam, P.T., Fröhner, J.H., Mohr, H., Smolka, M.N., Marxen, M., 2020. A comparison of fmri and behavioral models for predicting inter-temporal choices. NeuroImage 211, 116634.

Krishnan, A., Williams, L.J., McIntosh, A.R., Abdi, H., 2011. Partial least squares (pls) methods for neuroimaging: a tutorial and review. Neuroimage 56, 455–475.

Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K., 2009. Attribute and simile classifiers for face verification, in: 2009 IEEE 12th International Conference on Computer Vision, IEEE. pp. 365–372.

Kursun, O., Alpaydin, E., Favorov, O.V., 2011. Canonical correlation analysis using within-class coupling. Pattern Recognition Letters 32, 134–144.

Kyono, T., Gilbert, F.J., van der Schaar, M., 2020. Improving workflow efficiency for mammography using machine learning. Journal of the American College of Radiology 17, 56–63.

Lai, C., Guo, S., Cheng, L., Wang, W., 2017. A comparative study of feature selection methods for the discriminative analysis of temporal lobe epilepsy. Frontiers in neurology 8, 633.

Lange, F.J., Ashburner, J., Smith, S.M., Andersson, J.L., 2020. A symmetric prior for the regularisation of elastic deformations: Improved anatomical plausibility in nonlinear image registration. NeuroImage 219, 116962.

Lanka, P., Rangaprakash, D., Dretsch, M.N., Katz, J.S., Denney, T.S., Deshpande, G., 2020. Supervised machine learning for diagnostic classification from large-scale neuroimaging datasets. Brain imaging and behavior 14, 2378–2416.

Lee, H., Park, B.y., Byeon, K., Won, J.H., Kim, M., Kim, S.H., Park, H., 2020. Multivariate association between brain function and eating disorders using sparse canonical correlation analysis. Plos one 15, e0237511.

Li, Y., Wu, F., Ngom, A., 2018. A review on machine learning principles for multi-view biological data integration. Briefings in bioinformatics 19, 325–340.

Lian, W., Rai, P., Salazar, E., Carin, L., 2015. Integrating features and similarities: Flexible models for heterogeneous multiview data., in: AAAI, Citeseer. pp. 2757–2763.

Long, Z., Wang, Y., Liu, X., Yao, L., 2019. Two-step paretial least square regression classifiers in brain-state decoding using functional magnetic resonance imaging. Plos one 14, e0214937.

López, M., Ramírez, J., Górriz, J.M., Álvarez, I., Salas-Gonzalez, D., Segovia, F., Chaves, R., Padilla, P., Gómez-Río, M., Initiative, A.D.N., et al., 2011. Principal component analysis-based techniques and supervised classification schemes for the early detection of alzheimer's disease. Neurocomputing 74, 1260–1271.

Lukic, A.S., Wernick, M.N., Tzikas, D.G., Chen, X., Likas, A., Galatsanos, N.P., Yang, Y., Zhao, F., Strother, S.C., 2007. Bayesian kernel methods for analysis of functional neuroimages. IEEE Transactions on Medical Imaging 26, 1613–1624.

Makris, N., Goldstein, J.M., Kennedy, D., Hodge, S.M., Caviness, V.S., Faraone, S.V., Tsuang, M.T., Seidman, L.J., 2006. Decreased volume of left and total anterior insular lobule in schizophrenia. Schizophrenia research 83, 155–171.

Mansouri, F.A., Buckley, M.J., Mahboubi, M., Tanaka, K., 2015. Behavioral consequences of selective damage to frontal pole and posterior cingulate cortices. Proceedings of the National Academy of Sciences 112, E3940–E3949.

Michel, V., Gramfort, A., Varoquaux, G., Eger, E., Thirion, B., 2011. Total variation regularization for fmri-based prediction of behavior. IEEE transactions on medical imaging 30, 1328–1340.

Milham, M.P., Fair, D., Mennes, M., Mostofsky, S.H., et al., 2012. The adhd-200 consortium: a model to advance the translational potential of neuroimaging in clinical neuroscience. Frontiers in systems neuroscience 6, 62.

Moradi, E., Pepe, A., Gaser, C., Huttunen, H., Tohka, J., Initiative, A.D.N., et al., 2015. Machine learning framework for early MRI-based alzheimer's conversion prediction in MCI subjects. Neuroimage 104, 398–412.

Muller, K.R., Anderson, C.W., Birch, G.E., 2003. Linear and nonlinear methods for brain-computer interfaces. IEEE transactions on neural systems and rehabilitation engineering 11, 165–169.

Mulugeta, G., Eckert, M.A., Vaden, K.I., Johnson, T.D., Lawson, A.B., 2017. Methods for the analysis of missing data in fmri studies. Journal of biometrics & biostatistics 8.

Münch, M.M., van de Wiel, M.A., van der Vaart, A.W., Peeters, C.F., 2021. Semi-supervised empirical bayes group-regularized factor regression. arXiv preprint arXiv:2104.02419 .

Muñoz-Romero, S., Arenas-García, J., Gómez-Verdejo, V., 2015. Sparse and kernel opls feature extraction based on eigenvalue problem solving. Pattern Recognition 48, 1797–1811.

Muñoz-Romero, S., Gomez-Verdejo, V., Arenas-Garcia, J., 2016. Regularized multivariate analysis framework for interpretable high-dimensional variable selection. IEEE Computational Intelligence Magazine 11, 24–35.

Muñoz-Romero, S., Gómez-Verdejo, V., Parrado-Hernández, E., 2017. A novel framework for parsimonious multivariate analysis. Pattern Recognition 71, 173–186.

Murphy, K.P., 2012. Machine learning: a probabilistic perspective. MIT press.

Nadeau, C., Bengio, Y., 2000. Inference for the generalization error, in: Advances in neural information processing systems, pp. 307–313.

Neal, R.M., 2012. Bayesian learning for neural networks. volume 118. Springer Science & Business Media.

Nichols, T.E., Holmes, A.P., 2002. Nonparametric permutation tests for functional neuroimaging: a primer with examples. Human brain mapping 15, 1–25.

Ojala, M., Garriga, G.C., 2010. Permutation tests for studying classifier performance. Journal of Machine Learning Research 11.

Oswal, A., Litvak, V., Brown, P., Woolrich, M., Barnes, G., 2014. Optimising beamformer regions of interest analysis. Neuroimage 102, 945–954.

Pagani, M., Salmaso, D., Rodriguez, G., Nardo, D., Nobili, F., 2009. Principal component analysis in mild and moderate alzheimer's disease—a novel approach to clinical diagnosis. Psychiatry Research: Neuroimaging 173, 8–14.

Pan, Y., Pu, W., Chen, X., Huang, X., Cai, Y., Tao, H., Xue, Z., Mackinley, M., Limongi, R., Liu, Z., et al., 2020. Morphological profiling of schizophrenia: cluster analysis of mri-based cortical thickness data. Schizophrenia bulletin 46, 623–632.

Paolini, E., Moretti, P., Compton, M.T., 2016. Delusions in first-episode psychosis: principal component analysis of twelve types of delusions and demographic and clinical correlates of resulting domains. Psychiatry research 243, 5–13.

Parrado-Hernández, E., Gómez-Verdejo, V., Martínez-Ramón, M., Shawe-Taylor, J., Alonso, P., Pujol, J., Menchón, J.M., Cardoner, N., Soriano-Mas, C., 2014. Discovering brain regions relevant to obsessive–compulsive disorder identification through bagging and transduction. Medical image analysis 18, 435–448.

Pauger, D., Wagner, H., et al., 2019. Bayesian effect fusion for categorical predictors. Bayesian Analysis 14, 341–369.

Pearson, K., 1901. Liii. on lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2, 559–572.

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

Poldrack, R.A., Gorgolewski, K.J., 2014. Making big data open: data sharing in neuroimaging. Nature neuroscience 17, 1510–1517.

Qiu, S., Lane, T., 2008. A framework for multiple kernel support vector regression and its applications to sirna efficacy prediction. IEEE/ACM Transactions on Computational Biology and Bioinformatics 6, 190–199.

Qureshi, M.N.I., Min, B., Jo, H.J., Lee, B., 2016. Multiclass classification for the differential diagnosis on the adhd subtypes using recursive feature elimination and hierarchical extreme learning machine: structural mri study. PloS one 11, e0160697.

Remeseiro, B., Bolon-Canedo, V., 2019. A review of feature selection methods in medical applications. Computers in biology and medicine 112, 103375.

Risacher, S.L., Shen, L., West, J.D., Kim, S., McDonald, B.C., Beckett, L.A., Harvey, D.J., Jack, C.R., Weiner, M.W., Saykin, A.J., et al., 2010. Longitudinal MRI atrophy biomarkers: relationship to conversion in the ADNI cohort. Neurobiology of aging 31, 1401–1418.

Rondina, J.M., Hahn, T., de Oliveira, L., Marquand, A.F., Dresler, T., Leitner, T., Fallgatter, A.J., Shawe-Taylor, J., Mourao-Miranda, J., 2013. Scors—a method based on stability for feature selection and mapping in neuroimaging. IEEE transactions on medical imaging 33, 85–98.

Rosenberg, M.D., Casey, B., Holmes, A.J., 2018. Prediction complements explanation in understanding the developing brain. Nature communications 9, 1–13.

Sartipi, S., Kalbkhani, H., Shayesteh, M.G., 2020. Diagnosis of schizophrenia from r-fmri data using ripplet transform and olpp. Multimedia Tools and Applications 79, 23401–23423.

Schork, N.J., 2019. Artificial intelligence and personalized medicine, in: Precision Medicine in Cancer Therapy. Springer, pp. 265–283.

Sevilla-Salcedo, C., 2021. Bayesian factor analysis. https://github.com/sevisal/Bayesian_Formulations/blob/main/Bayesian_FA.pdf , 13.

Sevilla-Salcedo, C., Gómez-Verdejo, V., Olmos, P.M., 2021. Sparse semi-supervised heterogeneous interbattery bayesian analysis. Pattern Recognition , 108141.

Sevilla-Salcedo, C., Gómez-Verdejo, V., Tohka, J., Initiative, A.D.N., 2020a. Regularized bagged canonical component analysis for multiclass learning in brain imaging. Neuroinformatics 18, 641–659.

Sevilla-Salcedo, C., Guerrero-López, A., Olmos, P.M., Gómez-Verdejo, V., 2020b. Bayesian sparse factor analysis with kernelized observations. arXiv preprint arXiv:2006.00968 .

Shi, S., Nathoo, F., 2018. Feature learning and classification in neuroimaging: Predicting cognitive impairment from magnetic resonance imaging, in: 2018 4th International Conference on Big Data and Information Analytics (BigDIA), IEEE. pp. 1–5.

Smith, S.M., Nichols, T.E., 2018. Statistical challenges in "big data" human neuroimaging. Neuron 97, 263–268.

Song, X., Lu, H., 2017. Multilinear regression for embedded feature selection with application to fmri analysis, in: Proceedings of the AAAI Conference on Artificial Intelligence, p. 1.

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., Vlahavas, I., 2016. Multi-target regression via input space expansion: treating targets as inputs. Machine Learning 104, 55–98.

Stefan, A.M., Gronau, Q.F., Schönbrodt, F.D., Wagenmakers, E.J., 2019. A tutorial on bayes factor design analysis using an informed prior. Behavior Research Methods 51, 1042–1058.

Stephan, K.E., Iglesias, S., Heinzle, J., Diaconescu, A.O., 2015. Translational perspectives for computational neuroimaging. Neuron 87, 716–732.

Suk, H., Lee, S., 2012. A novel bayesian framework for discriminative feature extraction in brain-computer interfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 286–299.

Tan, H., Zhang, X., Lan, L., Huang, X., Luo, Z., 2019. Nonnegative constrained graph based canonical correlation analysis for multi-view feature learning. Neural Processing Letters 50, 1215–1240.

Terzi, E., Cengiz, M.A., 2013. Bayesian hierarchical modeling for categorical longitudinal data from sedation measurements. Computational and mathematical methods in medicine 2013.

Tibber, M.S., Kirkbride, J.B., Joyce, E.M., Mutsatsa, S., Harrison, I., Barnes, T.R., Huddy, V., 2018. The component structure of the scales for the assessment of positive and negative symptoms in first-episode psychosis and its dependence on variations in analytic methods. Psychiatry research 270, 869–879.

Tipping, M.E., 2001. Sparse bayesian learning and the relevance vector machine. Journal of machine learning research 1, 211–244.

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

Tohka, J., Moradi, E., Huttunen, H., Initiative, A.D.N., et al., 2016. Comparison of feature selection techniques in machine learning for anatomical brain mri in dementia. Neuroinformatics 14, 279–296.

Toutanova, K., Johnson, M., 2008. A bayesian lda-based model for semi-supervised part-of-speech tagging, in: Advances in neural information processing systems, pp. 1521–1528.

Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I., 2011. Mulan: A java library for multi-label learning. Journal of Machine Learning Research 12, 2411–2414.

Vaden Jr, K.I., Gebregziabher, M., Kuchinsky, S.E., Eckert, M.A., 2012. Multiple imputation of missing fmri data in whole brain analysis. Neuroimage 60, 1843–1855.

Wang, C., 2007. Variational bayesian approach to canonical correlation analysis. IEEE Transactions on Neural Networks 18, 905–910.

Wang, H.T., Smallwood, J., Mourao-Miranda, J., Xia, C.H., Satterthwaite, T.D., Bassett, D.S., Bzdok, D., 2020. Finding the needle in a high-dimensional haystack: canonical correlation analysis for neuroscientists. NeuroImage 216, 116745.

Watson, D.S., Krutzinna, J., Bruce, I.N., Griffiths, C.E., McInnes, I.B., Barnes, M.R., Floridi, L., 2019. Clinical applications of machine learning algorithms: beyond the black box. Bmj 364.

Weiner, M.W., Veitch, D.P., Aisen, P.S., Beckett, L.A., Cairns, N.J., Green, R.C., Harvey, D., Jack, C.R., Jagust, W., Morris, J.C., et al., 2017. Recent publications from the alzheimer's disease neuroimaging initiative: Reviewing progress toward improved ad clinical trials. Alzheimer's & Dementia 13, e1–e85.

Williams, C., Seeger, M., 2001. Using the nyström method to speed up kernel machines, in: Leen, T., Dietterich, T., Tresp, V. (Eds.), Advances in Neural Information Processing Systems 13 (NIPS 2000), MIT Press. pp. 682–688.

Woolrich, M., Hunt, L., Groves, A., Barnes, G., 2011. Meg beamforming using bayesian pca for adaptive data covariance matrix regularization. Neuroimage 57, 1466–1479.

Wottschel, V., Chard, D.T., Enzinger, C., Filippi, M., Frederiksen, J.L., Gasperini, C., Giorgio, A., Rocca, M.A., Rovira, A., De Stefano, N., et al., 2019. Svm recursive feature elimination analyses of structural brain mri predicts near-term relapses in patients with clinically isolated syndromes suggestive of multiple sclerosis. NeuroImage: Clinical 24, 102011.

Wu, A., Nastase, S.A., Baldassano, C.A., Turk-Browne, N.B., Norman, K.A., Engelhardt, B.E., Pillow, J.W., 2021. Brain kernel: a new spatial covariance function for fmri data. bioRxiv .

Xiao, H., Rasul, K., Vollgraf, R., 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017). arXiv preprint cs.LG/1708.07747 32.

Xu, W., Li, Q., Liu, X., Zhen, Z., Wu, X., 2020. Comparison of feature selection methods based on discrimination and reliability for fmri decoding analysis. Journal of neuroscience methods 335, 108567.

Yassin, W., Nakatani, H., Zhu, Y., Kojima, M., Owada, K., Kuwabara, H., Gonoi, W., Aoki, Y., Takao, H., Natsubori, T., et al., 2020. Machine-learning classification using neuroimaging data in schizophrenia, autism, ultra-high risk and first-episode psychosis. Translational psychiatry 10, 1–11.

Ye, J., Wang, T., 2006. Regularized discriminant analysis for high dimensional, low sample size data, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 454–463.

Yin, Z., Liu, L., Liu, L., Zhang, J., Wang, Y., 2017. Dynamical recursive feature elimination technique for neurophysiological signal-based emotion recognition. Cognition, Technology & Work 19, 667–685.

Yu, W., Ormerod, J.T., Stewart, M., 2020. Variational discriminant analysis with variable selection. Statistics and Computing , 1–19.

Yu, Y., Shen, H., Zhang, H., Zeng, L.L., Xue, Z., Hu, D., 2013. Functional connectivity-based signatures of schizophrenia revealed by multiclass pattern analysis of resting-state fmri from schizophrenic patients and their healthy siblings. Biomedical engineering online 12, 10.

Yuan, L., Wang, Y., Thompson, P.M., Narayan, V.A., Ye, J., Initiative, A.D.N., et al., 2012. Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data. NeuroImage 61, 622–632.

Zhang, B., Zhou, H., Wang, L., Sung, C., 2017a. Classification based on neuroimaging data by tensor boosting, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE. pp. 1174–1179.

Zhang, G., Yin, J., Su, X., Huang, Y., Lao, Y., Liang, Z., Ou, S., Zhang, H., 2014. Augmenting multi-instance multilabel learning with sparse bayesian models for skin biopsy image analysis. BioMed research international 2014.

Zhang, S., Bamakan, S.M.H., Qu, Q., Li, S., 2018. Learning for personalized medicine: a comprehensive review from a deep learning perspective. IEEE reviews in biomedical engineering 12, 194–208.

Zhang, X., Chen, B., Liu, H., Zuo, L., Feng, B., 2016. Infinite max-margin factor analysis via data augmentation. Pattern Recognition 52, 17–32.

Zhang, X., Yan, L.F., Hu, Y.C., Li, G., Yang, Y., Han, Y., Sun, Y.Z., Liu, Z.C., Tian, Q., Han, Z.Y., et al., 2017b. Optimizing a machine learning based glioma grading system using multi-parametric mri histogram and texture features. Oncotarget 8, 47816.

Zhong, Y., Wang, H., Lu, G., Zhang, Z., Jiao, Q., Liu, Y., 2009. Detecting functional connectivity in fmri using pca and regression analysis. Brain topography 22, 134–144.

Zhuang, X., Yang, Z., Cordes, D., 2020. A technical review of canonical correlation analysis for neuroscience applications. Human Brain Mapping 41, 3807–3833.

Appendix A

SSHIBA formulation

This appendix includes a detailed explanation of the mathematical development of the SSHIBA formulation. In particular, we include the different functionalities of the model together with the predictive distribution and the semi-supervised versions.

A.1 Sparse SSHIBA

A.1.1 Generative model

We can redefine matrix $\mathbf{W}^{(\mathrm{m})}$ so that it is capable of selecting automatically both the amount of extracted features and the amount of features of the input space that are used (FS). To do so, we can establish that the distribution of $\mathbf{W}^{(\mathrm{m})}$ is composed by a different distribution for each type of selection, having

$$\mathbf{W}^{(\mathrm{m})} \sim (ARD(\alpha_0, \beta_0) \times ARD(\alpha_0^{\gamma}, \beta_0^{\gamma})). \tag{A.1}$$

This way, we can now simultaneously force sparsity by columns and by rows, having

$$\mathrm{w}_{\mathrm{d,k}}^{(\mathrm{m})} \sim \mathcal{N}\Big(0, \Big(\gamma_{\mathrm{d}}^{(\mathrm{m})}\, \alpha_{\mathrm{k}}^{(\mathrm{m})}\Big)^{-1}\Big). \tag{A.2}$$

Hence, high values of $\alpha_{\mathrm{k}}^{(\mathrm{m})}$ make the column tend to 0 (low variance). On the other hand, if $\alpha_{\mathrm{k}}^{(\mathrm{m})}$ has a low value you allow the column to have any value (high variance). The same effect will be followed by $\gamma_{\mathrm{d}}^{(\mathrm{m})}$, having that high values make the row tend to zero. Therefore, $\alpha_{\mathrm{k}}^{(\mathrm{m})}$ will have the distribution defined in Equation (2.77) and equivalently:

$$\gamma_{\mathrm{d}}^{(\mathrm{m})} \sim \Gamma(\alpha_0^{\gamma}, \beta_0^{\gamma}) \tag{A.3}$$

A.1.2 Variational inference

We now need to include the change in the distribution of $\mathbf{W}^{(\mathrm{m})}$ in the q distributions of the different variables. Concretely, we need to change the distribution of $\mathbf{W}^{(\mathrm{m})}$ and $\boldsymbol{\alpha}^{(\mathrm{m})}$ while the distributions of both $\mathbf{Z}$ and $\tau^{(\mathrm{m})}$ keep the same definitions.

As most variables need to calculate the expectation over the distribution of $\mathbf{X}^{(\mathrm{m})}$, let's define here the log-probability for later usage:

$$\begin{aligned}
&\ln p\Big(\mathbf{X}^{(\mathrm{m})} \,|\, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \tau^{(\mathrm{m})}\Big) \approx \\
&\approx \sum_{\mathrm{n}=1}^{\mathrm{N}} \ln\Bigg(\mathcal{N}\bigg(\mathbf{z}_{\mathrm{n},:}\,\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}, \Big(\tau^{(\mathrm{m})}\Big)^{-1} I\bigg)\Bigg) + \text{const} \\
&= \sum_{\mathrm{n}=1}^{\mathrm{N}} \Bigg(-\frac{1}{2}\ln\bigg|\Big(\tau^{(\mathrm{m})}\Big)^{-1} I\bigg| - \frac{1}{2}\Big(\mathbf{x}^{(\mathrm{m})}_{\mathrm{n},:} - \mathbf{z}_{\mathrm{n},:}\,\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\Big)\,\tau^{(\mathrm{m})}\Big(\mathbf{x}^{(\mathrm{m})}_{\mathrm{n},:} - \mathbf{z}_{\mathrm{n},:}\,\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\Big)^{T}\Bigg) + \text{const}
\end{aligned} \tag{A.4}$$

Distribution of $\mathbf{W}^{(\mathrm{m})}$

Analysing the graphic model and using the mean field approximation, we get

$$\begin{aligned}
\ln\Big(q^*\Big(\mathbf{W}^{(\mathrm{m})}\Big)\Big) &= \mathbb{E}_{\mathbf{Z},\boldsymbol{\alpha}^{(\mathrm{m})},\tau^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})}, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \boldsymbol{\alpha}^{(\mathrm{m})}, \tau^{(\mathrm{m})}\Big)\Big)\Big] \\
&= \mathbb{E}_{\mathbf{Z},\tau^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})} \,|\, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \tau^{(\mathrm{m})}\Big)\Big)\Big] \\
&\quad + \mathbb{E}_{\boldsymbol{\alpha}^{(\mathrm{m})},\boldsymbol{\gamma}^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{W}^{(\mathrm{m})} \,|\, \boldsymbol{\alpha}^{(\mathrm{m})}, \boldsymbol{\gamma}^{(\mathrm{m})}\Big)\Big)\Big] + \text{const},
\end{aligned} \tag{A.5}$$

having that the first term of this summation is

$$\begin{aligned}
\mathbb{E}_{\mathbf{Z},\tau^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})} \,|\, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \tau^{(\mathrm{m})}\Big)\Big)\Big] &= \sum_{\mathrm{n}=1}^{\mathrm{N}} \Bigg(\langle\tau^{(\mathrm{m})}\rangle\, \mathbf{x}^{(\mathrm{m})}_{\mathrm{n},:}\, \mathbf{W}^{(\mathrm{m})}\langle\mathbf{z}_{\mathrm{n},:}\rangle \\
&\quad - \frac{\langle\tau^{(\mathrm{m})}\rangle}{2}\sum_{\mathrm{d}=1}^{\mathrm{D_m}}\Big(\mathbf{w}^{(\mathrm{m})}_{\mathrm{d},:}\langle\mathbf{z}^{\mathrm{T}}_{\mathrm{n},:}\,\mathbf{z}_{\mathrm{n},:}\rangle\,\mathbf{w}^{(\mathrm{m})^{\mathrm{T}}}_{\mathrm{d},:}\Big)\Bigg),
\end{aligned} \tag{A.6}$$

and the second term can be calculated as

$$\ln\Big(p\Big(\mathbf{W}^{(\mathrm{m})} \,|\, \boldsymbol{\alpha}^{(\mathrm{m})}, \boldsymbol{\gamma}^{(\mathrm{m})}\Big)\Big) = \sum_{\mathrm{d}=1}^{\mathrm{D_m}}\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\Bigg(\frac{1}{2}\ln\Big(\alpha^{(\mathrm{m})}_{\mathrm{k}}\,\gamma^{(\mathrm{m})}_{\mathrm{d}}\Big) - \frac{\alpha^{(\mathrm{m})}_{\mathrm{k}}}{2}\mathrm{w}^{(\mathrm{m})^2}_{\mathrm{d,k}}\Bigg) + \text{const},$$

$$\mathbb{E}_{\boldsymbol{\alpha}^{(\mathrm{m})},\boldsymbol{\gamma}^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{W}^{(\mathrm{m})} \,|\, \boldsymbol{\alpha}^{(\mathrm{m})}, \boldsymbol{\gamma}^{(\mathrm{m})}\Big)\Big)\Big] = \sum_{\mathrm{d}=1}^{\mathrm{D_m}}\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\Big(-\frac{1}{2}\mathrm{w}^{(\mathrm{m})}_{\mathrm{d,k}}\langle\alpha^{(\mathrm{m})}_{\mathrm{k}}\rangle\langle\gamma^{(\mathrm{m})}_{\mathrm{d}}\rangle\mathrm{w}^{(\mathrm{m})}_{\mathrm{d,k}}\Big) + \text{const.} \tag{A.7}$$

Finally, joining Equations (A.6) and (A.7), we have that the optimum solution for variable $\mathbf{W}^{(\mathrm{m})}$ is

$$\begin{aligned}
\ln\Big(q^*\Big(\mathbf{W}^{(\mathrm{m})}\Big)\Big) &= \sum_{\mathrm{d}=1}^{\mathrm{D_m}}\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\Big(-\frac{1}{2}\mathrm{w}^{(\mathrm{m})}_{\mathrm{d,k}}\langle\alpha^{(\mathrm{m})}_{\mathrm{k}}\rangle\langle\gamma^{(\mathrm{m})}_{\mathrm{d}}\rangle\mathrm{w}^{(\mathrm{m})}_{\mathrm{d,k}}\Big) + \sum_{\mathrm{n}=1}^{\mathrm{N}}\Bigg(\langle\tau^{(\mathrm{m})}\rangle\,\mathbf{x}^{(\mathrm{m})}_{\mathrm{n},:}\,\mathbf{W}^{(\mathrm{m})}\langle\mathbf{z}_{\mathrm{n},:}\rangle \\
&\quad - \frac{\langle\tau^{(\mathrm{m})}\rangle}{2}\sum_{\mathrm{d}=1}^{\mathrm{D_m}}\Big(\mathbf{w}^{(\mathrm{m})}_{\mathrm{d},:}\langle\mathbf{z}^{\mathrm{T}}_{\mathrm{n},:}\,\mathbf{z}_{\mathrm{n},:}\rangle\,\mathbf{w}^{(\mathrm{m})^{\mathrm{T}}}_{\mathrm{d},:}\Big)\Bigg) + \text{const} \\
&= \sum_{\mathrm{d}=1}^{\mathrm{D_m}}(-\frac{1}{2}\,\mathbf{w}^{(\mathrm{m})}_{\mathrm{d},:}\Big(\mathit{diag}(\langle\alpha^{(\mathrm{m})}_{\mathrm{k}}\rangle)\langle\gamma^{(\mathrm{m})}_{\mathrm{d}}\rangle + \langle\tau^{(\mathrm{m})}\rangle\langle\mathbf{z}^{\mathrm{T}}_{\mathrm{n},:}\,\mathbf{z}_{\mathrm{n},:}\rangle\Big)\,\mathbf{w}^{(\mathrm{m})^{\mathrm{T}}}_{\mathrm{d},:} \\
&\quad + \langle\tau^{(\mathrm{m})}\rangle\,\mathbf{w}^{(\mathrm{m})}_{\mathrm{d},:}\langle\mathbf{Z}\rangle^{T}\,\mathbf{x}^{(\mathrm{m})^{\mathrm{T}}}_{:,\mathrm{d}}\,) + \text{const.}
\end{aligned} \tag{A.8}$$

This way, comparing the results with the normal distribution, we can identify terms and obtain the resulting distribution

$$q^*\Big(\mathbf{W}^{(\mathrm{m})}\Big) \;=\; \prod_{\mathrm{d}=1}^{\mathrm{D_m}}\Big(\mathcal{N}\Big(\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})}\,|\langle\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})}\rangle,\Sigma_{W_d^{(m)}}\Big)\Big), \tag{A.9}$$

where the variance is

$$\Sigma_{W_d^{(m)}}^{-1} \;=\; diag(\langle\boldsymbol{\alpha}^{(\mathrm{m})}\rangle)\langle\gamma_{\mathrm{d}}^{(\mathrm{m})}\rangle+\langle\tau^{(\mathrm{m})}\rangle\langle\mathbf{Z}^{\mathrm{T}}\,\mathbf{Z}\rangle, \tag{A.10}$$

and the mean can be expressed as

$$\langle\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})}\rangle \;=\; \langle\tau^{(\mathrm{m})}\rangle\,\mathbf{x}_{:,\mathrm{d}}^{(\mathrm{m})^{\mathrm{T}}}\langle\mathbf{Z}\rangle^{T}\Sigma_{W_d^{(m)}}. \tag{A.11}$$

Distribution of $\boldsymbol{\alpha}^{(\mathrm{m})}$

Analysing $\boldsymbol{\alpha}^{(\mathrm{m})}$ we get that the approximation of the posterior distribution is

$$\ln\Big(q^*\Big(\boldsymbol{\alpha}^{(\mathrm{m})}\Big)\Big) \;=\; \mathbb{E}_{\mathbf{W}^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{W}^{(\mathrm{m})}\,|\,\boldsymbol{\alpha}^{(\mathrm{m})},\boldsymbol{\gamma}^{(\mathrm{m})}\Big)\Big)\Big]+\mathbb{E}\Big[\ln\Big(p\Big(\boldsymbol{\alpha}^{(\mathrm{m})}\Big)\Big)\Big]+\mathrm{const.} \tag{A.12}$$

Following the previous development, we can determine both terms independently, having that the first term is

$$\ln\Big(p\Big(\mathbf{W}^{(\mathrm{m})}\,|\,\boldsymbol{\alpha}^{(\mathrm{m})},\boldsymbol{\gamma}^{(\mathrm{m})}\Big)\Big) \;=\; \sum_{\mathrm{d}=1}^{\mathrm{D_m}}\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\Big(\frac{1}{2}\ln\Big(\alpha_{\mathrm{k}}^{(\mathrm{m})}\,\gamma_{\mathrm{d}}^{(\mathrm{m})}\Big)-\frac{1}{2}\,\mathrm{w}_{\mathrm{d,k}}^{(\mathrm{m})}\,\alpha_{\mathrm{k}}^{(\mathrm{m})}\,\gamma_{\mathrm{d}}^{(\mathrm{m})}\,\mathrm{w}_{\mathrm{d,k}}^{(\mathrm{m})}\Big)+\mathrm{const}$$

$$\begin{aligned}\mathbb{E}\Big[\ln\Big(p\Big(\mathbf{W}^{(\mathrm{m})}\,|\,\boldsymbol{\alpha}^{(\mathrm{m})},\boldsymbol{\gamma}^{(\mathrm{m})}\Big)\Big)\Big] \;=\; &\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\Big(\frac{\mathrm{D_m}}{2}\ln\Big(\alpha_{\mathrm{k}}^{(\mathrm{m})}\Big)\\ &-\frac{1}{2}\,\alpha_{\mathrm{k}}^{(\mathrm{m})}\sum_{\mathrm{d}=1}^{\mathrm{D_m}}\Big(\langle\gamma_{\mathrm{d}}^{(\mathrm{m})}\rangle\langle\mathbf{w}_{:,\mathrm{k}}^{(\mathrm{m})^{\mathrm{T}}}\,\mathbf{w}_{:,\mathrm{k}}^{(\mathrm{m})}\rangle\Big)\Big)+\mathrm{const},\end{aligned} \tag{A.13}$$

and the second term is

$$\ln\Big(p\Big(\alpha_{\mathrm{k}}^{(\mathrm{m})}\Big)\Big) \;=\; -\,\beta_0\,\alpha_{\mathrm{k}}^{(\mathrm{m})}+(\alpha_0-1)\ln\Big(\alpha_{\mathrm{k}}^{(\mathrm{m})}\Big)+\mathrm{const}$$

$$\mathbb{E}\Big[\ln\Big(p\Big(\boldsymbol{\alpha}^{(\mathrm{m})}\Big)\Big)\Big] \;=\; \sum_{\mathrm{k}=1}^{\mathrm{K_c}}\Big(\ln\Big(p\Big(\alpha_{\mathrm{k}}^{(\mathrm{m})}\Big)\Big)\Big)=\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\Big(-\beta_0\,\alpha_{\mathrm{k}}^{(\mathrm{m})}+(\alpha_0-1)\ln\Big(\alpha_{\mathrm{k}}^{(\mathrm{m})}\Big)\Big)+\mathrm{const.} \tag{A.14}$$

So the distribution can be written as

$$q^*\Big(\boldsymbol{\alpha}^{(\mathrm{m})}\Big) \;=\; \prod_{\mathrm{k}=1}^{\mathrm{K_c}}\Big(\Gamma\Big(\alpha_{\mathrm{k}}^{(\mathrm{m})}\,|a_{\alpha_{\mathrm{k}}^{(\mathrm{m})}},b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}}\Big)\Big), \tag{A.15}$$

where the variance is

$$a_{\alpha_{\mathrm{k}}^{(\mathrm{m})}} \;=\; \frac{\mathrm{D_m}}{2}+\alpha_0, \tag{A.16}$$

and the mean can be expressed as

$$b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}} \;=\; \beta_0+\frac{1}{2}\sum_{\mathrm{d}=1}^{\mathrm{D_m}}\Big(\langle\gamma_{\mathrm{d}}^{(\mathrm{m})}\rangle\langle\mathrm{w}_{\mathrm{d,k}}^{(\mathrm{m})}\,\mathrm{w}_{\mathrm{d,k}}^{(\mathrm{m})}\rangle\Big). \tag{A.17}$$

Distribution of $\boldsymbol{\gamma}^{(\mathrm{m})}$

In this case the posterior approximation is

$$\ln\Big(q^*\Big(\boldsymbol{\gamma}^{(\mathrm{m})}\Big)\Big) = \mathbb{E}_{\mathbf{W}^{(\mathrm{m})},\boldsymbol{\alpha}^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{W}^{(\mathrm{m})}\,|\,\boldsymbol{\alpha}^{(\mathrm{m})}\Big)\Big)\Big] + \mathbb{E}\Big[\ln\Big(p\Big(\boldsymbol{\alpha}^{(\mathrm{m})}\Big)\Big)\Big] + \mathrm{const.} \quad (A.18)$$

The results of this equation are the same as the ones we obtained for the variable $\boldsymbol{\alpha}^{(\mathrm{m})}$, but applying the expectation to different variables, having that the distribution will be equivalent to the one of $\boldsymbol{\alpha}^{(\mathrm{m})}$:

$$q^*\Big(\boldsymbol{\gamma}^{(\mathrm{m})}\Big) = \prod_{\mathrm{k}=1}^{\mathrm{K_c}}\Big(\Gamma\Big(\gamma_{\mathrm{d}}^{(\mathrm{m})}\,|a_{\gamma_{\mathrm{d}}^{(\mathrm{m})}}, b_{\gamma_{\mathrm{d}}^{(\mathrm{m})}}\Big)\Big), \quad (A.19)$$

where the variance is

$$a_{\gamma_{\mathrm{d}}^{(\mathrm{m})}} = \frac{\mathrm{K_c}}{2} + \alpha_0^{\gamma}, \quad (A.20)$$

and the mean can be expressed as

$$b_{\gamma_{\mathrm{d}}^{(\mathrm{m})}} = \beta_0^{\gamma} + \frac{1}{2}\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\Big(\langle\alpha_{\mathrm{k}}^{(\mathrm{m})}\rangle\langle\mathrm{w}_{\mathrm{d,k}}^{(\mathrm{m})}\,\mathrm{w}_{\mathrm{d,k}}^{(\mathrm{m})}\rangle\Big). \quad (A.21)$$

A.1.3 Update of the lower bound

In this context, the lower bound is modified by the terms of the input space, as the sparsity is forced only for the features. Hence, the only terms of entropy that are modified are

$$\mathbb{E}_q\Big[\ln\Big(q\Big(\mathbf{W}^{(1)}\Big)\Big)\Big] = \sum_{\mathrm{d}=1}^{\mathrm{D_m}}\Big(\frac{1}{2}\ln|\Sigma_{\mathbf{w}_{\mathrm{d},:}^{(1)}}|\Big) + \mathrm{const}, \quad (A.22)$$

and

$$\mathbb{E}_q\Big[\ln\Big(q\Big(\boldsymbol{\gamma}^{(1)}\Big)\Big)\Big] = -\sum_{\mathrm{d}=1}^{\mathrm{D_m}}\Big(\ln\Big(b_{\gamma_{\mathrm{d}}^{(1)}}\Big)\Big) + \mathrm{const.} \quad (A.23)$$

Conversely, the terms related to $\mathbb{E}[\ln(p(...))]$ will have the same expression with the same two exceptions, which are calculated as

$$\mathbb{E}\Big[\ln\Big(p\Big(\boldsymbol{\gamma}^{(1)}\Big)\Big)\Big] = \sum_{d=1}^{\mathrm{D_1}}\left(-\beta_0^{\gamma}\frac{a_{\gamma_{\mathrm{d}}^{(1)}}}{b_{\gamma_{\mathrm{d}}^{(1)}}} + (\alpha_0^{\gamma}-1)\Big(\psi\Big(a_{\gamma_{\mathrm{d}}^{(1)}}\Big) - \ln\Big(b_{\gamma_{\mathrm{d}}^{(1)}}\Big)\Big)\right) + \mathrm{const}, \quad (A.24)$$

and

$$\begin{aligned}\mathbb{E}\Big[\ln\Big(p\Big(\mathbf{W}^{(1)}\,|\,\boldsymbol{\alpha}^{(1)},\boldsymbol{\gamma}^{(1)}\Big)\Big)\Big] &= \frac{\mathrm{D_1}}{2}\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\Big(\psi\Big(a_{\alpha_{\mathrm{k}}^{(1)}}\Big) - \ln\Big(b_{\alpha_{\mathrm{k}}^{(1)}}\Big)\Big) - \sum_{\mathrm{k}=1}^{\mathrm{K_c}}\left(\frac{a_{\alpha_{\mathrm{k}}^{(1)}}}{b_{\alpha_{\mathrm{k}}^{(1)}}}\Big(b_{\alpha_{\mathrm{k}}^{(1)}} - \beta_0\Big)\right)\\ &+ \frac{\mathrm{K_c}}{2}\sum_{d=1}^{\mathrm{D_1}}\Big(\psi\Big(a_{\gamma_{\mathrm{d}}^{(1)}}\Big) - \ln\Big(b_{\gamma_{\mathrm{d}}^{(1)}}\Big)\Big) + \mathrm{const.}\end{aligned} \quad (A.25)$$

Using the lower Bound equation, developed in section BIBFA in Sevilla-Salcedo (2021), we get

$$\begin{aligned} L_q \;=\; & -\frac{1}{2}\,\mathrm{Tr}\{\langle \mathbf{Z}^{\mathrm{T}}\,\mathbf{Z}\rangle\} - \sum_{\mathrm{m}=1}^{\mathrm{M}}\left(\left(\frac{\mathrm{D_m}}{2}+\alpha_0-1\right)\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\left(\ln\left(b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}}\right)\right)\right) \\ & -\sum_{\mathrm{m}=1}^{\mathrm{M}}\left(\left(\frac{\mathrm{D_m}}{2}+\alpha_0^{\tau}-1\right)(\ln(b_{\tau^{(\mathrm{m})}}))\right)+\sum_{\mathrm{m}=1}^{\mathrm{M}}(\ln(b_{\tau^{(\mathrm{m})}})) \\ & -\frac{N}{2}\ln|\Sigma_{\mathbf{Z}}|-\sum_{\mathrm{m}=1}^{\mathrm{M}}\left(\frac{\mathrm{D_m}}{2}\ln|\Sigma_{\mathbf{W}^{(\mathrm{m})}}|\right)+\sum_{\mathrm{m}=1}^{\mathrm{M}}\left(\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\left(\ln\left(b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}}\right)\right)\right), \end{aligned} \tag{A.26}$$

and we can now include these new terms and discard the terms that are constant, so we have that the updated lower bound is

$$\begin{aligned} L_q \;=\; & -\frac{1}{2}\,\mathrm{Tr}\{\langle \mathbf{Z}^{\mathrm{T}}\,\mathbf{Z}\rangle\} \\ & +\sum_{\mathrm{m}=1}^{\mathrm{M}}\left(\left(\frac{N\,\mathrm{D_m}}{2}+\alpha_0^{\tau}-1\right)(-\ln(b_{\tau^{(\mathrm{m})}}))\right) \\ & +\left(\frac{\mathrm{D_1}}{2}+\alpha_0-1\right)\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\left(-\ln\left(b_{\alpha_{\mathrm{k}}^{(1)}}\right)\right)+\left(\frac{\mathrm{K_c}}{2}+\alpha_0-1\right)\sum_{d=1}^{\mathrm{D_1}}\left(-\ln\left(b_{\gamma_{\mathrm{d}}^{(1)}}\right)\right) \\ & +\left(\frac{\mathrm{D_2}}{2}+\alpha_0-1\right)\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\left(-\ln\left(b_{\alpha_{\mathrm{k}}^{(2)}}\right)\right)-\sum_{d=1}^{\mathrm{D_1}}\left(\beta_0^{\gamma}\,\frac{a_{\gamma_{\mathrm{d}}^{(1)}}}{b_{\boldsymbol{\gamma}^{(1)}}}\right) \\ & -\frac{N}{2}\ln|\Sigma_{\mathbf{Z}}|+\sum_{\mathrm{m}=1}^{\mathrm{M}}\left(\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\left(\ln\left(b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}}\right)\right)\right)+\sum_{\mathrm{m}=1}^{\mathrm{M}}(\ln(b_{\tau^{(\mathrm{m})}}))+\sum_{d=1}^{\mathrm{D_1}}\left(\ln\left(b_{\gamma_{\mathrm{d}}^{(1)}}\right)\right) \\ & -\sum_{\mathrm{m}=1}^{\mathrm{M}}\sum_{\mathrm{d}=1}^{\mathrm{D_m}}\left(\frac{1}{2}\ln|\Sigma_{\mathbf{W}^{(\mathrm{m})}}|\right) \end{aligned} \tag{A.27}$$

where the entropy of the variable $\mathbf{W}^{(\mathrm{m})}$ will change for the view in which we impose the sparsity.

A.2 Multidimensional binary SSHIBA

A.2.1 Generative model

In this case, the probability distribution of the model is modified to include the new variable $\mathbf{T}^{(\mathrm{m_t})}$, where $\mathrm{m_t}$ represents the view with multidimensional binary data, as seen in Figure 4.3:

$$\begin{aligned} & p\left(\Theta|\,\mathbf{T}^{\{\mathcal{M}_{\mathrm{t}}\}},\mathbf{X}^{\{\mathcal{M}_r\}}\right) \;\approx \\ & q(\mathbf{Z})\prod_{m_t\in\mathcal{M}_t}\left(\prod_{n=1}^{N}q\left(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\right)\right)\prod_{\mathrm{m}=1}^{\mathrm{M}}q\left(\mathbf{W}^{(\mathrm{m})}\right)q\left(\mathbf{b}^{(\mathrm{m})}\right)q\left(\boldsymbol{\alpha}^{(\mathrm{m})}\right)q\left(\tau^{(\mathrm{m})}\right)q\left(\boldsymbol{\gamma}^{(\mathrm{m})}\right). \end{aligned} \tag{A.28}$$

Comparing this distribution with the distribution of BIBFA, we can see that the only term that changes is the distribution of $\mathbf{T}^{(\mathrm{m_t})}$, so this can be defined as

$$\begin{aligned} p\left(\mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m_t})} \,|\, \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})}\right) &= e^{\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})}\, \mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m_t})}} \sigma\left(-\,\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})}\right) \geq \\ & e^{\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})}\, \mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m_t})}} \sigma\left(\xi_{\mathrm{n,d}}^{(\mathrm{m})}\right) e^{-\frac{\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})} + \xi_{\mathrm{n,d}}^{(\mathrm{m_t})}}{2} - \lambda\left(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\right)\left(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})^2} - \xi_{\mathrm{n,d}}^{(\mathrm{m_t})^2}\right)}, \end{aligned} \tag{A.29}$$

where $\sigma\left(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\right) = \frac{1}{1+e^{-\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}}}$ and $\lambda\left(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\right) = \frac{1}{2\,\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}}\left(\sigma\left(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\right) - \frac{1}{2}\right)$. Therefore, the distribution can be calculated as

$$\begin{aligned} & p\left(\mathbf{T}^{(\mathrm{m_t})} \,|\, \mathbf{X}^{(\mathrm{m_t})}\right) \geq h\left(\mathbf{X}^{(\mathrm{m_t})}, \boldsymbol{\xi}^{(\mathrm{m_t})}\right) \\ &= \prod_{\mathrm{n=1}}^{\mathrm{N}} \prod_{\mathrm{d=1}}^{\mathrm{D_{m_t}}} \left(\sigma\left(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\right) e^{\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})}\, \mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m_t})} - \frac{\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})} + \xi_{\mathrm{n,d}}^{(\mathrm{m_t})}}{2} - \lambda\left(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\right)\left(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})^2} - \xi_{\mathrm{n,d}}^{(\mathrm{m_t})^2}\right)}\right). \end{aligned} \tag{A.30}$$

A.2.2 Variational inference

Following the definitions presented in the semi-supervised section, we can see the inclusion of the variable $\mathbf{T}^{(\mathrm{m_t})}$ only affects $\mathbf{X}^{(\mathrm{m_t})}$. Therefore, we have to adapt its update rules as

$$\ln\left(q^*\left(\mathbf{X}^{(\mathrm{m_t})}\right)\right) = \mathbb{E}\left[\ln\left(h\left(\mathbf{X}^{(\mathrm{m_t})}, \boldsymbol{\xi}^{(\mathrm{m_t})}\right)\right) + \ln\left(p\left(\mathbf{X}^{(\mathrm{m_t})} \,|\, \mathbf{W}^{(\mathrm{m_t})}, \mathbf{Z}, \tau^{(\mathrm{m_t})}, \boldsymbol{\alpha}^{(\mathrm{m_t})}\right)\right)\right]. \tag{A.31}$$

Like in the previous extension, we can analyse both terms independently. Regarding the second term, we can follow a similar procedure to the one done for Equation (A.4):

$$\begin{aligned} & \mathbb{E}\left[\ln\left(p\left(\mathbf{X}^{(\mathrm{m_t})} \,|\, \mathbf{W}^{(\mathrm{m_t})}, \mathbf{Z}, \tau^{(\mathrm{m_t})}, \boldsymbol{\alpha}^{(\mathrm{m_t})}\right)\right)\right] \\ &= -\frac{\langle\tau^{(\mathrm{m_t})}\rangle}{2} \sum_{\mathrm{n=1}}^{\mathrm{N}} \left(\mathbf{x}_{\mathrm{n,:}}^{(\mathrm{m_t})}\, \mathbf{x}_{\mathrm{n,:}}^{(\mathrm{m_t})^{\mathrm{T}}} - 2\langle\mathbf{z}_{\mathrm{n,:}}\rangle\langle\mathbf{W}^{(\mathrm{m_t})^{\mathrm{T}}}\rangle\, \mathbf{x}_{\mathrm{n,:}}^{(\mathrm{m_t})^{\mathrm{T}}}\right) \end{aligned} \tag{A.32}$$

$$\begin{aligned} & \mathbb{E}\left[\ln\left(h\left(\mathbf{X}^{(\mathrm{m_t})}, \boldsymbol{\xi}^{(\mathrm{m_t})}\right)\right)\right] \\ &= \mathbb{E}\left[\sum_{\mathrm{n=1}}^{\mathrm{N}} \sum_{\mathrm{d=1}}^{\mathrm{D_t}} \left(\ln\left(\sigma\left(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\right)\right) + \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})}\, \mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m_t})} - \frac{1}{2}\left(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})} + \xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\right) - \lambda\left(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\right)\left(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})^2} - \xi_{\mathrm{n,d}}^{(\mathrm{m_t})^2}\right)\right)\right] \\ &= \sum_{\mathrm{n=1}}^{\mathrm{N}} \sum_{\mathrm{d=1}}^{\mathrm{D_t}} \left(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})}\, \mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m_t})} - \frac{1}{2}\, \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})} - \lambda\left(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\right) \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})^2}\right) + \mathrm{const} \\ &= \sum_{\mathrm{n=1}}^{\mathrm{N}} \left(\left(\mathbf{t}_{\mathrm{n,:}}^{(\mathrm{m_t})} - \frac{1}{2}\right) \mathbf{x}_{\mathrm{n,:}}^{(\mathrm{m})^{\mathrm{T}}} - \mathbf{x}_{\mathrm{n,:}}^{(\mathrm{m_t})}\, \Lambda_{\boldsymbol{\xi}_{\mathrm{n,:}}^{(\mathrm{m_t})}}\, \mathbf{x}_{\mathrm{n,:}}^{(\mathrm{m_t})^{\mathrm{T}}}\right), \end{aligned} \tag{A.33}$$

where $\Lambda_{\boldsymbol{\xi}_{\mathrm{n,:}}^{(\mathrm{m_t})}}$ is a diagonal matrix which diagonal elements are $\lambda\left(\xi_{\mathrm{n,1}}^{(\mathrm{m_t})}\right), \lambda\left(\xi_{\mathrm{n,2}}^{(\mathrm{m_t})}\right), \ldots, \lambda\left(\xi_{\mathrm{n,D_t}}^{(\mathrm{m_t})}\right)$. Therefore, joining both terms we have that

$$\begin{aligned} \ln\left(q^*\left(\mathbf{X}^{(\mathrm{m_t})}\right)\right) &= \sum_{\mathrm{n=1}}^{\mathrm{N}} \left(\left(\mathbf{t}_{\mathrm{n,:}}^{(\mathrm{m_t})} - \frac{1}{2} + \langle\tau^{(\mathrm{m_t})}\rangle\langle\mathbf{z}_{\mathrm{n,:}}\rangle\langle\mathbf{W}^{(\mathrm{m_t})^{\mathrm{T}}}\rangle\right) \mathbf{x}_{\mathrm{n,:}}^{(\mathrm{m_t})^{\mathrm{T}}} \right. \\ & \left. -\frac{1}{2}\, \mathbf{x}_{\mathrm{n,:}}^{(\mathrm{m_t})}\left(\langle\tau^{(\mathrm{m_t})}\rangle I + 2\Lambda_{\boldsymbol{\xi}_{\mathrm{n,:}}^{(\mathrm{m_t})}}\right) \mathbf{x}_{\mathrm{n,:}}^{(\mathrm{m_t})^{\mathrm{T}}}\right) + \mathrm{const}. \end{aligned} \tag{A.34}$$

This way we obtain that the distribution is as follows:

$$q^*\Big(\mathbf{X}^{(\mathrm{m_t})}\Big) \;=\; \prod_{\mathrm{n}=1}^{\mathrm{N}}\Big(\mathcal{N}\Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\,|\langle\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\rangle,\Sigma_{\mathbf{X}^{(\mathrm{m_t})}}\Big)\Big), \tag{A.35}$$

where

$$\Sigma^{-1}_{\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}} \;=\; \langle\tau^{(\mathrm{m_t})}\rangle I + 2\Lambda_{\boldsymbol{\xi}_{\mathrm{n},:}^{(\mathrm{m_t})}}, \tag{A.36}$$

and

$$\langle\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\rangle \;=\; \left(\mathbf{t}_{\mathrm{n},:}^{(\mathrm{m_t})} - \frac{1}{2} + \langle\tau^{(\mathrm{m_t})}\rangle\langle\mathbf{z}_{\mathrm{n},:}\rangle\langle\mathbf{W}^{(\mathrm{m_t})^{\mathrm{T}}}\rangle\right)\Sigma_{\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}}. \tag{A.37}$$

A.2.3 Update of the lower bound

By the change of the model the lower bound now needs to consider the variable $\mathbf{X}^{(\mathrm{m_t})}$ as a model parameter that is inferred. As so, we need to include the entropy of this parameter:

$$\mathbb{E}_q\Big[\ln\Big(q\Big(\mathbf{X}^{(\mathrm{m_t})}\Big)\Big)\Big] \;=\; \sum_{\mathrm{n}=1}^{\mathrm{N}}\left(\frac{\mathrm{D_t}}{2}\ln(2\pi e) + \frac{1}{2}\ln|\Sigma_{\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}}|\right). \tag{A.38}$$

At the same time, we now need to consider the variable $\mathbf{T}^{(2)}$ as a known variable, thus needing to include its influence into the lower bound:

$$\begin{aligned}
&\mathbb{E}_q\Big[\ln\Big(p\Big(\mathbf{T}^{(\mathrm{m_t})}\,|\,\mathbf{X}^{(\mathrm{m_t})}\Big)\Big)\Big] = \mathbb{E}_q\Big[\ln\Big(h\Big(\mathbf{X}^{(\mathrm{m_t})},\boldsymbol{\xi}^{(\mathrm{m_t})}\Big)\Big)\Big]\\
&= \mathbb{E}_q\left[\sum_{\mathrm{n}=1}^{\mathrm{N}}\sum_{\mathrm{d}=1}^{\mathrm{D_t}}\left(\ln\Big(\sigma\Big(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\Big)\Big) + \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})}\,\mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m_t})} - \frac{1}{2}\Big(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})} + \xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\Big) - \lambda\Big(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\Big)\Big(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})^2} - \xi_{\mathrm{n,d}}^{(\mathrm{m_t})^2}\Big)\right)\right]\\
&= \sum_{\mathrm{n}=1}^{\mathrm{N}}\sum_{\mathrm{d}=1}^{\mathrm{D_t}}\left(\ln\Big(\sigma\Big(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\Big)\Big) + \langle\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})}\rangle\,\mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m_t})} - \frac{1}{2}\Big(\langle\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})}\rangle + \xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\Big) - \lambda\Big(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\Big)\Big(\mathbb{E}\Big[\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})^2}\Big] - \xi_{\mathrm{n,d}}^{(\mathrm{m_t})^2}\Big)\right).
\end{aligned} \tag{A.39}$$

A.2.4 Variational parameter calculation ($\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}$)

In order to calculate this new parameter we have to maximise the lower bound. This way we have to maximise the terms of the lower bound that depend on $\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}$, Equation (A.39). We can now set the derivative with respect to the parameter $\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}$ equal to zero:

$$\frac{\partial L(q)}{\partial\,\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}} \;=\; \lambda'\Big(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\Big)\Big(\mathbb{E}\Big[\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})^2}\Big] - \xi_{\mathrm{n,d}}^{(\mathrm{m_t})^2}\Big) = 0, \tag{A.40}$$

where $\lambda'\Big(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\Big)$ is a monotonic function of $\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}$ for $\xi_{\mathrm{n,d}}^{(\mathrm{m_t})} \geq 0$, and we can restrict attention to nonnegative values of $\boldsymbol{\xi}^{(\mathrm{m_t})}$ due to symmetry around $\boldsymbol{\xi}^{(\mathrm{m_t})} = 0$

$$\begin{aligned}
\lambda'\Big(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\Big) \neq 0 \longrightarrow \xi_{\mathrm{n,d}}^{(\mathrm{m_t})^{new^2}} &= \mathbb{E}\Big[\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})^2}\Big] = \langle\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})}\rangle^2 + \Sigma_{\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})}}\\
\boldsymbol{\xi}_{\mathrm{n},:}^{(\mathrm{m_t})^{new^2}} &= \langle\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\rangle^2 + diag\Big(\Sigma_{\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}}\Big).
\end{aligned} \tag{A.41}$$

A.3 Categorical SSHIBA

A.3.1 Generative model

In this case, the structure is similar to the one followed by the multilabel case seen in the previous section. Nevertheless, in this case, we are going to take some new considerations. First of all, in this case we are going to work with $\tau^{(\mathrm{m_t})} = 1$, as seen in Girolami and Rogers (2006), and, at the same time, the variable $\mathbf{t}^{(\mathrm{m_t})}$ is a vector instead of a matrix.

We can establish the probability distribution of our model looking at Figure 4.3:

$$p\Big(\Theta |\, \mathbf{t}^{\{\mathcal{M}_\mathrm{t}\}}, \mathbf{X}^{\{\mathcal{M}_r\}}\Big) \approx \\ q(\mathbf{Z}) \prod_{m_t \in \mathcal{M}_t} \left(\prod_{n=1}^{N} q\Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\Big) \right) \prod_{\mathrm{m}=1}^{\mathrm{M}} q\Big(\mathbf{W}^{(\mathrm{m})}\Big) q\Big(\mathbf{b}^{(\mathrm{m})}\Big) q\Big(\boldsymbol{\alpha}^{(\mathrm{m})}\Big) q\Big(\tau^{(\mathrm{m})}\Big) q\Big(\boldsymbol{\gamma}^{(\mathrm{m})}\Big). \tag{A.42}$$

In order to work with multiclass classification, we use a multinomial probit model. This implies that $\dim\Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\Big) = \mathrm{D_t}$ (number of classes) and, at the same time

$$\mathrm{t}_{\mathrm{n}}^{(\mathrm{m_t})} = i \qquad if \qquad \mathrm{x}_{\mathrm{n,i}}^{(\mathrm{m_t})} = \max_{1 \le \mathrm{d} \le \mathrm{D_t}} \Big\{ \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})} \Big\}. \tag{A.43}$$

Therefore, the multinomial probit takes the following form (Girolami and Rogers, 2006):

$$p\Big(\mathrm{t}_{\mathrm{n}}^{(\mathrm{m_t})} = i |\, \mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\Big) = \delta\Big(\mathrm{x}_{\mathrm{n,i}}^{(\mathrm{m_t})} > \mathrm{x}_{\mathrm{n,j}}^{(\mathrm{m_t})}\Big) \qquad /j \neq i \tag{A.44}$$

$$p\Big(\mathrm{t}_{\mathrm{n}}^{(\mathrm{m_t})} = i |\, \mathbf{z}_{\mathrm{n},:}, \mathbf{W}^{(\mathrm{m_t})}\Big) = \mathbb{E}_{p(u)}\left[\prod_{j \neq i} \Big(\Phi\Big(u + \mathrm{y}_{\mathrm{n,i}}^{(\mathrm{m_t})} - \mathrm{y}_{\mathrm{n,j}}^{(\mathrm{m_t})} \Big) \Big) \right], \tag{A.45}$$

where $\mathbf{y}_{\mathrm{n},:}^{(\mathrm{m_t})} = \mathbf{z}_{\mathrm{n},:} \mathbf{W}^{(\mathrm{m_t})^{\mathrm{T}}}$ and $p(u) \sim \mathcal{N}(0, 1)$.

A.3.2 Variational inference

As seen before, the addition of these new terms will only affect $q(\mathbf{X}^{(\mathrm{m})})$:

$$\ln\Big(q^*\Big(\mathbf{X}^{(\mathrm{m_t})}\Big)\Big) = \mathbb{E}\left[\sum_{\mathrm{n}=1}^{\mathrm{N}} \Big(\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m_t})} \,|\, \mathbf{W}^{(\mathrm{m_t})}, \mathbf{Z}, \boldsymbol{\alpha}^{(\mathrm{m_t})}, 1 \Big)\Big)\Big) \right] + \mathbb{E}\left[\sum_{\mathrm{n}=1}^{\mathrm{N}} \Big(\ln\Big(p\Big(\mathrm{t}_{\mathrm{n}}^{(\mathrm{m_t})} \,|\, \mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})} \Big)\Big)\Big) \right] \tag{A.46}$$

Let $\langle \mathbf{Y}^{(\mathrm{m_t})} \rangle = \langle \mathbf{Z} \rangle \langle \mathbf{W}^{(\mathrm{m_t})^{\mathrm{T}}} \rangle$. If $\mathrm{t}_{\mathrm{n}}^{(\mathrm{m_t})} = i$, then:

$$q^*\Big(\mathbf{X}^{(\mathrm{m_t})}\Big) = \frac{1}{\xi_n} \mathcal{N}_u\Big(\langle \mathbf{Y}^{(\mathrm{m_t})} \rangle, I \Big) \delta\Big(\mathrm{x}_{\mathrm{n,i}}^{(\mathrm{m_t})} > \mathrm{x}_{\mathrm{n,j}}^{(\mathrm{m_t})}\, \forall i \neq j \Big), \tag{A.47}$$

where

$$\xi_n = \mathbb{E}_{p(u)}\left[\prod_{j \neq i} \Big(\Phi\Big(u + \langle \mathrm{y}_{\mathrm{n,i}}^{(\mathrm{m_t})} \rangle - \langle \mathrm{y}_{\mathrm{n,j}}^{(\mathrm{m_t})} \rangle \Big) \Big) \right]. \tag{A.48}$$

And, thus, assuming again that $\mathrm{t}_{\mathrm{n}}^{(\mathrm{m_t})} = i$, the posterior expectations are

$$\langle \mathrm{x}_{\mathrm{n,i}}^{(\mathrm{m_t})} \rangle = \langle \mathrm{y}_{\mathrm{n,i}}^{(\mathrm{m_t})} \rangle + \sum_{j \neq i} \Big(\langle \mathrm{y}_{\mathrm{n,j}}^{(\mathrm{m_t})} \rangle - \langle \mathrm{x}_{\mathrm{n,j}}^{(\mathrm{m_t})} \rangle \Big) \tag{A.49}$$

$$\langle \mathrm{x}_{\mathrm{n,j}}^{(\mathrm{m_t})} \rangle = \langle \mathrm{y}_{\mathrm{n,j}}^{(\mathrm{m_t})} \rangle - \frac{1}{\xi_n} \mathbb{E}_{p(u)} \left[\mathcal{N}_u \Big(\langle \mathrm{y}_{\mathrm{n,j}}^{(\mathrm{m_t})} \rangle - \langle \mathrm{y}_{\mathrm{n,i}}^{(\mathrm{m_t})} \rangle, 1 \Big) \prod_{k \neq i \neq j} \Big(\Phi \Big(u + \langle \mathrm{y}_{\mathrm{n,i}}^{(\mathrm{m_t})} \rangle - \langle \mathrm{y}_{\mathrm{n,k}}^{(\mathrm{m_t})} \rangle \Big) \Big) \right]. \tag{A.50}$$

A.4 SSHIBA with bias term

A.4.1 Generative model

Throughout this model adaptation, we have been modelling the data as seen in Klami et al. (2013): as a combination of matrices $\mathbf{Z}$ and $\mathbf{W}^{(\mathrm{m})}$. Nevertheless, this definition might be outdated when referring to the heterogeneous model, where we might have input data of different types, either categorical, mutilabel or real. Concretely, by working with data that is not normalised when training the model, different views might incur in different weights for each view in the latent space. This implies that not normalised input data, such as categorical data, will have a higher weight in the training process and, therefore, the model obtained is biased.

To avoid that, we propose the inclusion of a bias term in the definition of the input data

$$\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})} \,|\, \mathbf{z}_{\mathrm{n},:} \sim \mathcal{N}(\mathbf{z}_{\mathrm{n},:} \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}} + \mathbf{b}^{(\mathrm{m})}, \Sigma^{(m)}), \tag{A.51}$$

where

$$\mathbf{b}^{(\mathrm{m})} \sim \mathcal{N}(0, I_{\mathrm{D_m}}) \tag{A.52}$$

is the bias term which will allow us to nullify the effect of the not normalisation of the data.

A.4.2 Variational inference: Real view

With this change in matrix $\mathbf{X}^{(\mathrm{m})}$ we need to determine the distribution of the variables that directly relate to this distribution. Thus, the distribution of variables $\mathbf{b}^{(\mathrm{m})}$, $\mathbf{W}^{(\mathrm{m})}$ and $\mathbf{Z}$ need to be recalculated.

Distribution of $\mathbf{b}^{(\mathrm{m})}$

First of all, we can calculate the distribution q of the new variable included in the model

$$\begin{aligned} \ln\Big(q^*\Big(\mathbf{b}^{(\mathrm{m})}\Big)\Big) &= \mathbb{E}_{\mathbf{Z}, \mathbf{W}^{(\mathrm{m})}, \tau^{(\mathrm{m})}} \Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})}, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \boldsymbol{\alpha}^{(\mathrm{m})}, \tau^{(\mathrm{m})}, \mathbf{b}^{(\mathrm{m})}\Big)\Big)\Big] \\ = \mathbb{E}_{\mathbf{Z}, \mathbf{W}^{(\mathrm{m})}, \tau^{(\mathrm{m})}} &\Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})} \,|\, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \tau^{(\mathrm{m})}, \mathbf{b}^{(\mathrm{m})}\Big)\Big)\Big] + \mathbb{E}\Big[\ln\Big(p\Big(\mathbf{b}^{(\mathrm{m})}\Big)\Big)\Big] + \mathrm{const}, \end{aligned} \tag{A.53}$$

where the prior of the bias can be determined equivalently to the prior of variable $\mathbf{Z}$

$$\ln\Big(p\Big(\mathbf{b}^{(\mathrm{m})}\Big)\Big) = \ln(\mathcal{N}(0, I)) = -\frac{1}{2} \mathbf{b}^{(\mathrm{m})} \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} + \mathrm{const},$$

and the remaining term of the distribution can be calculated as

$$\begin{aligned}
&\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})} \,|\, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \tau^{(\mathrm{m})}, \mathbf{b}^{(\mathrm{m})}\Big)\Big) \\
=\;& \sum_{\mathrm{n}=1}^{\mathrm{N}} \ln\Big(\mathcal{N}\Big(\mathbf{z}_{\mathrm{n},:}\, \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}} + \mathbf{b}^{(\mathrm{m})}, \Big(\tau^{(\mathrm{m})}\Big)^{-1} I\Big)\Big) + \text{const} \\
=\;& -\frac{\tau^{(\mathrm{m})}}{2} \sum_{\mathrm{n}=1}^{\mathrm{N}} \Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}\, \mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})^{\mathrm{T}}} - 2\, \mathbf{z}_{\mathrm{n},:}\, \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\, \mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})^{\mathrm{T}}} + \mathbf{z}_{\mathrm{n},:}\, \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\, \mathbf{W}^{(\mathrm{m})}\, \mathbf{z}_{\mathrm{n},:}^{\mathrm{T}} \\
& -2\, \mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} + 2\, \mathbf{z}_{\mathrm{n},:}\, \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} + \mathbf{b}^{(\mathrm{m})}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}}\Big) + \text{const} \\
=\;& \sum_{\mathrm{n}=1}^{\mathrm{N}} \Big(\tau^{(\mathrm{m})} \Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})} - \mathbf{z}_{\mathrm{n},:}\, \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\Big)\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}}\Big) - \frac{N\, \tau^{(\mathrm{m})}}{2}\, \mathbf{b}^{(\mathrm{m})}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} + \text{const.} \qquad \text{(A.54)}
\end{aligned}$$

This way the expectation can be calculated as

$$\begin{aligned}
&\mathbb{E}_{\mathbf{Z}, \mathbf{W}^{(\mathrm{m})}, \tau^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})}, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \boldsymbol{\alpha}^{(\mathrm{m})}, \tau^{(\mathrm{m})}, \mathbf{b}^{(\mathrm{m})}\Big)\Big)\Big] \\
=\;& \langle\tau^{(\mathrm{m})}\rangle \sum_{\mathrm{n}=1}^{\mathrm{N}} \Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})} - \langle\mathbf{z}_{\mathrm{n},:}\rangle \langle\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\rangle\Big)\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} - \frac{N\langle\tau^{(\mathrm{m})}\rangle + 1}{2}\, \mathbf{b}^{(\mathrm{m})}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}}. \qquad \text{(A.55)}
\end{aligned}$$

Once this expectation is calculated, we can determine that the distribution followed by the parameter is given by

$$q^*\Big(\mathbf{b}^{(\mathrm{m})}\Big) \;=\; \mathcal{N}\Big(\mathbf{b}^{(\mathrm{m})} \,|\langle\mathbf{b}^{(\mathrm{m})}\rangle, \Sigma_{\mathbf{b}^{(\mathrm{m})}}\Big), \qquad \text{(A.56)}$$

where the variance is

$$\Sigma_{\mathbf{b}^{(\mathrm{m})}}^{-1} \;=\; \Big(N\langle\tau^{(\mathrm{m})}\rangle + 1\Big) I_{\mathrm{D_m}}, \qquad \text{(A.57)}$$

and the mean is

$$\langle\mathbf{b}^{(\mathrm{m})}\rangle \;=\; \langle\tau^{(\mathrm{m})}\rangle \sum_{\mathrm{n}=1}^{\mathrm{N}} \Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})} - \langle\mathbf{z}_{\mathrm{n},:}\rangle \langle\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\rangle\Big) \Sigma_{\mathbf{b}^{(\mathrm{m})}}. \qquad \text{(A.58)}$$

Distribution of $\mathbf{W}^{(\mathrm{m})}$

In this case, the distribution of variable $\mathbf{W}^{(\mathrm{m})}$ is given by

$$\begin{aligned}
\ln\Big(q^*\Big(\mathbf{W}^{(\mathrm{m})}\Big)\Big) \;&=\; \mathbb{E}_{\mathbf{Z}, \tau^{(\mathrm{m})}, \mathbf{b}^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})}, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \boldsymbol{\alpha}^{(\mathrm{m})}, \tau^{(\mathrm{m})}, \mathbf{b}^{(\mathrm{m})}\Big)\Big)\Big] \\
&=\; \mathbb{E}_{\mathbf{Z}, \tau^{(\mathrm{m})}, \mathbf{b}^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})} \,|\, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \tau^{(\mathrm{m})}, \mathbf{b}^{(\mathrm{m})}\Big)\Big)\Big] \\
&\quad + \mathbb{E}_{\boldsymbol{\alpha}^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{W}^{(\mathrm{m})} \,|\, \boldsymbol{\alpha}^{(\mathrm{m})}\Big)\Big)\Big] + \text{const}, \qquad \text{(A.59)}
\end{aligned}$$

where the second term was previously calculated and the first term consist on applying the expectation to Equation (A.54)

$$\begin{aligned}
&\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})} \,|\, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \tau^{(\mathrm{m})}, \mathbf{b}^{(\mathrm{m})}\Big)\Big) \\
= &-\frac{\langle\tau^{(\mathrm{m})}\rangle}{2}\sum_{\mathrm{n=1}}^{\mathrm{N}}\Big(-2\langle\mathbf{z}_{\mathrm{n},:}\rangle\,\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\,\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})^{\mathrm{T}}} + \langle\mathbf{z}_{\mathrm{n},:}\,\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\,\mathbf{W}^{(\mathrm{m})}\,\mathbf{z}_{\mathrm{n},:}^{\mathrm{T}}\rangle + 2\langle\mathbf{b}^{(\mathrm{m})}\rangle\,\mathbf{W}^{(\mathrm{m})}\langle\mathbf{z}_{\mathrm{n},:}^{\mathrm{T}}\rangle\Big) + \mathrm{const} \\
= &-\frac{\tau^{(\mathrm{m})}}{2}\sum_{\mathrm{n=1}}^{\mathrm{N}}\sum_{\mathrm{d=1}}^{\mathrm{D_m}}\Big(-2\langle\mathbf{z}_{\mathrm{n},:}\rangle\,\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})^{\mathrm{T}}}\,\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})^{\mathrm{T}}} + \langle\mathbf{z}_{\mathrm{n},:}\,\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})^{\mathrm{T}}}\,\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})}\,\mathbf{z}_{\mathrm{n},:}^{\mathrm{T}}\rangle + 2\langle\mathrm{b}_{\mathrm{d}}^{(\mathrm{m})}\rangle\,\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})}\langle\mathbf{z}_{\mathrm{n},:}^{\mathrm{T}}\rangle\Big) + \mathrm{const} \\
= &-\frac{\langle\tau^{(\mathrm{m})}\rangle}{2}\sum_{\mathrm{n=1}}^{\mathrm{N}}\sum_{\mathrm{d=1}}^{\mathrm{D_m}}\Big(-2\,\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})^{\mathrm{T}}}\langle\mathbf{z}_{\mathrm{n},:}\rangle\,\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})^{\mathrm{T}}} + 2\langle\mathrm{b}_{\mathrm{d}}^{(\mathrm{m})}\rangle\langle\mathbf{z}_{\mathrm{n},:}\rangle\,\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})^{\mathrm{T}}} + \mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})}\langle\mathbf{z}_{\mathrm{n},:}^{\mathrm{T}}\,\mathbf{z}_{\mathrm{n},:}\rangle\,\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})^{\mathrm{T}}}\Big) + \mathrm{const},
\end{aligned} \tag{A.60}$$

and applying (2.62) we get

$$\begin{aligned}
&\mathbb{E}_{\mathbf{Z},\tau^{(\mathrm{m})},\mathbf{b}^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})}, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \boldsymbol{\alpha}^{(\mathrm{m})}, \tau^{(\mathrm{m})}, \mathbf{b}^{(\mathrm{m})}\Big)\Big)\Big] \\
= &\sum_{\mathrm{n=1}}^{\mathrm{N}}\sum_{\mathrm{d=1}}^{\mathrm{D_m}}\Big(\langle\tau^{(\mathrm{m})}\rangle\Big(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})^{\mathrm{T}}}\langle\mathbf{z}_{\mathrm{n},:}\rangle - \langle\mathrm{b}_{\mathrm{d}}^{(\mathrm{m})}\rangle\langle\mathbf{z}_{\mathrm{n},:}\rangle\Big)\,\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})^{\mathrm{T}}} \\
&-\frac{1}{2}\,\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})}\Big(diag(\langle\boldsymbol{\alpha}^{(\mathrm{m})}\rangle) + \langle\tau^{(\mathrm{m})}\rangle\langle\mathbf{z}_{\mathrm{n},:}^{\mathrm{T}}\,\mathbf{z}_{\mathrm{n},:}\rangle\Big)\,\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})^{\mathrm{T}}}\Big) + \mathrm{const}.
\end{aligned} \tag{A.61}$$

Identifying terms, we have that the q distribution of the variable is

$$q^*\Big(\mathbf{W}^{(\mathrm{m})}\Big) \;=\; \prod_{\mathrm{d=1}}^{\mathrm{D_m}}\mathcal{N}\Big(\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})}\,|\langle\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})}\rangle, \Sigma_{\mathbf{w}_{\mathrm{d},:}^{(\mathrm{m})}}\Big), \tag{A.62}$$

with variance

$$\Sigma_{\mathbf{W}^{(\mathrm{m})}}^{-1} \;=\; diag(\langle\boldsymbol{\alpha}^{(\mathrm{m})}\rangle) + \langle\tau^{(\mathrm{m})}\rangle\langle\mathbf{Z}^{\mathrm{T}}\,\mathbf{Z}\rangle, \tag{A.63}$$

and mean

$$\langle\mathbf{W}^{(\mathrm{m})}\rangle \;=\; \langle\tau^{(\mathrm{m})}\rangle\Big(\mathbf{X}^{(\mathrm{m})} - \mathbb{1}_{\mathrm{N}}\langle\mathbf{b}^{(\mathrm{m})}\rangle\Big)^{T}\langle\mathbf{Z}\rangle\Sigma_{\mathbf{W}^{(\mathrm{m})}}, \tag{A.64}$$

where $\mathbb{1}_N$ is a row vector of ones of dimension N.

Distribution of Z

We can now calculate the distribution of this variable as

$$\begin{aligned}
\ln(q^*(\mathbf{Z})) \;=&\; \mathbb{E}_{\mathbf{W}^{(\mathrm{m})},\tau^{(\mathrm{m})},\mathbf{b}^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})}, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \tau^{(\mathrm{m})}, \mathbf{b}^{(\mathrm{m})}\Big)\Big)\Big] \\
=&\; \sum_{\mathrm{m=1}}^{\mathrm{M}}\mathbb{E}_{\mathbf{W}^{(\mathrm{m})},\tau^{(\mathrm{m})},\mathbf{b}^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})} \,|\, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \tau^{(\mathrm{m})}, \mathbf{b}^{(\mathrm{m})}\Big)\Big)\Big] + \mathbb{E}[\ln(p(\mathbf{Z}))] + \mathrm{const}
\end{aligned} \tag{A.65}$$

where the term related to the prior of variable $\mathbf{Z}$ was previously determined. Equivalently to the previous cases, the first term can be calculated as in Equation (A.60)

$$\begin{aligned}&\mathbb{E}_{\mathbf{W}^{(\mathrm{m})},\tau^{(\mathrm{m})},\mathbf{b}^{(\mathrm{m})}}\left[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})}\,|\,\mathbf{W}^{(\mathrm{m})},\mathbf{Z},\tau^{(\mathrm{m})},\mathbf{b}^{(\mathrm{m})}\Big)\Big)\right]\\ =& \sum_{\mathrm{n=1}}^{\mathrm{N}}\left(\langle\tau^{(\mathrm{m})}\rangle\Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}\langle\mathbf{W}^{(\mathrm{m})}\rangle-\langle\mathbf{b}^{(\mathrm{m})}\rangle\langle\mathbf{W}^{(\mathrm{m})}\rangle\Big)\,\mathbf{z}_{\mathrm{n},:}^{\mathrm{T}}-\frac{\langle\tau^{(\mathrm{m})}\rangle}{2}\,\mathbf{z}_{\mathrm{n},:}\langle\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\,\mathbf{W}^{(\mathrm{m})}\rangle\,\mathbf{z}_{\mathrm{n},:}^{\mathrm{T}}\right)+\mathrm{const}\end{aligned} \tag{A.66}$$

Therefore, joining the both terms, the distribution has the form:

$$\begin{aligned}\ln(q^*(\mathbf{Z})) \;=\; &\sum_{\mathrm{n=1}}^{\mathrm{N}}\left(\sum_{\mathrm{m=1}}^{\mathrm{M}}\Big(\langle\tau^{(\mathrm{m})}\rangle\Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}\langle\mathbf{W}^{(\mathrm{m})}\rangle-\langle\mathbf{b}^{(\mathrm{m})}\rangle\langle\mathbf{W}^{(\mathrm{m})}\rangle\Big)\,\mathbf{z}_{\mathrm{n},:}^{\mathrm{T}}\right.\\ &\left.-\frac{1}{2}\,\mathbf{z}_{\mathrm{n},:}\Big(I+\langle\tau^{(\mathrm{m})}\rangle\langle\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\,\mathbf{W}^{(\mathrm{m})}\rangle\Big)\,\mathbf{z}_{\mathrm{n},:}^{\mathrm{T}}\Big)\right)+\mathrm{const}\end{aligned} \tag{A.67}$$

This way, we can now identify terms, having that:

$$q^*(\mathbf{Z}) \;=\; \prod_{n=1}^{N}\mathcal{N}(\mathbf{z}_{\mathrm{n},:}\,|\langle\mathbf{z}_{\mathrm{n},:}\rangle,\Sigma_{\mathbf{Z}}) \tag{A.68}$$

where the variance is

$$\Sigma_{\mathbf{Z}}^{-1} \;=\; I+\sum_{\mathrm{m=1}}^{\mathrm{M}}\Big(\langle\tau^{(\mathrm{m})}\rangle\langle\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\,\mathbf{W}^{(\mathrm{m})}\rangle\Big) \tag{A.69}$$

and the mean can be expressed as

$$\langle\mathbf{Z}\rangle \;=\; \sum_{\mathrm{m=1}}^{\mathrm{M}}\Big(\langle\tau^{(\mathrm{m})}\rangle\Big(\mathbf{X}^{(\mathrm{m})}-\mathbb{1}_N\langle\mathbf{b}^{(\mathrm{m})}\rangle\Big)\langle\mathbf{W}^{(\mathrm{m})}\rangle\Big)\Sigma_{\mathbf{Z}} \tag{A.70}$$

where $\mathbb{1}_N$ is a row vector of ones of dimension N.

Distribution of $\tau^{(\mathrm{m})}$

Again, we determine the posterior distribution of $\tau^{(\mathrm{m})}$

$$\ln\Big(q^*\Big(\tau^{(\mathrm{m})}\Big)\Big) \;=\; \mathbb{E}_{\mathbf{W}^{(\mathrm{m})},\mathbf{Z},\mathbf{b}^{(\mathrm{m})}}\left[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})}\,|\,\mathbf{W}^{(\mathrm{m})},\mathbf{Z},\tau^{(\mathrm{m})},\mathbf{b}^{(\mathrm{m})}\Big)\Big)\right]+\mathbb{E}\left[\ln\Big(p\Big(\tau^{(\mathrm{m})}\Big)\Big)\right]+\mathrm{const}. \tag{A.71}$$

Similarly to what was done in equation (A.6):

$$
\begin{aligned}
\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})} \,|\, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \tau^{(\mathrm{m})}, \mathbf{b}^{(\mathrm{m})}\Big)\Big) &= \sum_{\mathrm{n=1}}^{\mathrm{N}} \ln\Big(\mathcal{N}\Big(\mathbf{z}_{\mathrm{n,:}}\, \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}} + \mathbf{b}^{(\mathrm{m})}, \Big(\tau^{(\mathrm{m})}\Big)^{-1} I\Big)\Big) + \text{const} \\
= \sum_{\mathrm{n=1}}^{\mathrm{N}} \sum_{\mathrm{d=1}}^{\mathrm{D_m}} \left(\frac{1}{2} \ln\Big|\tau^{(\mathrm{m})}\Big| - \frac{\tau^{(\mathrm{m})}}{2}\Big(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})} - \mathbf{z}_{\mathrm{n,:}}\, \mathbf{w}_{\mathrm{d,:}}^{(\mathrm{m})^{\mathrm{T}}} - \mathrm{b}_{\mathrm{d}}^{(\mathrm{m})}\Big)^2\right) + \text{const} \\
= \frac{\mathrm{D_m}\,\mathrm{N}}{2} \ln\Big(\tau^{(\mathrm{m})}\Big) - \frac{\tau^{(\mathrm{m})}}{2} \sum_{\mathrm{n=1}}^{\mathrm{N}} \sum_{\mathrm{d=1}}^{\mathrm{D_m}} \Big(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})^2} - 2\, \mathbf{w}_{\mathrm{d,:}}^{(\mathrm{m})}\, \mathbf{z}_{\mathrm{n,:}}^{\mathrm{T}}\, \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})} + \Big(\mathbf{z}_{\mathrm{n,:}}\, \mathbf{w}_{\mathrm{d,:}}^{(\mathrm{m})^{\mathrm{T}}}\Big)^2 - 2\, \mathrm{b}_{\mathrm{d}}^{(\mathrm{m})}\, \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})} \\
+ 2\, \mathrm{b}_{\mathrm{d}}^{(\mathrm{m})}\, \mathbf{w}_{\mathrm{d,:}}^{(\mathrm{m})}\, \mathbf{z}_{\mathrm{n,:}}^{\mathrm{T}} + \mathrm{b}_{\mathrm{d}}^{(\mathrm{m})^2}\Big) + \text{const} \\
= \frac{\mathrm{D_m}\,\mathrm{N}}{2} \ln\Big(\tau^{(\mathrm{m})}\Big) - \frac{\tau^{(\mathrm{m})}}{2} \left(\sum_{\mathrm{n=1}}^{\mathrm{N}} \sum_{\mathrm{d=1}}^{\mathrm{D_m}} \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})^2} - 2 \sum_{\mathrm{d=1}}^{\mathrm{D_m}} \mathbf{w}_{\mathrm{d,:}}^{(\mathrm{m})}\, \mathbf{Z}^{\mathrm{T}}\, \mathbf{x}_{:,\mathrm{d}}^{(\mathrm{m})} + \sum_{\mathrm{d=1}}^{\mathrm{D_m}} \mathbf{w}_{\mathrm{d,:}}^{(\mathrm{m})}\, \mathbf{Z}^{\mathrm{T}}\, \mathbf{Z}\, \mathbf{w}_{\mathrm{d,:}}^{(\mathrm{m})^{\mathrm{T}}} \right. \\
\left. - 2 \sum_{\mathrm{n=1}}^{\mathrm{N}} \mathbf{x}_{\mathrm{n,:}}^{(\mathrm{m})}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} + 2 \sum_{\mathrm{n=1}}^{\mathrm{N}} \mathbf{z}_{\mathrm{n,:}}\, \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} + N\, \mathbf{b}^{(\mathrm{m})}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} \right) + \text{const} \\
= \frac{\mathrm{D_m}\,\mathrm{N}}{2} \ln\Big(\tau^{(\mathrm{m})}\Big) - \frac{\tau^{(\mathrm{m})}}{2} \left(\sum_{\mathrm{n=1}}^{\mathrm{N}} \sum_{\mathrm{d=1}}^{\mathrm{D_m}} \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})^2} - 2\, \mathrm{Tr}\Big\{\mathbf{W}^{(\mathrm{m})}\, \mathbf{Z}^{\mathrm{T}}\, \mathbf{X}^{(\mathrm{m})}\Big\} + \mathrm{Tr}\Big\{\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\, \mathbf{W}^{(\mathrm{m})}\, \mathbf{Z}^{\mathrm{T}}\, \mathbf{Z}\Big\} \right. \\
\left. - 2 \sum_{\mathrm{n=1}}^{\mathrm{N}} \mathbf{x}_{\mathrm{n,:}}^{(\mathrm{m})}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} + 2 \sum_{\mathrm{n=1}}^{\mathrm{N}} \mathbf{z}_{\mathrm{n,:}}\, \mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} + N\, \mathbf{b}^{(\mathrm{m})}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}} \right) + \text{const}. \qquad \text{(A.72)}
\end{aligned}
$$

If we now calculate the expectation, we have that

$$
\begin{aligned}
\mathbb{E}_{\mathbf{W}^{(\mathrm{m})},\mathbf{Z},\mathbf{b}^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})} \,|\, \mathbf{W}^{(\mathrm{m})}, \mathbf{Z}, \tau^{(\mathrm{m})}\Big)\Big)\Big] &= \frac{\mathrm{D_m}\,\mathrm{N}}{2} \ln\Big(\tau^{(\mathrm{m})}\Big) - \frac{\tau^{(\mathrm{m})}}{2} \left(\sum_{\mathrm{n=1}}^{\mathrm{N}} \sum_{\mathrm{d=1}}^{\mathrm{D_m}} \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})^2} \right. \\
- 2\, \mathrm{Tr}\Big\{\langle\mathbf{W}^{(\mathrm{m})}\rangle \langle\mathbf{Z}^{\mathrm{T}}\rangle\, \mathbf{X}^{(\mathrm{m})}\Big\} + \mathrm{Tr}\Big\{\langle\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\, \mathbf{W}^{(\mathrm{m})}\rangle \langle\mathbf{Z}^{\mathrm{T}}\, \mathbf{Z}\rangle\Big\} - 2 \sum_{\mathrm{n=1}}^{\mathrm{N}} \mathbf{x}_{\mathrm{n,:}}^{(\mathrm{m})} \langle\mathbf{b}^{(\mathrm{m})^{\mathrm{T}}}\rangle \\
\left. + 2 \sum_{\mathrm{n=1}}^{\mathrm{N}} \langle\mathbf{z}_{\mathrm{n,:}}\rangle \langle\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\rangle \langle\mathbf{b}^{(\mathrm{m})^{\mathrm{T}}}\rangle + N \langle\mathbf{b}^{(\mathrm{m})}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}}\rangle \right) + \text{const}. \qquad \text{(A.73)}
\end{aligned}
$$

So, if we join together both expectation elements and identify distribution terms as we did for the case of the standard BIBFA, we have that the new distribution is

$$
q^*\Big(\tau^{(\mathrm{m})}\Big) \;=\; \Gamma\Big(\tau^{(\mathrm{m})} \,|a_{\tau^{(\mathrm{m})}}, b_{\tau^{(\mathrm{m})}}\Big), \qquad \text{(A.74)}
$$

where the parameter $a_{\tau^{(\mathrm{m})}}$ is

$$
a_{\tau^{(\mathrm{m})}} \;=\; \frac{\mathrm{D_m}\,\mathrm{N}}{2} + \alpha_0^{\tau}, \qquad \text{(A.75)}
$$

and the parameter $b_{\tau^{(\mathrm{m})}}$ can be expressed as

$$
\begin{aligned}
b_{\tau^{(\mathrm{m})}} \;=\; \beta_0^{\tau} + \frac{1}{2} &\left(\sum_{\mathrm{n=1}}^{\mathrm{N}} \sum_{\mathrm{d=1}}^{\mathrm{D_m}} \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})^2} - 2\, \mathrm{Tr}\Big\{\langle\mathbf{W}^{(\mathrm{m})}\rangle \langle\mathbf{Z}^{\mathrm{T}}\rangle\, \mathbf{X}^{(\mathrm{m})}\Big\} + \mathrm{Tr}\Big\{\langle\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\, \mathbf{W}^{(\mathrm{m})}\rangle \langle\mathbf{Z}^{\mathrm{T}}\, \mathbf{Z}\rangle\Big\} \right. \\
&\left. - 2 \sum_{\mathrm{n=1}}^{\mathrm{N}} \mathbf{x}_{\mathrm{n,:}}^{(\mathrm{m})} \langle\mathbf{b}^{(\mathrm{m})^{\mathrm{T}}}\rangle + 2 \sum_{\mathrm{n=1}}^{\mathrm{N}} \langle\mathbf{z}_{\mathrm{n,:}}\rangle \langle\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\rangle \langle\mathbf{b}^{(\mathrm{m})^{\mathrm{T}}}\rangle + N \langle\mathbf{b}^{(\mathrm{m})}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}}\rangle \right), \qquad \text{(A.76)}
\end{aligned}
$$

where $\langle\mathbf{b}^{(\mathrm{m})}\, \mathbf{b}^{(\mathrm{m})^{\mathrm{T}}}\rangle$ is the mean of a noncentral chi-squared distribution.

A.4.3 Variational inference: Sparse view

If we include sparsity in the features, the only term that is affected by the inclusion of the bias is the one related to variable $\mathbf{W}^{(\mathrm{m})}$. This variable, as we have seen for the case of working with a real view, is modified only in its mean value when considering the new variable $\mathbf{b}^{(\mathrm{m})}$. At the same time, we have seen that the inclusion of the sparsity in matrix $\mathbf{W}^{(\mathrm{m})}$ only affects the covariance matrix. For this reason we can conclude that the sparse version of matrix $\mathbf{W}^{(\mathrm{m})}$ will have the same covariance as without a bias term and the mean will be the same as in the bias case.

A.4.4 Variational inference: Multilabel view

With this new element, only the distribution of variable $\mathbf{X}^{(\mathrm{m})}$ will change for multilabel views, while the distribution of the new variable $\mathbf{b}^{(\mathrm{m})}$ is the same as in Equation (A.56).

$$\begin{aligned}\ln\Big(q^*\Big(\mathbf{X}^{(\mathrm{m})}\Big)\Big) &= \mathbb{E}_{\mathbf{Z},\mathbf{W}^{(\mathrm{m})},\tau^{(\mathrm{m})},\mathbf{b}^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})},\mathbf{W}^{(\mathrm{m})},\mathbf{Z},\boldsymbol{\alpha}^{(\mathrm{m})},\tau^{(\mathrm{m})},\mathbf{b}^{(\mathrm{m})}\Big)\Big)\Big] \\ &= \mathbb{E}\Big[\ln\Big(h\Big(\mathbf{X}^{(\mathrm{m})},\xi\Big)\Big)\Big] + \mathbb{E}_{\mathbf{Z},\mathbf{W}^{(\mathrm{m})},\tau^{(\mathrm{m})},\mathbf{b}^{(\mathrm{m})}}\Big[\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})}\,|\,\mathbf{W}^{(\mathrm{m})},\mathbf{Z},\tau^{(\mathrm{m})},\mathbf{b}^{(\mathrm{m})}\Big)\Big)\Big] + \text{const.}\end{aligned} \tag{A.77}$$

The first term of the distribution does not depend on variable $\mathbf{b}^{(\mathrm{m})}$, so the result is the same as in standard multilabel case. The second term of the distribution can be calculated equivalently as in Equation (A.54), having that

$$\begin{aligned}\ln\Big(p\Big(\mathbf{X}^{(\mathrm{m})}\,|\,\mathbf{W}^{(\mathrm{m})},\mathbf{Z},\tau^{(\mathrm{m})}\Big)\Big) &= \sum_{\mathrm{n}=1}^{\mathrm{N}}\ln\Big(\mathcal{N}\Big(\mathbf{z}_{\mathrm{n},:}\,\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}+\mathbf{b}^{(\mathrm{m})},\Big(\tau^{(\mathrm{m})}\Big)^{-1}I\Big)\Big) + \text{const} \\ &= -\frac{\tau^{(\mathrm{m})}}{2}\sum_{\mathrm{n}=1}^{\mathrm{N}}\Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}\,\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})^{\mathrm{T}}} - 2\Big(\mathbf{z}_{\mathrm{n},:}\,\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}+\mathbf{b}^{(\mathrm{m})}\Big)\,\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})^{\mathrm{T}}}\Big) + \text{const.}\end{aligned}$$

Therefore, joining both terms we have that:

$$\begin{aligned}\ln\Big(q^*\Big(\mathbf{X}^{(\mathrm{m})}\Big)\Big) = &\sum_{\mathrm{n}=1}^{\mathrm{N}}\Big(\Big(\mathbf{t}_{\mathrm{n},:}^{(\mathrm{m})} - \frac{1}{2} + \langle\tau^{(\mathrm{m})}\rangle\Big(\langle\mathbf{z}_{\mathrm{n},:}\rangle\langle\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\rangle\Big) + \langle\mathbf{b}^{(\mathrm{m})}\rangle\Big)\,\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})^{\mathrm{T}}} \\ &-\frac{1}{2}\,\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}\Big(\langle\tau^{(\mathrm{m})}\rangle I + 2\Lambda_{\xi_n}\Big)\,\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})^{\mathrm{T}}}\Big) + \text{const.}\end{aligned} \tag{A.78}$$

This way we obtain that the distribution is as follows:

$$q^*\Big(\mathbf{X}^{(\mathrm{m})}\Big) = \prod_{\mathrm{n}=1}^{\mathrm{N}}\Big(\mathcal{N}\Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}\,|\langle\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}\rangle,\Sigma_{\mathbf{X}^{(\mathrm{m})}}\Big)\Big), \tag{A.79}$$

where the variance is

$$\Sigma_{\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}}^{-1} = \langle\tau^{(\mathrm{m})}\rangle I_{\mathrm{D_m}} + 2\Lambda_{\xi_n}, \tag{A.80}$$

and the mean is

$$\langle\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}\rangle = \Big(\mathbf{t}_{\mathrm{n},:}^{(\mathrm{m})} - \frac{1}{2} + \langle\tau^{(\mathrm{m})}\rangle\Big(\langle\mathbf{z}_{\mathrm{n},:}\rangle\langle\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}}\rangle + \langle\mathbf{b}^{(\mathrm{m})}\rangle\Big)\Big)\Sigma_{\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}}. \tag{A.81}$$

A.4.5 Variational inference: Categorical view

In this particular case, the way the model is defined makes the inclusion of the bias term straightforward, as this term simply implies the redefinition of $\mathbf{y}_{\mathrm{n},:}^{(\mathrm{m})}$ as

$$\mathbf{y}_{\mathrm{n},:}^{(\mathrm{m})} \;=\; \mathbf{z}_{\mathrm{n},:}\,\mathbf{W}^{(\mathrm{m})^{\mathrm{T}}} + \mathbf{b}^{(\mathrm{m})}\,. \tag{A.82}$$

A.4.6 Update of the lower bound

By the inclusion of the bias term the lower bound needs to be updated to include the effect of the term. In particular, this will not affect in any manner the rest of the terms, as their dependence on the bias is implicitly included in the formulation. Therefore, the new terms of entropy are

$$\mathbb{E}_q\Big[\ln\Big(p\Big(\mathbf{b}^{(\mathrm{m})}\Big)\Big)\Big] \;=\; -\frac{\mathrm{D_m}}{2}\ln(2\pi) - \frac{1}{2}\langle\mathbf{b}^{(\mathrm{m})}\,\mathbf{b}^{(\mathrm{m})^{\mathrm{T}}}\rangle \tag{A.83}$$

$$\mathbb{E}_q\Big[\ln\Big(q\Big(\mathbf{b}^{(\mathrm{m})}\Big)\Big)\Big] \;=\; \frac{\mathrm{D_m}}{2}\ln(2\pi e) + \frac{1}{2}\ln|\Sigma_{\mathbf{b}^{(\mathrm{m})}}|. \tag{A.84}$$

Therefore, the updated lower bound is

$$\begin{aligned} L_q \;=\; & -\frac{1}{2}\mathrm{Tr}\{\langle\mathbf{Z}^{\mathrm{T}}\,\mathbf{Z}\rangle\} - \sum_{\mathrm{m}=1}^{\mathrm{M}}\left(\left(\frac{\mathrm{D_m}}{2}+\alpha_0-1\right)\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\Big(\ln\Big(b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}}\Big)\Big)\right) - \frac{1}{2}\sum_{\mathrm{m}=1}^{\mathrm{M}}\langle\mathbf{b}^{(\mathrm{m})}\,\mathbf{b}^{(\mathrm{m})^{\mathrm{T}}}\rangle \\ & - \sum_{\mathrm{m}=1}^{\mathrm{M}}\left(\left(\frac{\mathrm{D_m}}{2}+\alpha_0^{\tau}-1\right)(\ln(b_{\tau^{(\mathrm{m})}}))\right) \\ & - \frac{N}{2}\ln|\Sigma_{\mathbf{Z}}| - \sum_{\mathrm{m}=1}^{\mathrm{M}}\left(\frac{\mathrm{D_m}}{2}\ln|\Sigma_{\mathbf{W}^{(\mathrm{m})}}|\right) - \frac{1}{2}\sum_{\mathrm{m}=1}^{\mathrm{M}}\ln|\Sigma_{\mathbf{b}^{(\mathrm{m})}}| + \sum_{\mathrm{m}=1}^{\mathrm{M}}\left(\sum_{\mathrm{k}=1}^{\mathrm{K_c}}\Big(\ln\Big(b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}}\Big)\Big)\right) \\ & + \sum_{\mathrm{m}=1}^{\mathrm{M}}(\ln(b_{\tau^{(\mathrm{m})}}))\,. \end{aligned} \tag{A.85}$$

A.5 Predictive SSHIBA

Once the models are defined, they can be combined with input data to calculate the values of the different variables we are working with. This process is the training process and uses some of the available data to define the model and through iterations obtain the optimum values of the variables. Therefore, after this process is completed, one can use some new input data in some views to predict the output data of a particular view using this trained model.

To do so, we have to keep in mind that the only variable that depends on $\mathbf{x}_{*,:}^{(\mathrm{m})}$ is $\mathbf{z}_{*,:}$, as seen in section 2.2.2.3, where the $*$ denotes these are test samples used for the prediction. At this point, given the marginal Gaussian distribution for $\mathbf{Z}$ defined in Equation (2.74) and the conditional Gaussian distribution for $\mathbf{X}^{(\mathrm{m})}$ given $\mathbf{Z}$ defined in Equation (2.76) we can write the conditional distribution of $\mathbf{z}_{*,:}$ as

$$p\Big(\mathbf{z}_{*,:}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\mathrm{in}}\}}\Big) \;=\; \int p(\mathbf{z}_{*,:})p\Big(\mathbf{x}_{*,:}^{\{\mathcal{M}_{\mathrm{in}}\}}\,|\,\mathbf{z}_{*,:}\Big)d\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\mathrm{in}}\}}, \tag{A.86}$$

where $\mathcal{M}_{\text{in}}$ denotes the set of input views, while $\mathcal{M}_{\text{out}}$ denotes the set of output views. We can define the distribution as

$$p\Big(\mathbf{z}_{*,:}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}}\Big) \;=\; \mathcal{N}\big(\mathbf{z}_{*,:}\,|\langle\mathbf{z}_{*,:}\rangle, \Sigma_{\tilde{\mathbf{Z}}}\big), \tag{A.87}$$

where we can now identify terms considering the Gaussian properties of the marginal and conditional Gaussians. Doing that, we obtain

$$\Sigma_{\tilde{\mathbf{Z}}}^{-1} \;=\; I + \sum_{m=1}^{\mathcal{M}_{\text{in}}}\Big(\langle\tau^{(\text{m})}\rangle\langle\mathbf{W}^{(\text{m})^{\text{T}}}, \mathbf{W}^{(\text{m})}\rangle\Big),$$

where $\mathcal{M}_{\text{in}}$ over the sum denotes we are doing the summation over the known input views to predict the output view. The mean value of $\mathbf{z}_{*,:}$ can be calculated in a similar way, having

$$\langle\mathbf{z}_{*,:}\rangle \;=\; \sum_{m=1}^{\mathcal{M}_{\text{in}}}\Big(\langle\tau^{(\text{m})}\rangle\,\mathbf{x}_{*,:}^{(\text{m})}\langle\mathbf{W}^{(\text{m})}\rangle\Big)\Sigma_{\tilde{\mathbf{Z}}}.$$

We can now write the expression of the distribution of the output variable $\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{out}}\}}$ as

$$p\Big(\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{out}}\}}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}}\Big) \;=\; \int p\Big(\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{out}}\}}\,|\,\mathbf{z}_{*,:}\Big)p\Big(\mathbf{z}_{*,:}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}}\Big)d\,\mathbf{z}_{*,:}, \tag{A.88}$$

where $p\Big(\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{out}}\}}\,|\,\mathbf{z}_{*,:}\Big)$ was defined in Equation (2.76) and can be determined either considering a point estimate of $\mathbf{W}^{\{\mathcal{M}_{\text{out}}\}}$ or using Monte Carlo Integration. In this case, we are going to determine the distribution using just the mean. This way, knowing the distribution of both $p\Big(\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{out}}\}}\,|\,\mathbf{z}_{*,:}\Big)$ and $p\Big(\mathbf{z}_{*,:}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}}\Big)$, we can obtain the resulting distribution using Gaussian properties having that:

$$\begin{aligned} &p\Big(\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{out}}\}}\,|\,\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{in}}\}}\Big) \\ &= \mathcal{N}\Big(\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{out}}\}}\,|\langle\mathbf{z}_{*,:}\rangle\langle\mathbf{W}^{(\text{m}_{\text{out}})^{\text{T}}}\rangle, \langle\tau^{\{\mathcal{M}_{\text{out}}\}}\rangle^{-1}I + \langle\mathbf{W}^{\{\mathcal{M}_{\text{out}}\}}\rangle\Sigma_{\mathbf{z}_{*,:}}\langle\mathbf{W}^{(\text{m}_{\text{out}})^{\text{T}}}\rangle\Big) \end{aligned} \tag{A.89}$$

A.5.1 Predictive model for multilabel classification problems

The change of paradigm in this scenario relates to the relation between $\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{out}}\}}$ and $\mathbf{t}_{*,:}^{\{\mathcal{M}_{\text{out}}\}}$. In this aspect, the two variables that do not change between the standard BIBFA method and the sparse version are $\mathbf{z}_{*,:}$ and $\mathbf{x}_{*,:}^{\{\mathcal{M}_{\text{out}}\}}$, having that their calculation is carried out exactly as seen in the previous case. Hence, the calculation of the variable $\mathbf{t}_{*,:}^{\{\mathcal{M}_{\text{out}}\}}$ is the only new equation we have to add.

Therefore, we can start with Equation (A.89) which is obtained in the same way to calculate the distribution of $\mathbf{t}_{*,:}^{\{\mathcal{M}_{\text{out}}\}}$. We can start by writing its conditional probability as follows:

$$\begin{aligned} p\Big(\text{t}_{*,\text{d}}^{\{\mathcal{M}_{\text{out}}\}} = 1|\,\text{x}_{*,\text{d}}^{\{\mathcal{M}_{\text{in}}\}}\Big) \;&=\; \int p\Big(\text{t}_{\text{n,d}}^{\{\mathcal{M}_{\text{out}}\}} = 1|\,\text{x}_{*,\text{d}}^{\{\mathcal{M}_{\text{out}}\}}\Big)p\Big(\text{x}_{*,\text{d}}^{\{\mathcal{M}_{\text{out}}\}}\,|\,\text{x}_{*,\text{d}}^{\{\mathcal{M}_{\text{in}}\}}\Big)d\,\text{x}_{*,\text{d}}^{\{\mathcal{M}_{\text{out}}\}} \\ &=\; \int \sigma\Big(\text{x}_{*,\text{d}}^{\{\mathcal{M}_{\text{out}}\}}\Big)p\Big(\text{x}_{*,\text{d}}^{\{\mathcal{M}_{\text{out}}\}}\,|\,\text{x}_{*,\text{d}}^{\{\mathcal{M}_{\text{in}}\}}\Big)d\,\text{x}_{*,\text{d}}^{\{\mathcal{M}_{\text{out}}\}}, \end{aligned} \tag{A.90}$$

where we have calculated in Equation(A.89) the distribution of $p\Big(\text{x}_{*,\text{d}}^{\{\mathcal{M}_{\text{out}}\}}\,|\,\text{x}_{*,\text{d}}^{\{\mathcal{M}_{\text{in}}\}}\Big)$. As stated in Bishop (2006) the integral over $\text{x}_{*,\text{d}}^{\{\mathcal{M}_{\text{out}}\}}$ cannot be evaluated analytically, but we can obtain a good approximation. To do so we consider the similarity between the sigmoid function and the probit function. This similarity was found to be maximal when the probit function is re-scaled by a factor of $\lambda^2 = \pi/8$.

By using the probit function, we can know calculate analytically the result of the previous convolution, considering that:

$$\int \Phi\Big(\lambda\, \mathrm{x}_{*,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}}\Big)\mathcal{N}\Big(\mathrm{x}_{*,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}}\,|\langle \mathrm{x}_{*,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}}\rangle, \Sigma_{\mathbf{x}_{:,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}}}\Big)d\,\mathrm{x}_{*,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}} \;=\; \Phi\left(\frac{\langle \mathrm{x}_{*,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}}\rangle}{\Big(\lambda^{-2} + \Sigma_{\mathbf{x}_{:,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}}}\Big)^{1/2}}\right). \tag{A.91}$$

Therefore, given the approximation $\sigma\Big(\mathrm{x}_{*,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}}\Big) \simeq \Phi\Big(\lambda\,\mathrm{x}_{*,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}}\Big)$ we can apply it have the following results:

$$p\Big(\mathrm{t}_{*,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}} = 1|\,\mathrm{x}_{*,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{in}}\}}\Big) \;=\; \sigma\left(\frac{\langle \mathrm{x}_{*,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}}\rangle}{\Big(1 + \frac{\pi}{8}\Sigma_{\mathbf{x}_{:,\mathrm{d}}^{\{\mathcal{M}_{\mathrm{out}}\}}}\Big)^{1/2}}\right). \tag{A.92}$$

A.5.2 Categorical predictive model

In this case the predictive model needs to calculate the values of $\mathrm{t}_*^{(\mathrm{m})}$. In comparison with the model learning, in this case we do not know the distribution of $\mathrm{t}_*^{(\mathrm{m})}$, as it is not seen. Therefore we do not have a restriction to apply to the distribution of the variable $\mathbf{x}_{*,:}^{(\mathrm{m})}$. As we do not have restrictions, we can now consider that this variable has a standard Gaussian distribution, instead of the truncated one we considered before, and, thus, we now that its mean is given by

$$\langle \tilde{\mathbf{x}}_{\mathrm{n},:}^{(\mathrm{m})}\rangle \;=\; \langle \mathbf{z}_{*,:}\rangle\langle \mathbf{W}^{(\mathrm{m})}\rangle^{\mathrm{T}}. \tag{A.93}$$

This way the categorical variable $\mathrm{t}_*^{(\mathrm{m})}$ can be calculated as the argument that maximises $\langle \mathbf{x}_{*,:}^{(\mathrm{m})}\rangle$:

$$\mathrm{t}_*^{(\mathrm{m})} = i \qquad if \qquad \langle \mathrm{x}_{*,\mathrm{i}}^{(\mathrm{m})}\rangle = \max_{1\leq \mathrm{d}\leq \mathrm{D_m}}\Big\{\langle \mathrm{x}_{*,\mathrm{d}}^{(\mathrm{m})}\rangle\Big\}. \tag{A.94}$$

A.6 Semi-supervised SSHIBA

Equivalently, here we present the semi-supervised formulations associated to the different SSHIBA extensions.

A.6.1 Semi-supervised for real-data views

A.6.1.1 Generative model

Once the model has been defined for the standard supervised case, we can adapt it to work in a semi-supervised way if needed. To do so, we have to keep in mind that in this context we do not know some data from $\mathbf{X}^{(\mathrm{m})}$. These unknown data is going to be referred as $\tilde{\mathbf{X}}^{(\mathrm{m})}$. Therefore, we can now redefine the inference model for this new scheme

$$q(\Theta) \;=\; q\Big(\tilde{\mathbf{Z}}, \mathbf{Z}\Big) \prod_{m=1}^{\mathrm{M}} \Big(q\Big(\mathbf{W}^{(\mathrm{m})}\Big) q\Big(\boldsymbol{\alpha}^{(\mathrm{m})}\Big) q\Big(\tau^{(\mathrm{m})}\Big) q\Big(\tilde{\mathbf{X}}^{(\mathrm{m})}\Big)\Big) \tag{A.95}$$

A.6.1.2 Variational mean field

As we saw in previous sections, we now need to determine the q distributions for each variable in order to obtain the updating steps taken by the model for each variable. Regarding variable $\boldsymbol{\alpha}^{(\mathrm{m})}$, there is no change on its values, as the inclusion of the semi-supervised method affects $\mathbf{X}^{(\mathrm{m})}$ and $\mathbf{Z}$ but nothing else.

In the case of the variables $\mathbf{W}^{(\mathrm{m})}$, $\mathbf{Z}$ and $\tau^{(\mathrm{m})}$ although the equations of their distribution parameters do not change, the definition of the means included change, having

$$\begin{array}{ll} \langle \mathbf{X}^{(\mathrm{m})} \rangle = \mathbf{X}^{(\mathrm{m})} & \langle \tilde{\mathbf{X}}^{(\mathrm{m})} \rangle = \langle \tilde{\mathbf{X}}^{(\mathrm{m})} \rangle' \\ \langle \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})^2} \rangle = \mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m})^2} & \langle \tilde{\mathrm{x}}_{\mathrm{n,d}}^{(\mathrm{m})^2} \rangle = \Sigma'_{\tilde{\mathrm{x}}_{\mathrm{n,d}}^{(\mathrm{m})}} + \langle \tilde{\mathrm{x}}_{\mathrm{n,d}}^{(\mathrm{m})^2} \rangle', \end{array}$$

where $\langle \tilde{\mathrm{x}}_{\mathrm{n,d}}^{(\mathrm{m})^2} \rangle'$ and $\Sigma'_{\tilde{\mathrm{x}}_{\mathrm{n,d}}^{(\mathrm{m})}}$ are the mean and variance of the q distribution. The calculation of this distribution is equivalent to the ones done in the previous section, where if we consider Equation (A.4) we have

$$\begin{aligned} \ln\Big(q\Big(\tilde{\mathbf{X}}^{(\mathrm{m})}\Big)\Big) \;&=\; \mathbb{E}\left[\sum_{\mathrm{n}=1}^{\mathrm{N}} \ln\Big(p\Big(\tilde{\mathbf{X}}^{(\mathrm{m})} \,|\, \mathbf{W}^{(\mathrm{m})}, \mathbf{z}_{*,:}, \tau^{(\mathrm{m})}, \boldsymbol{\alpha}^{(\mathrm{m})}\Big)\Big)\right] + \mathrm{const} \\ &=\; -\frac{\tau^{(\mathrm{m})}}{2} \sum_{\mathrm{n}=1}^{\mathrm{N}} \Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})}\, \mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})\mathrm{T}} - 2\, \mathbf{x}_{\mathrm{n},:}^{(\mathrm{m})} \langle \mathbf{W}^{(\mathrm{m})} \rangle \langle \tilde{\mathbf{z}}_{\mathrm{n},:}^{\mathrm{T}} \rangle \Big) + \mathrm{const}. \end{aligned} \tag{A.96}$$

We can now compare this results with the normal distribution and identify terms in order to extract the values of the distribution's parameters

$$q\Big(\tilde{\mathbf{X}}^{(\mathrm{m})}\Big) \;=\; \prod_{\mathrm{n}=1}^{\mathrm{N}} \mathcal{N}\Big(\tilde{\mathbf{x}}_{\mathrm{n},:}^{(\mathrm{m})} \,|\langle \tilde{\mathbf{x}}_{\mathrm{n},:}^{(\mathrm{m})} \rangle, \Sigma_{\tilde{\mathbf{X}}^{(\mathrm{m})}}\Big). \tag{A.97}$$

Therefore, the covariance matrix of the distribution is

$$\Sigma_{\tilde{\mathbf{X}}^{(\mathrm{m})}} \;=\; \langle \tau^{(\mathrm{m})} \rangle I, \tag{A.98}$$

and, equivalently, the mean can be calculated as

$$\langle \tilde{\mathbf{X}}^{(\mathrm{m})} \rangle \;=\; \langle \mathbf{z}_{*,:} \rangle \langle \mathbf{W}^{(\mathrm{m})} \rangle^{T}, \tag{A.99}$$

where the mean value of $\tilde{\mathbf{Z}}$ can be calculated using the mean value of $\mathbf{Z}$ for BIBFA (see Table

2.3), knowing that $\Sigma_{\tilde{\mathbf{Z}}} = \Sigma_{\mathbf{Z}}$:

$$\begin{aligned}
\langle\tilde{\mathbf{Z}}\rangle &= \sum_{\mathrm{m}=1}^{\mathrm{M}} \Big(\langle\tau^{(\mathrm{m})}\rangle\, \tilde{\mathbf{X}}^{(\mathrm{m})} \langle\mathbf{W}^{(\mathrm{m})}\rangle \Sigma_{\tilde{\mathbf{Z}}} \Big) \\
&= \langle\tau^{(1)}\rangle\, \tilde{\mathbf{X}}^{\{\mathcal{M}_{\mathrm{in}}\}} \langle\mathbf{W}^{(1)}\rangle \Sigma_{\tilde{\mathbf{Z}}} + \langle\tau^{(\mathrm{m})}\rangle \langle\tilde{\mathbf{X}}^{(\mathrm{m})}\rangle \langle\mathbf{W}^{(\mathrm{m})}\rangle \Sigma_{\tilde{\mathbf{Z}}} \\
&= \langle\tau^{(1)}\rangle\, \tilde{\mathbf{X}}^{\{\mathcal{M}_{\mathrm{in}}\}} \langle\mathbf{W}^{(1)}\rangle \Sigma_{\tilde{\mathbf{Z}}} + \langle\tau^{(\mathrm{m})}\rangle \langle\mathbf{z}_{*,:}\rangle \langle\mathbf{W}^{(\mathrm{m})}\rangle^{T} \langle\mathbf{W}^{(\mathrm{m})}\rangle \Sigma_{\tilde{\mathbf{Z}}} \\
&= \Big(I - \langle\tau^{(\mathrm{m})}\rangle \langle\mathbf{W}^{(\mathrm{m})}\rangle^{T} \langle\mathbf{W}^{(\mathrm{m})}\rangle \Sigma_{\tilde{\mathbf{Z}}} \Big)^{-1} \langle\tau^{(1)}\rangle\, \tilde{\mathbf{X}}^{\{\mathcal{M}_{\mathrm{in}}\}} \langle\mathbf{W}^{(\mathrm{m})}\rangle \Sigma_{\tilde{\mathbf{Z}}}. \qquad \text{(A.100)}
\end{aligned}$$

A.6.1.3 Update of the lower bound

Regarding the lower bound, the definition of $\mathbf{X}^{(\mathrm{m})}$ we have done, which includes all the unknown values, allows us to have a more consistent definition of the lower bound. As happened calculating the update rules, the definition of the equations does not varies, although the matrices $\mathbf{X}^{(\mathrm{m})}$ we are using now are different in this semi-supervised context. Therefore, the only change needed to include in the previous lower bound equation is the inclusion of $\mathbf{X}^{(\mathrm{m})}$ as a new variable and, thus, its entropy needs to be added to the equation

$$\begin{aligned}
\mathbb{E}_q\Big[\ln\Big(q\Big(\tilde{\mathbf{X}}^{(\mathrm{m})}\Big)\Big)\Big] &= \sum_{n=N^*}^{N} \left(\frac{\mathrm{D}_2}{2} \ln(2\pi e) + \frac{1}{2} \ln|\Sigma_{\tilde{\mathbf{X}}^{(\mathrm{m})}}| \right) \\
&= \frac{(N - N^*)\,\mathrm{D}_2}{2} \ln(2\pi e) + \frac{N - N^*}{2} \ln|\Sigma_{\tilde{\mathbf{X}}^{(\mathrm{m})}}|. \qquad \text{(A.101)}
\end{aligned}$$

Adding this new term to Equation (A.26) and discarding the terms that are constant, we have that the lower bound is updated following

$$\begin{aligned}
L_q &= -\frac{1}{2}\,\mathrm{Tr}\{\langle\mathbf{Z}^{\mathrm{T}}\,\mathbf{Z}\rangle\} \\
&+ \sum_{\mathrm{m}=1}^{\mathrm{M}} \left(\left(\frac{\mathrm{D_m}}{2} + \alpha_0 - 1 \right) \sum_{\mathrm{k}=1}^{\mathrm{K_c}} \Big(-\ln\Big(b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}} \Big)\Big) \right) \\
&+ \sum_{\mathrm{m}=1}^{\mathrm{M}} \left(\left(\frac{\mathrm{D_m}}{2} + \alpha_0^{\tau} - 1 \right) (-\ln(b_{\tau^{(\mathrm{m})}})) \right) \\
&- \frac{N}{2} \ln|\Sigma_{\mathbf{Z}}| - \sum_{\mathrm{m}=1}^{\mathrm{M}} \left(\frac{\mathrm{D_m}}{2} \ln|\Sigma_{\mathbf{W}^{(\mathrm{m})}}| \right) + \sum_{\mathrm{m}=1}^{\mathrm{M}} \left(\sum_{\mathrm{k}=1}^{\mathrm{K_c}} \Big(\ln\Big(b_{\alpha_{\mathrm{k}}^{(\mathrm{m})}} \Big)\Big) \right) \\
&+ \sum_{\mathrm{m}=1}^{\mathrm{M}} (\ln(b_{\tau^{(\mathrm{m})}})) - \frac{\mathrm{N} - \mathrm{N}^*}{2} \ln|\Sigma_{\tilde{\mathbf{X}}^{(\mathrm{m})}}|. \qquad \text{(A.102)}
\end{aligned}$$

A.6.2 Semi-supervised for multi-dimensional binary views

A.6.2.1 Generative model

Similarly to what we obtained for the standard version of BIBFA, the multilabel version needs some considerations regarding the variables are semi-supervised. In this case, our attention needs to focus on the variable $\mathbf{T}^{(\mathrm{m})}$ which, as happened in the previous semi-supervised version, needs to be defined for the different situations we can encounter:

$$\begin{aligned} \langle \mathbf{T}^{(\mathrm{m_t})} \rangle &= \mathbf{T}^{(\mathrm{m_t})} & \langle \tilde{\mathbf{T}}^{(\mathrm{m_t})} \rangle &= \langle \tilde{\mathbf{T}}^{(\mathrm{m_t})} \rangle' \\ \langle \mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m_t})^2} \rangle &= \mathrm{t}_{\mathrm{n,d}}^{(\mathrm{m_t})^2} & \langle \tilde{\mathrm{t}}_{\mathrm{n,d}}^{(\mathrm{m_t})^2} \rangle &= \Sigma'_{\tilde{\mathrm{t}}_{\mathrm{n,d}}^{(\mathrm{m_t})}} + \langle \tilde{\mathrm{t}}_{\mathrm{n,d}}^{(\mathrm{m_t})^2} \rangle'. \end{aligned}$$

Once this previous definition is stated, we can obtain the model parameters distribution as

$$q(\Theta) \;=\; q\Big(\tilde{\mathbf{Z}}, \mathbf{Z}\Big) \prod_{m=1}^{\mathrm{M}} \Big(q\Big(\mathbf{W}^{(\mathrm{m})}\Big) q\Big(\boldsymbol{\alpha}^{(\mathrm{m})}\Big) q\Big(\tau^{(\mathrm{m})}\Big) q\Big(\mathbf{X}^{(\mathrm{m})}, \tilde{\mathbf{X}}^{(\mathrm{m})}\Big)\Big) \prod_{m_t \in \mathcal{M}_t} q\Big(\tilde{\mathbf{T}}^{(\mathrm{m_t})}\Big). \tag{A.103}$$

At this point we would need to make use of the variational mean field to calculate these distributions.

A.6.2.2 Variational mean field

Considering the graph presented in Figure 4.3, we see the values of the distribution of $\tilde{\mathbf{Z}}$, $\mathbf{W}^{(\mathrm{m})}$, $\boldsymbol{\alpha}^{(\mathrm{m})}$, $\tau^{(\mathrm{m})}$ are the same as seen in the standard version. The only change that has to be considered is that now the calculation of the mean of $\tilde{\mathbf{X}}^{(\mathrm{m_t})}$ depends on $\tilde{\mathbf{T}}^{(\mathrm{m_t})}$.

We can now calculate the distribution of $\tilde{\mathbf{T}}^{(\mathrm{m_t})}$ in order to use it to calculate the one of variable $\tilde{\mathbf{X}}^{(\mathrm{m})}$. As we now the conditional probability, we can write the distribution of $\tilde{\mathbf{T}}^{(\mathrm{m_t})}$ as

$$\begin{aligned} &\ln\Big(q\Big(\tilde{\mathbf{T}}^{(\mathrm{m_t})}\Big)\Big) = \mathbb{E}\Big[\ln\Big(p\Big(\tilde{\mathbf{T}}^{(\mathrm{m_t})} \,|\, \tilde{\mathbf{X}}^{(\mathrm{m_t})}\Big)\Big)\Big] = \mathbb{E}\Big[\ln\Big(h\Big(\tilde{\mathbf{X}}^{(\mathrm{m_t})}, \boldsymbol{\xi}^{(\mathrm{m_t})}\Big)\Big)\Big] \\ &= \mathbb{E}\Bigg[\sum_{n=N^*}^{N} \sum_{\mathrm{d}=1}^{\mathrm{D_t}} \Big(\ln\Big(\sigma\Big(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\Big)\Big) + \tilde{\mathrm{x}}_{\mathrm{n,d}}^{(\mathrm{m_t})}\, \tilde{\mathrm{t}}_{\mathrm{n,d}}^{(\mathrm{m_t})} - \frac{1}{2}\Big(\tilde{\mathrm{x}}_{\mathrm{n,d}}^{(\mathrm{m_t})} + \xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\Big) - \lambda\Big(\xi_{\mathrm{n,d}}^{(\mathrm{m_t})}\Big)\Big(\tilde{\mathrm{x}}_{\mathrm{n,d}}^{(\mathrm{m_t})^2} - \xi_{\mathrm{n,d}}^{(\mathrm{m_t})^2}\Big)\Big)\Bigg] \\ &= \;\tilde{\mathbf{T}}^{(\mathrm{m})} \langle \tilde{\mathbf{X}}^{(\mathrm{m})} \rangle + \text{const.} \end{aligned} \tag{A.104}$$

We will have two different distributions depending on the two values variable $\tilde{\mathbf{T}}^{(\mathrm{m_t})}$ can take

$$\begin{aligned} q\Big(\tilde{\mathbf{T}}^{(\mathrm{m_t})} = 1\Big) &= \frac{e^{\langle \tilde{\mathbf{X}}^{(\mathrm{m_t})} \rangle}}{1 + e^{\langle \tilde{\mathbf{X}}^{(\mathrm{m_t})} \rangle}} = \sigma\Big(\langle \tilde{\mathbf{X}}^{(\mathrm{m_t})} \rangle\Big) \\ q\Big(\tilde{\mathbf{T}}^{(\mathrm{m_t})} = 0\Big) &= \frac{1}{1 + e^{\langle \tilde{\mathbf{X}}^{(\mathrm{m_t})} \rangle}}, \end{aligned} \tag{A.105}$$

where the denominator has been determined so that the sum of both probabilities is 1.Once these two values are calculated, we can calculate the variance of this variable as

$$\Sigma_{\tilde{\mathbf{T}}^{(\mathrm{m_t})}} \;=\; q\Big(\tilde{\mathbf{T}}^{(\mathrm{m_t})} = 1\Big) * q\Big(\tilde{\mathbf{T}}^{(\mathrm{m_t})} = 0\Big) = \frac{e^{\langle \tilde{\mathbf{X}}^{(\mathrm{m_t})} \rangle}}{\Big(1 + e^{\langle \tilde{\mathbf{X}}^{(\mathrm{m_t})} \rangle}\Big)^2}, \tag{A.106}$$

and the mean value of a binary random variable can be calculated as

$$\langle \tilde{\mathbf{T}}^{(\mathrm{m_t})} \rangle \;=\; 1 * q\Big(\tilde{\mathbf{T}}^{(\mathrm{m_t})} = 1\Big) + 0 * q\Big(\tilde{\mathbf{T}}^{(\mathrm{m_t})} = 0\Big) = \sigma\Big(\langle \tilde{\mathbf{X}}^{(\mathrm{m_t})} \rangle\Big). \tag{A.107}$$

Therefore, we can use this result as a semi-supervised estimation of the output variable $\tilde{\mathbf{T}}^{(\mathrm{m_t})}$. As this variable is only used in the distribution of $\mathbf{X}^{(\mathrm{m})}$, this mean value will only affect it. Meaning that, in the case when we are considering the data that is not seen, $\tilde{\mathbf{T}}^{(\mathrm{m_t})}$, we will use the mean value.

A.6.2.3 Update of the lower bound

The definition of $\mathbf{T}^{(\mathrm{m_t})}$, which includes all the unknown values, allows us to have a more consistent definition of the lower bound. As happened calculating the update rules, the definition of the equations does not varies, although the matrices $\mathbf{X}^{(\mathrm{m_t})}$ we are using now are different in this semi-supervised context. Therefore, the only change needed to include in the previous lower bound equation is the entropy of $\mathbf{X}^{(\mathrm{m_t})}$

$$\mathbb{E}_q\Big[\ln\Big(q\Big(\mathbf{T}^{(\mathrm{m_t})}\Big)\Big)\Big] \;=\; \frac{(\mathrm{N}-\mathrm{N}^*)\mathrm{D_t}}{2}\ln(2\pi e) + \frac{(\mathrm{N}-\mathrm{N}^*)}{2}\ln|\Sigma_{\tilde{\mathbf{T}}^{(2)}}|. \tag{A.108}$$

A.6.3 Semi-supervised for categorical views

A.6.3.1 Generative model

Following the models calculated for the previous versions, we can now calculate the semi-supervised version of the categorical model. In particular, the definition of the different cases of variable $\mathbf{T}^{(2)}$, whether it is observed or not, is the same as we had for the multilabel model.

At this point, we can see that the posterior distribution in this case can be calculated as

$$q(\Theta) \;=\; q\Big(\tilde{\mathbf{Z}},\mathbf{Z}\Big)\prod_{m=1}^{\mathrm{M}}\Big(q\Big(\mathbf{W}^{(\mathrm{m})}\Big)q\Big(\boldsymbol{\alpha}^{(\mathrm{m})}\Big)q\Big(\tau^{(\mathrm{m})}\Big)q\Big(\mathbf{X}^{(\mathrm{m})},\tilde{\mathbf{X}}^{(\mathrm{m})}\Big)\Big)q\Big(\tilde{\mathbf{t}}^{(\mathrm{m_t})}\Big). \tag{A.109}$$

A.6.3.2 Variational mean field

To calculate the term $q\big(\mathbf{X}^{(\mathrm{m})}\big)$. This one will remain the same if the data is observed but will need to be calculated for the case in which the data is unknown

$$\begin{aligned}\ln\Big(q\Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\Big)\Big) \;&=\; \mathbb{E}\Big[\ln\Big(p\Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\,|\,\mathbf{W}^{(\mathrm{m_t})},\mathbf{z}_{\mathrm{n},:},\boldsymbol{\alpha}^{(\mathrm{m_t})},1\Big)\Big)\Big]\\ &\quad+\mathbb{E}\left[\sum_{\mathrm{d}=1}^{\mathrm{D_t}}\Big(\ln\Big(p\Big(\tilde{\mathrm{t}}_{\mathrm{n}}^{(\mathrm{m_t})}=\mathrm{d}|\,\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\Big)\Big)q\Big(\tilde{\mathrm{t}}_{\mathrm{n}}^{(\mathrm{m_t})}=\mathrm{d}\Big)\Big)\right]\end{aligned} \tag{A.110}$$

Keeping this in mind, we determine that the distribution can be written as

$$q\Big(\mathbf{x}_{\mathrm{n},:}^{(\mathrm{m_t})}\Big) \propto \sum_{\mathrm{d}=1}^{\mathrm{D_t}}\Big(\mathcal{N}\Big(\tilde{\mathbf{y}}_{\mathrm{n},:}^{(\mathrm{m_t})},I\Big)\delta\Big(\mathrm{x}_{\mathrm{n,d}}^{(\mathrm{m_t})}\,\mathrm{x}_{\mathrm{n,j}}^{(\mathrm{m_t})}\,\forall d\neq j\Big)q\Big(\tilde{\mathrm{t}}_{\mathrm{n}}^{(\mathrm{m_t})}=\mathrm{d}\Big)\Big) \tag{A.111}$$

having that the distribution of the variable $\mathbf{X}^{(\mathrm{m})}$ is calculated as a weighted sum of truncated Gaussians, which can be seen as the distribution above weighted by the distribution of the different classes. Hence, assuming again that $\mathrm{t}_{\mathrm{n}}^{(\mathrm{m_t})}=i$, then

$$\langle \mathrm{x}_{\mathrm{n,i}}^{(\mathrm{m_t})}\rangle \;=\; \langle \tilde{\mathrm{y}}_{\mathrm{n,i}}^{(\mathrm{m_t})}\rangle + \sum_{j\neq i}\Big(\langle \tilde{\mathrm{y}}_{\mathrm{n,j}}^{(\mathrm{m_t})}\rangle - \langle \tilde{\mathrm{x}}_{\mathrm{n,j}}^{(\mathrm{m_t})}\rangle\Big) \tag{A.112}$$

$$\langle \mathrm{x}_{\mathrm{n,j}}^{(\mathrm{m_t})}\rangle \;=\; \langle \tilde{\mathrm{y}}_{\mathrm{n,j}}^{(\mathrm{m_t})}\rangle - \frac{1}{\xi_n}\mathbb{E}_{p(u)}\left[\mathcal{N}_u\Big(\langle \tilde{\mathrm{y}}_{\mathrm{n,j}}^{(\mathrm{m_t})}\rangle - \langle \tilde{\mathrm{y}}_{\mathrm{n,i}}^{(\mathrm{m_t})}\rangle,1\Big)\prod_{k\neq i\neq j}\Big(\Phi\Big(u+\langle \tilde{\mathrm{y}}_{\mathrm{n,i}}^{(\mathrm{m_t})}\rangle - \langle \tilde{\mathrm{y}}_{\mathrm{n,k}}^{(\mathrm{m_t})}\rangle\Big)\Big)\right]. \tag{A.113}$$

Appendix B

ADNI brain regions

To fully exploit the multi-view configuration of SSHIBA, we decided to divide the fMRI voxels into different brain areas to combine them in different views. Therefore, we decided to combine 3 different Harvard-Oxford atlases (Makris et al., 2006): Cortical, subcortical and cerebellum. The cortical atlas is available at https://identifiers.org/neurovault.image:1702, the subcortical atlas at https://identifiers.org/neurovault.image:1697 and the Cerebellar atlas in MNI152 after FNIRT is available at http://www.diedrichsenlab.org/imaging/propatlas.htm. Table B.1 has a description of the subcortical and cerebellum regions and Table B.2 of the cortical regions. El veloz murciélago hindú comía

Abbreviation	Description
'CWM'	Cerebral White Matter
'Cor'	Cerebral Cortex
'LV'	Lateral Ventrical
'Tha'	Thalamus
'Cau'	Caudate
'Put'	Putamen
'P'	Pallidum
'BS'	Brain-Stem
'HC'	Hippocampus
'AM'	Amygdala
'Acc'	Accumbens
'CEI-IV'	Cerebellum regions
'CEV'	
'CEVI'	
'CECrI'	
'CECrII'	
'CEVIIb'	
'CEVIIIa'	
'CEVIIIb'	
'CEIX'	
'CEX'	

Table B.1: Description of the Harvard-Oxford subcortical and cerebellum regions abbreviations. Note that we combined the left and right regions, as well as the vermis in the case of the cerebellum.

Abbreviation	Description
'FP'	Frontal Pole
'Ins'	Insular Cortex
'SFG'	Superior Frontal Gyrus
'MFG'	Middle Frontal Gyrus
'IFGpt'	Inferior Frontal Gyrus pars triangularis
'IFGpo'	Inferior Frontal Gyrus pars opercularis
'PrG'	Precentral Gyrus
'TP'	Temporal Pole
'STGa'	Superior Temporal Gyrus anterior division
'STGp'	Superior Temporal Gyrus posterior division
'MTGa'	Middle Temporal Gyrus anterior division
'MTGp'	Middle Temporal Gyrus posterior division
'MTGtp'	Middle Temporal Gyrus temporooccipital part
'ITGa'	Inferior Temporal Gyrus anterior division
'ITGp'	Inferior Temporal Gyrus posterior division
'ITGtp'	Inferior Temporal Gyrus temporooccipital part
'PoG'	Postcentral Gyrus
'SPL'	Superior Parietal Lobule
'SmGa'	Supramarginal Gyrus anterior division
'SmGp'	Supramarginal Gyrus posterior division
'AG'	Angular Gyrus
'LOCs'	Lateral Occipital Cortex superior division
'LOCi'	Lateral Occipital Cortex inferior division
'IcC'	Intracalcarine Cortex
'FMC'	Frontal Medial Cortex
'SMC'	Juxtapositional Lobule Cortex (formerly Supplementary Motor Cortex)
'ScC'	Subcallosal Cortex
'PcG'	Paracingulate Gyrus
'CGa'	Cingulate Gyrus anterior division
'CGp'	Cingulate Gyrus posterior division
'PcC'	Precuneous Cortex
'CC'	Cuneal Cortex
'FOC'	Frontal Orbital Cortex
'PhGa'	Parahippocampal Gyrus anterior division
'PaGp'	Parahippocampal Gyrus posterior division
'LG'	Lingual Gyrus
'TFCa'	Temporal Fusiform Cortex anterior division
'TFCp'	Temporal Fusiform Cortex posterior division
'TOF'	Temporal Occipital Fusiform Cortex
'OFG'	Occipital Fusiform Gyrus
'FOpC'	Frontal Operculum Cortex
'COpC'	Central Opercular Cortex
'POpC'	Parietal Operculum Cortex
'PP'	Planum Polare
'H1/H2'	Heschl's Gyrus (includes H1 and H2)
'PT'	Planum Temporale
'SccC'	Supracalcarine Cortex
'OcP'	Occipital Pole

Table B.2: Description of the Harvard-Oxford cortical regions abbreviations.