Master of Science
Double Degree in *Communications and Computer Networks Engineering*
and *Data Science and Engineering*

Bayesian latent variable model for the analysis of the progression of Alzheimer's disease

Andrea Senacheribbe

Supervisors

Marco Lorenzi
Université Côte d'Azur, Epione Research Group - Inria Sophia Antipolis

Irene Balelli
Université Côte d'Azur, Epione Research Group - Inria Sophia Antipolis

Monica Visintin
Department of Electronics and Telecommunications - Politecnico di Torino

Maria A. Zuluaga
Data Science Department - EURECOM

Master thesis prepared at
Inria Sophia Antipolis, France

March-April, 2021

Abstract

Alzheimer's disease (AD) is an incurable neurodegenerative disorder which affects neurons, reducing their function and causing their death. AD is the most common form of dementia and it manifests with memory impairment and cognitive loss.
Data science can play an important role in enhancing our understanding of this disease and it can help to characterise the pathological evolution of the biomedical parameters in AD patients.
We propose here the latent slope-intercept model, a Bayesian latent variable model for longitudinal data analysis, inspired by the Probabilistic Principal Component Analysis. The model was derived analytically, implemented in Python and then applied to clinical scores and brain imaging data from Alzheimer's patients.
We showed that we are able to characterise the intrinsic variability of the data in the latent space, where the separation between healthy individuals and patients is enhanced. Moreover, we were able to interpret the effect of AD on the considered biomarkers and on their rate of variation, obtaining results consistent with the known pathophysiology of the disease.
We finally present two possible extensions of the model, to multi-centric data with a federated learning scheme, and to a more general modelling of the global disease progression.

Résumé

La maladie d'Alzheimer (MA) est une maladie neurodégénérative incurable qui affecte les neurones, réduisant leur fonction et entraînant leur mort. La MA est la forme de démence la plus courante et se manifeste par des troubles de la mémoire et des pertes cognitives.
La science des données peut jouer un rôle important pour améliorer notre compréhension de cette maladie et peut aider à caractériser l'évolution pathologique des paramètres biomédicaux chez les patients atteints de la MA.
Nous proposons ici le latent slope-intercept model, un modèle bayésien à variables latentes pour l'analyse des données longitudinales, inspiré de le Probabilistic Principal Component Analysis. Le modèle a été dérivé analytiquement, implémenté en Python puis appliqué aux scores cliniques et aux données d'imagerie cérébrale des patients atteints de la maladie d'Alzheimer.
Nous avons montré que nous sommes capables de caractériser la variabilité intrinsèque des données dans l'espace latent, où la séparation entre les individus en bonne santé et les patients est renforcée. De plus, nous avons pu interpréter l'effet de la MA sur les biomarqueurs considérés et sur leur taux de variation, obtenant des résultats conformes à la physiopathologie connue de la maladie.
Nous présentons enfin deux extensions possibles du modèle, à des données multicentriques avec un schéma d'apprentissage fédéré, et à une modélisation plus générale de la progression globale de la maladie.

Acknowledgements

This thesis is the result of my 6 months internship at Inria. I would like to thank all the colleagues and friends that made this stage special, in particular, Etrit Haxholli and Yann Fraboni for our walks that produced enriching (and never-ending) discussions and Santiago Silva for his super friendly company and his Spanish courses.

I'm extremely grateful to my tutors Marco Lorenzi and Irene Balelli. You were always available with invaluable advices and you profoundly believed in my work and abilities, engaging me in several activities during the internship. I also wish to thank my academic tutors Monica Visintin and Maria Zuluaga, for their help and support before and during the internship.

A special thank goes to Maria Luisa Doglio and Politecnico di Torino for awarding me with the scholarship in memory of Concetto Arena for the best thesis proposal in my field of studies.

This thesis is the part of a longer path, my studies toward my master degree, which started in Turin, Italy and ended in Nice, France. It was a great pleasure to share this path with my friends and classmates in both universities. You are too many to name you all!
I would also like to acknowledge the European Commission and Politecnico di Torino for financing my studies in France as part of the ERASMUS+ double degree programme.

Finally, but not less important, I cannot begin to express my thanks to all my friends Andrea Boido, Andrea Gerbi, Davide Valente, Federico Natale, Gabriele Ponzio, Letizia Bergamasco, Luca Rabezzana, Marco Eterno, Matteo Alasio, Morena Porzio, Stefano Panaro and to my family, my grandmas, my mum and dad. You were always present in the good and in the difficult moments, and you always supported me.

Thank you all!

Ringraziamenti

Questa tesi è il risultato dei miei 6 mesi di tirocinio a Inria. Vorrei ringraziare tutti i colleghi e gli amici che hanno reso questo stage speciale, in particolare, Etrit Haxholli e Yann Fraboni per le nostre passeggiate che hanno prodotto discussioni interessanti (e senza fine) e Santiago Silva per la sua compagnia super amichevole e i suoi corsi di spagnolo.

Sono estremamente grato ai miei tutor Marco Lorenzi e Irene Balelli. Siete sempre stati disponibili con preziosi consigli e avete creduto profondamente nel mio lavoro e nelle mie capacità, coinvolgendomi in diverse attività durante lo stage. Desidero anche ringraziare i miei tutor accademici Monica Visintin e Maria Zuluaga, per il loro aiuto e supporto prima e durante il tirocinio.

Un ringraziamento speciale è rivolto a Maria Luisa Doglio e al Politecnico di Torino per avermi assegnato la borsa di studio in memoria di Concetto Arena per la migliore proposta di tesi nel mio ambito di studi.

Questa tesi è parte di un percorso più lungo, i miei studi per la laurea magistrale, iniziati a Torino, Italia e terminati a Nizza, in Francia. È stato un grande piacere condividere questo percorso con i miei amici e compagni di corso, in entrambe le università. Siete troppi per nominarvi tutti!
Vorrei anche ringraziare la Commissione Europea e il Politecnico di Torino per aver finanziato i miei studi in Francia nell'ambito del programma di doppia laurea ERASMUS+.

Infine, ma non meno importante, non posso ringraziare abbastanza tutti i miei amici Andrea Boido, Andrea Gerbi, Davide Valente, Federico Natale, Gabriele Ponzio, Letizia Bergamasco, Luca Rabezzana, Marco Eterno, Matteo Alasio, Morena Porzio, Stefano Panaro e la mia famiglia, le mie nonne, mia mamma e mio papà. Siete sempre stati presenti nei momenti belli e in quelli difficili e mi avete sempre sostenuto e supportato.

Grazie a tutti!

Contents

Chapter 1

Introduction

In the last decades, Alzheimer's disease (AD) has been gaining attention as a public health priority. AD is an incurable neurodegenerative disease and the most common form of dementia. With dementia, we refer to a set of symptoms affecting especially older people which includes memory loss, impairment of cognitive functions and impossibility to perform everyday tasks, causing disability and dependency on others [1].
According to the World Health Organization, in 2015, around 47 million people in the world were affected by dementia and this figure is expected to increase to 132 million by 2050 [1]. It is clear then that AD has a very high human, social and economic cost for the patient, for his family and for the entire society, and this cost is expected to raise [1].

AD affects brain cells, reducing their function and causing their death. It is progressive disease because, in an initial phase, the symptoms are small and imperceptible but they can rapidly worsen over time. For this reason, AD is usually detected in its late stages.
Even though there are currently no treatments available to cure or reverse the progression of AD, early detection can provide several benefits to the patients and their caregivers: (a) a better preparedness of the patient and a prioritisation of an healthy lifestyle to preserve his/her cognitive abilities, (b) early access to palliative treatments that can reduce the effect of the disease, (c) better plans for the future and cost savings and (d) possibility to participate to clinical trials [2].
It becomes therefore of great importance to develop methods able to detect AD in its early stages, before the appearance of symptoms.

In the last decades, healthcare facilities are collecting an increasing amount of medical data from AD affected patients, including clinical scores, biological samples and imaging data. The analysis of these large collections of data

can be of great importance for a better understanding of the pathophysiology of Alzheimer's disease. In particular, it has been shown that the dynamics of several biomarkers can inform on the development of the disease in a presymptomatic phase [3]. These dynamics can be used to model a global progression of AD [4]. The evolution of the parameters of one subject can then be compared to the global evolution for diagnostic and monitoring purposes.

In this thesis, we propose the latent slope-intercept (LSI) model, a latent variable model for the study of longitudinal data, i.e. data varying with time. This model is designed to be applied to longitudinal observational studies of Alzheimer's patients.
LSI assumes a linear time evolution of the biomarkers of a subject and aims to project this evolution on a latent space, which is new hidden representation of the original observable data. In particular, we capture from the longitudinal data an intercept term, which is the baseline value of different biological parameters, and a slope term, which is the rate of variation of the biomarkers. Since healthy individual and sick patients shows different progressions for their biological parameters, we aim to model this variability in the latent representation of the data and we expect to see the points coming from AD patients distributed in a different portion of the latent space with respect to the cognitive normal subjects. This difference can be exploited to classify the patients in the two groups of diagnosis. Moreover, the model is a generative one and this gives the possibility to simulate a sick or healthy patient and observe the difference in evolution of their biomarkers.

This thesis is structured as follows. In chapter 2 we present the state of the art, focusing in particular on latent variable models and their application to the study of AD. Chapter 3 contains theoretical description of the LSI model. In chapter 4 the LSI is applied to synthetic data (generated exploiting the generative capabilities of the model itself) and to real longitudinal data coming from cognitive normal individuals and AD affected patients, for which we interpret and discuss the clinical validity of the results. In chapter 5, we present an extension of the LSI model to multi-centric studies where the data is securely stored in different centres and the training is performed using a federated learning scheme.
In chapter 6, we show another generalisation and extension of the model, in order to estimate a global evolution of the disease from the individual progression captured by LSI. Finally, in chapter 7 we draw some conclusion on the model, the obtained results and the presented extensions and we provide some possible future works.

Chapter 2

State of the art

The LSI model belongs to the category of latent variable models. Here we present more in details what latent variable models are and how they are applied to the study of Alzheimer's disease in the literature.

2.1 Latent variables models

Latent variables models are based on the assumption of the existence of latent variables, hidden representations of the data that cannot be directly measured. These latent variables should be inferred from the observable ones, so to they represent a low dimensional space in which one can analyse more easily the data.
High dimensional datasets are very common in the medical domain, for instance when we consider medical images or genetic data. In this case, a lower dimensional representation can be sufficient to capture the relevant variability of the data and the relation among features and it can be much more tractable than the original high dimensional dataset.
In machine learning, this concept is also called dimensionality reduction.

Latent variables models are typically generative models, i.e. they are able to generate likely realisation of the data. Indeed, by randomly sampling in the latent space we can create new data with similar characteristics to the one used to train the model (coming from the same data distribution).
A classical example of generative models are the generative adversarial networks (GAN) [5], which became very popular for their ability to generate photo-realistic fake human faces [6]. Another common example of generative model is the variational autoencoder (VAE) [7]. VAE can encode an input sample into a Gaussian distribution on the latent space (encoding) and then reconstruct the sample from the latent space (decoding). This method is using

neural networks for the encoding and decoding operation, and it can therefore model non-linear transformations.

To sum up, there exists different techniques that we can classify as latent variable models: some of them are linear and easier to train (like PCA), others are using non-linear transformations and are more complex (GANs and VAE), but their principles are similar.
We will now look into more details of the Principal Component Analysis method and its probabilistic extension, the Probabilistic Principal Component Analysis (PPCA), as they will be the starting point for formulating our model in chapter 3.

2.1.1 Principal Component Analysis

Principal Component Analysis (PCA) [8] is a technique to project some d-dimensional data on a new latent space of dimensions $q \leq d$, identified by q orthonormal axes (also called principal axes), along which the variance of the projected data is maximised.

Given a set y_n of N data points of dimension d, the aim of PCA is to find a transformation W that project the observable data y_n to a compact representation in the latent space x_n of dimension $q \leq d$ as follows:

$$x_n = W^T(y_n - \mu_y) \tag{2.1}$$

where $\mu_y = \frac{1}{N}\sum_n y_n$ is the sample mean of all points.
To find W we concatenate, along the columns, the q principal eigenvectors associated to the largest eigenvalues of the sample covariance matrix S given by:

$$S = \frac{1}{N}\sum_n (y_n - \mu_y)(y_n - \mu_y)^T \tag{2.2}$$

Given the latent representation, we can reconstruct the original data by applying the following inverse transformation:

$$y_n = W x_n + \mu_y \tag{2.3}$$

It can be proven that finding the transformation W that maximises the variance in the projection is equivalent to finding the transformation that minimises the squared reconstruction error, i.e. $\sum_n ||y_n - \hat{y_n}||^2$ where $\hat{y_n}$ is obtained by projecting the original data into the latent space using eq. (2.1) and reconstructing it with eq. (2.3).

In fig. 2.1 we report an example of PCA, where we compute and plot the two principal components (scaled by the squared root of the associated eigenvalue) using 1000 data points sampled from the following 2 dimensional multivariate Gaussian distribution:

$$y_n \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}4 & 2\\2 & 2\end{bmatrix}\right) \tag{2.4}$$

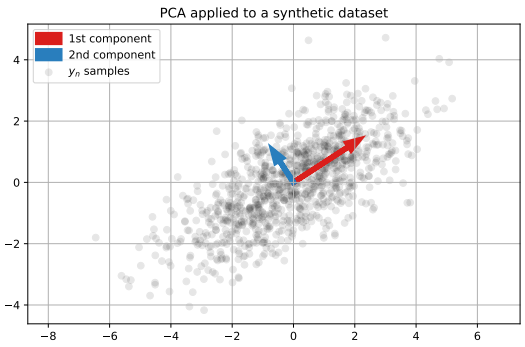


Figure 2.1 – Example of PCA applied on a synthetic dataset. We can clearly see that the first principal component is aligned in the direction of maximum variance of the Gaussian. The second principal component is perpendicular to the first.

2.1.2 Probabilistic Principal Component Analysis

Probabilistic Principal Component Analysis (PPCA) is a probabilistic extension of principal component analysis proposed by Tipping and Bishop in [9]. PPCA is based on a Bayesian probabilistic framework which gives several advantages over the standard PCA. First of all, it estimates a probability distribution of the data that can be used to generate new samples. Secondly, it allows to estimate the uncertainty of a data. This is especially relevant in medical application, where we are interested in both the result a model provides and its uncertainty. Moreover, we can use Bayesian model comparison, to automatically compare different models and their generalisation. And finally, we can use the inferred distribution to compensate for missing data in the projection.

Let y_n be the d-dimensional observable data and x_n the q-dimensional latent representation, PPCA is formally defined by the following generative model:

$$y_n = Wx_n + \mu_y + \epsilon \tag{2.5}$$

where W is the projection matrix, μ_y is the sample mean of the y_n and $\epsilon \sim \mathcal{N}(0,\, \sigma^2 \mathbb{I}_d)$ is a Gaussian noise.

From eq. (2.5), we can derive a likelihood function of the data given the latent representation x_n and the model parameters:

$$y_n | x_n \sim \mathcal{N}(W x_n + \mu_y,\, \sigma^2 \mathbb{I}_d) \tag{2.6}$$

We set a prior over the latent variable:

$$x_n \sim \mathcal{N}(0,\, \mathbb{I}_q) \tag{2.7}$$

and this allows us to find the marginal likelihood of the observable data y_n marginalised over the latent prior:

$$y_n \sim \mathcal{N}(\mu_y,\, \sigma^2 \mathbb{I}_d + W W^T) \tag{2.8}$$

Equation (2.8) represents the likelihood of the data to be generated from the latent model: this quantity should be maximised with respect to the model parameters W, μ and σ^2. This can be performed by maximum likelihood estimation over the Gaussian distribution eq. (2.8). However, as detailed in [9] this can be computationally expensive (it requires the eigendecomposition of the sample covariance matrix S) and therefore an alternative optimisation using the Expectation-Maximisation (EM) method is proposed.

With the PPCA method, we can derive a full posterior distribution of the latent representation x_n given y_n as:

$$x_n | y_n \sim \mathcal{N}(M^{-1} W^T (y - \mu_y),\, \sigma^2 M^{-1}) \tag{2.9}$$

and $M = W^T W + \sigma^2 \mathbb{I}_q$.
In contrast to PCA, where the projection for one data point y_n is given by a single point x_n in the latent space, in this case we instead have a full distribution $p(x_n | y_n)$.

As already stated, PPCA is also a generative model: by sampling a point in the latent space following eq. (2.7), we can use eq. (2.5) to generate a new data sample coming from the same distribution of y_n.

2.2 Latent variable models applied to Alzheimer's disease

Antelmi et al. proposed an extension of VAE for multiple views or modalities, i.e. different features of the data coming from different sources, and tested it with data from Alzheimer's patient [10].
In the case of neurodegenerative diseases like Alzheimer's these multi-view data can include clinical scores (standardised assessments of the cognitive abilities of the patient) and images coming from Magnetic Resonance Imaging scans (MRI) and Positron emission tomography (PET).

A recent work from Balelli, Silva, and Lorenzi [11] proposed a multi-view latent variable model based on the PPCA technique. In the same work, they also propose a federated learning scheme for training the latent variable model on multiple datasets stored in different centres, argument we will focus on with a detailed description in chapter 5.
Their latent variable model is similar to the one we will propose in chapter 3, since they are both based on the PPCA [9] technique. But while [11] extended PPCA for the use of multi-view data, our model focuses on the analysis of longitudinal data.

Chapter 3

Latent slope-intercept model: a latent variable model for longitudinal data

In this chapter we present the latent slope-intercept model (LSI), a latent variable model for the analysis of longitudinal data. We give the mathematical formulation of the model and the optimisation scheme. Further details are available in appendix A.

3.1 Theoretical formulation

The LSI model extends the framework of Probabilistic Principal Component Analysis (PPCA) [9], to handle longitudinal data, i.e. repeated measurements over time.
Let us consider data from N subjects: for each subject $n \in N$ we dispose of T_n samples $y_{nt} \in \mathbb{R}^d$, where $t \in \{t_{n1}, \dots, t_{nT_n}\}$ denotes the time point of sample acquisition. Overall, the data from each patient n can be summarized by a matrix Y_n of size $T_n \times d$, a d dimensional time-series of length T_n.
Assuming that the variations across features are correlated, our aim is to provide a common q-dimensional latent space representation of the d dynamics represented by Y_n.
Considering only one patient n and a generic time instant t (one row of the matrix Y_n), we assume that the observable variable y_{nt} follows the generative model equation given by:

$$y_{nt} = t(Wx_n + \omega) + Vx_n + \mu + \epsilon \tag{3.1}$$

where $\epsilon \sim \mathcal{N}(0,\, \sigma^2 \mathbb{I}_d)$ and x_n is the q-dimensional latent variable that models the evolution of all the features of patient n.

The evolution of Y_n over time is assumed to be linear and characterised by the intercept $Vx_n + \mu$ and the slope $Wx_n + \omega$: V and W are the $d \times q$ projection matrices for the intercept and the slope, respectively.
We note that we are using the same latent variable x_n for the projection of both the slope and the intercept.
In addition, we introduce the parameter ω as the average slope and μ as the average intercept to centre the projected slope and intercept around 0, thus allowing the model to have zero mean.
The term ϵ is an additive Gaussian observational noise with variance σ^2, independent and identically distributed (iid) for every t and along every dimension d. Appendix A provides a more detailed description of all the model parameters.

From eq. (3.1), one can derive the likelihood of the observable variables conditioned on the value of the latent variable x_n as:

$$y_{nt}|x_n \sim \mathcal{N}((tW + V)x_n + t\omega + \mu,\, \sigma^2\mathbb{I}_d) \tag{3.2}$$

By setting a prior on x_n, assuming independence between the latent dimensions:

$$x_n \sim \mathcal{N}(0,\, \mathbb{I}_q) \tag{3.3}$$

we can easily derive the marginal likelihood for the observed data:

$$y_{nt} \sim \mathcal{N}(t\omega + \mu,\, C) \tag{3.4}$$

where $C = \sigma^2\mathbb{I}_d + (tW + V)(tW + V)^T$

We note that eq. (3.4) provides a direct relation between the model parameters and the observable variables.

Exactly like the PPCA method, with the LSI we can derive a full posterior distribution of the latent representation x_n given y_{nt} as:

$$x_n|y_{nt} \sim \mathcal{N}(M_n^{-1}(tW + V)^T(y_{nt} - t\omega - \mu),\, \sigma^2 M_n^{-1}) \tag{3.5}$$

and $M_n = (tW + V)^T(tW + V) + \sigma^2\mathbb{I}_q$.

Up to this point, we considered only one patient n at a generic time t.
As stated before, for each patient we have T_n samples at time points $t_{n1}, \ldots, t_{nT_n}$.
In appendix A, we derive and report the complete vectorial formulation of the model for all N patients and all time instants t_n.

3.2 Optimisation

There are five unknown parameters in the LSI model: $W \in \mathbb{R}^{d\times q}$, $V \in \mathbb{R}^{d\times q}$, $\omega \in \mathbb{R}^d$, $\mu \in \mathbb{R}^d$ and $\sigma^2 \in \mathbb{R}$. In order to learn their optimal values, we want to maximize the marginal likelihood eq. (3.4) with respect to the model parameters (maximum likelihood estimation).
One possibility is to derive an exact analytic expression for the parameters, by setting the derivative of the log marginal likelihood with respect to each of them to 0. This can be computationally challenging since it requires to compute the sample covariance matrix in time, yielding to an asymptotic complexity of $\mathcal{O}(NT^2d^2)$.

Here, we decided to use the Expectation-Maximization method (EM method), by relying on the optimization framework originally introduced for PPCA [9]. EM is an iterative method which can be adapted to estimate the model parameters, accounting for the inference of the distribution of latent or missing variables. This method aims to maximise the marginal likelihood $p(y_n)$ indirectly, by iteratively maximising the expected complete log likelihood $\mathbb{E}_{p(x_n|y_n)} \ln p(y_n, x_n)$, corresponding to the expected value of the joint distribution of the observable and latent variables, computed with respect to the posterior distribution $p(x_n|y_n)$. It can be proven that, by applying this method, we converge to a local optimum of the marginal likelihood $p(y_n)$.
The algorithm is composed of two steps: an expectation step (E-step), in which the expected complete log likelihood is computed using the current estimates of the parameters, and a maximisation step (M-step) in which this likelihood is maximised with respect to the model parameters.
With our formulation, for the E-step we just need to compute the first and second moment of the posterior distribution eq. (3.5), while for the M-step we can derive an analytical closed form for the inference of every parameter of the model: W, V, ω, μ and σ^2. The results of these derivations are reported in the appendix A.2.
To sum up, we show in algorithm 1 the steps used to train the model, referring to the formulas derived and presented in the appendix A.2.

3.3 Implementation

The latent slope-intercept model was implemented in Python. The code was structured in a easy to use class, with a programming interface similar to the one of the scikit-learn library [12].
The class exposes several methods and allows to: (a) randomly initialise the model parameters, (b) generate synthetic data by sampling from the latent

Algorithm 1: Training of the latent slope-intercept model

randomly initialise W, V, ω, μ and σ^2

for $i = 1, \ldots, n\_epochs$ **do**

`/* E-step */`

compute $\langle x_n \rangle$ and $\langle x_n x_n^T \rangle$, the moments of the posterior $p(x_n|y_n)$, eqs. (A.14) and (A.15)

`/* M-step - optimise` $\mathbb{E}_{p(x_n|y_n)} \ln p(y_n, x_n)$ `with respect to the model parameters */`

compute $\widetilde{\mu}$, eq. (A.16)
compute $\widetilde{\omega}$, eq. (A.17)
compute $\widetilde{W}$, eq. (A.18)
compute $\widetilde{V}$, eq. (A.19)
compute $\widetilde{\sigma^2}$, eq. (A.20)

end

space, (c) encode (projection) and decode (inverse projection) from the latent space, (d) perform training of the model parameter and (e) load and save the model to file.

Particular attention is given to the computational efficiency of the implementation, since we aim to apply the model to large datasets that may take a long time to train. Whenever possible, the computations were written and implemented in vectorial form, to sensibly speed up the computation times, using numpy library for scientific computing [13].
Moreover we used the library numba [14], which performs Just-In-Time (JIT) compilation of compatible Python code, to further speed-up the code and parallelise the execution over all the available cores of the CPU.

Automated tests using the pytest framework [15] were performed. These tests allowed us to verify if the implemented vectorial formulas are correct and provide the same results as the non-vectorial formulas derived analytically. Moreover, we used the pytest-benchmark plugin [16] to compute the execution time for the optimisation of each model parameter. In this way, we were able to discover the presence of bottlenecks in the code and fix them.

Chapter 4

Applications

In this chapter, we show the results of the application of the LSI model on two datasets: a synthetically generated one and a dataset extracted from the Alzheimer's Disease Neuroimaging Initiative (ADNI) [17].

4.1 Synthetic dataset

By exploiting the generative capabilities of the LSI model, we can generate a synthetic dataset to assess the ability of LSI to recover the ground truth parameters. We sampled $N = 1000$ latent observations x_n, with $n = 0, \dots N-1$, of dimension $q_{gen} = 4$, which follow a standard Gaussian distribution $\mathcal{N}(0, \mathbb{I})$. We then applied the decoding equation eq. (3.1) with randomly generated W, V, ω and μ to reconstruct y_n in the observable space, of dimension $d = 10$. A Gaussian noise with variance $\sigma^2 = 3$ is added to each sample at every time point. For every subject n, we consider a random number of time points T_n between 2 and 10, sampled from a uniform distribution. Therefore, t_n is set as the integers between 0 and $T_n - 1$: for example if $T_n = 4$, we set $t_n = 0, 1, 2, 3$. In fig. 4.1, we provide an example of a synthetically generated sample. Each feature evolves almost linearly in time up to the additive noise.

4.1.1 Cross validation for hyperparameter tuning

The hyperparameter q, corresponding to the latent dimension is an user defined input for the LSI model.
We then set up an experiment to find out which is the best q to use for the specific dataset we are considering, something in machine learning we usually call hyperparameter tuning.
Actually, since we generated the data from a latent dimension of $q_{gen} = 4$, we already know that this is the best value for q: adding other dimensions will not

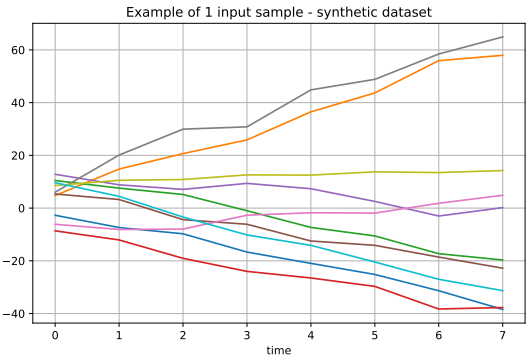


Figure 4.1 – Example of a randomly selected sample of the synthetic dataset. Each colour corresponds to a different dimension. We can observe the almost linear evolution in time, up to the additive noise.

improve the modelling of the data, since the additional dimensions will only capture noise and we risk to overfit. We are therefore using this experiment as a confirmation and verification for the correct functioning of the model.

The k-fold cross validation method (k-fold CV) is a very common one to test the generalisation of a model in machine learning and statistical learning. First of all, the training dataset is split in k non overlapping folds. Then, k iterations are performed: the model is fitted on $k-1$ folds and evaluated on the remaining fold, changing at each iteration the testing fold and resetting model. The final score of the model is the average of the score computed in each of the k iteration. Notice that with cross validation we are always evaluating the model on unseen data. Indeed the test dataset at a specific iteration is kept apart and not used for training. This is very important, because it allows for a unbiased evaluation of the performances on an independent dataset. Instead, in the case the evaluation was performed on the same data used for training, we will always prefer more complex models, risking to overfit them.

We performed a 5-fold cross validation, varying the dimension of the latent space q from 1 to 9. We consider the Mean Absolute Error (MAE) computed in the test set to perform model comparison and selection. In particular, MAE is defined as:

$$MAE = \frac{1}{Nd} \sum_{n} \frac{1}{T_n} |y_n - y_{n,est}| \tag{4.1}$$

where N is the number of patients in the test set, y_n is the ground truth and $y_{n,est}$ is the estimated sample obtained by projecting y_n into the latent space and reconstructing it.

Moreover, we also compare the marginal likelihood of the observable variable eq. (3.4) on the test folds. Note that in Bayesian machine learning, we usually compute the marginal likelihood, marginalised over the model parameters, and use it to perform model comparison by evaluating it on the training set (seen data). Here, we are instead computing the marginal likelihood marginalised over the latent variable. This measure should on one hand penalise larger dimensions of the latent space q, since the prior would spread on multiple dimensions and so decrease its value, thus avoiding overfitting. On the other hand, there may be a leakage of information from the training set through the model parameters, which can optimistically bias this measure. Therefore, we decided to compute it on the unseen split (as for the MAE) rather than on the whole dataset, so to rule out this potential bias.

Figure 4.2 shows the results of the 5-fold cross validation for the synthetic dataset. For each iteration of the CV, we trained the model for 2000 EM iterations, which allowed to reach convergence. According to both the MAE and the marginal likelihood, the best latent dimension is $q = 4$. This represent the value for which we have the minimum for the MAE and the maximum for the marginal likelihood. For latent dimensions $q > 4$, we do not get any improvements, meaning that further increasing the number of parameters does not improve the quality of the reconstruction. Our model was able to recover the true dimension of the latent space used for generating the data, $q_{gen} = 4$.

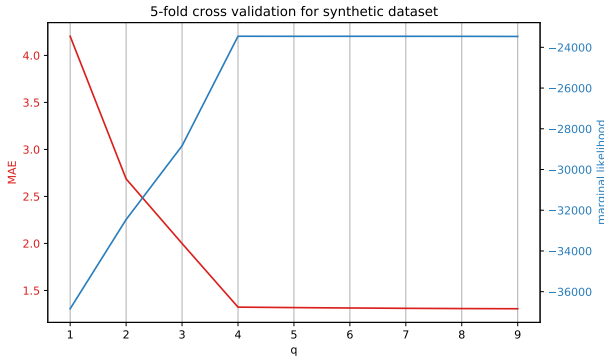


Figure 4.2 – 5-fold cross validation varying the latent dimension q for the synthetic dataset

4.1.2 Results and discussion

With the optimal value of $q = 4$ found by performing model comparison, we retrained the model for 2000 epochs on the synthetic dataset.

Figure 4.3 shows an example of reconstruction of a random training sample. The procedure to obtain the reconstruction is the following: the original data point is projected in the latent space by using the posterior distribution eq. (3.5) and then reconstructed by applying the modelling equation eq. (3.1). From the picture, we can clearly see that the first 4 dimensions (out of the total $d = 10$ dimensions) of the time-series are all matching the uncertainty region of the estimate identified by the model parameter σ.

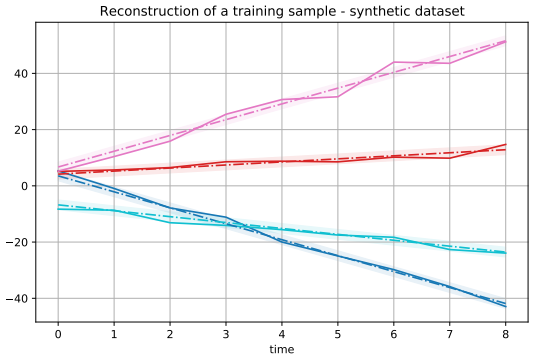


Figure 4.3 – Reconstruction of the first 4 dimensions of a training sample for the synthetic dataset. Solid lines correspond to real data, dashed lines to estimates.

Finally, we report the projection of the training data points on the latent space. In fig. 4.4 we focus on the first two dimensions x_0 and x_1. From these figures, we can observe that the points are distributed around the origin, with the same variance and the independence among the 2 dimensions. Indeed when we generated the synthetic dataset, we sampled from a $\mathcal{N}(0, \mathbb{I})$ and this was captured in the latent space of the trained model.
In fig. B.1, we report all the pairwise plots of the 4 dimensions of the latent space, where we observe the same distributions.

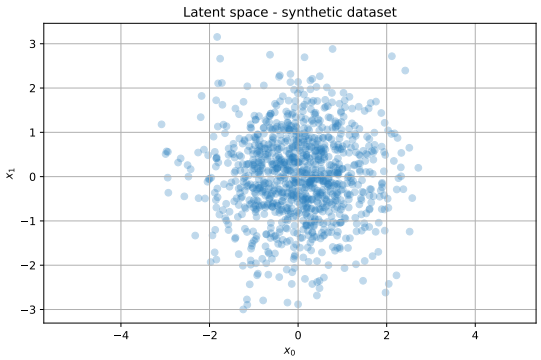


Figure 4.4 – Projection of training points on the first 2 dimensions of latent space for the synthetic dataset.

4.2 ADNI dataset

In this section we consider the Alzheimer's Disease Neuroimaging Initiative* (ADNI) [17] dataset.
For our experiments, we consider 8 clinical scores, which are standardised assessments given to patients after performing tasks and answering to specific questions: Alzheimer's Disease Assessment Scale - Cognitive 11-tasks version (ADAS11) [19], Mini-Mental State Examination (MMSE) [20], Clinical Dementia Rating Scale - Sum of Boxes (CDRSB) [21], Functional Activities Questionnaire (FAQ) [22] and Rey's Auditory Verbal Learning Test (RAVLT.immediate, RAVLT.learning, RAVLT.forgetting, RAVLT.perc.forgetting) [23, 24]. In addition we consider 5 brain volumes measurements extracted from Magnetic Resonance Imaging (MRI) scans: Whole Brain (WholeBrain), Ventricles, Hippocampus, Middle temporal gyrus (MidTemp) and Entorhinal.
In total we have $d = 13$ features for each exam taken by each patient.
More information about the clinical scores, the volume data and the procedures used to obtain them can be found in [25, 26].

*__Data acknowledgement:__ Data used in preparation of this thesis were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database [17]. As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found in [18].
The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been 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 dataset contains for each patient several exams, each with an exam date. We encode as t_n the number of years from the first exam (baseline). Finally, we performed some cleaning and preprocessing of the data. We removed for instance the exams containing missing values (nan) and the patients with only one exam. We also normalized each feature using min-max normalisation, so to range from 0 to 1.

After the preprocessing, we end up with a final dataset composed by a total of 2019 exams from 505 patients (175 diagnosed with AD and 330 cognitively normal).

4.2.1 Cross validation for hyperparameter tuning

For the ADNI dataset, we repeat the same 5-fold cross validation described in section 4.1.1.

In fig. 4.5, we observe a monotonic increasing function for the marginal likelihood and a monotonic decreasing function for the MAE. This is due to the fact that the ADNI dataset is complex and high dimensional, hence it requires a higher capacity latent space to reconstruct the data.
Nevertheless, we can notice that for $q > 6$, the relative variation of both the MAE and the marginal likelihood is milder. Therefore we will consider $q = 6$ as the optimal dimension of the latent space for the following experiments.

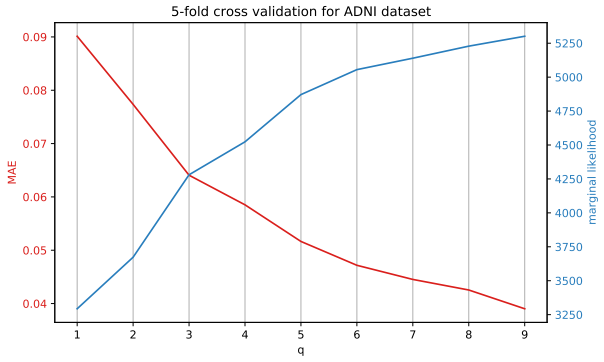
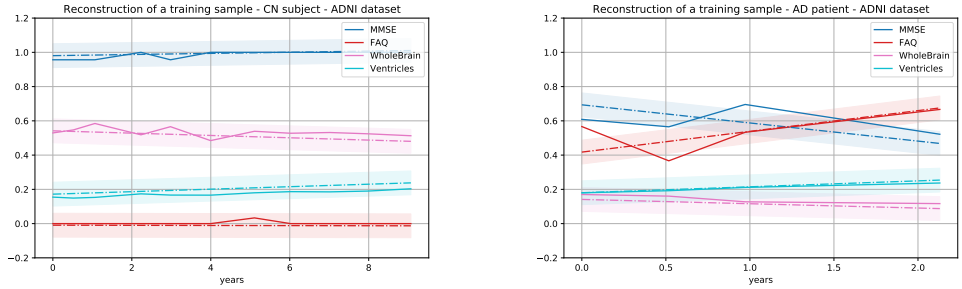


Figure 4.5 – 5-fold cross validation varying the latent dimension q for the ADNI dataset

4.2.2 Results and discussion

We train the model for 2000 iterations using $q = 6$, as discussed in the previous section.

In fig. 4.6 we show the reconstruction of two training samples, one from a cognitively normal (CN) subject and one from a patient affected by Alzheimer's disease (AD). The LSI model is able to reconstruct the trajectory of all the displayed features (MMSE, FAQ, WholeBrain, Ventricles) for the two subjects.



(a) CN subject

(b) AD patient

Figure 4.6 – Reconstruction of two training samples for the ADNI dataset. Solid lines correspond to real data, dashed lines to estimates.

We then report the latent representation for the training data points. In fig. 4.7 we show the projection on the latent dimensions x_1 and x_4, while in fig. B.2 we report all the pairwise scatter plots for the 6 dimensions of q. Interestingly, the observations coming from healthy individuals and the AD patients form two separated clusters in the latent space represented in fig. 4.7. We recall that the model has no knowledge of the diagnosis of the patient (it is unsupervised), but because of the intrinsic difference between healthy and sick patients, the data is projected in different regions of the latent space. This can be exploited in case we need to perform classification of cognitively normal subjects vs patients with Alzheimer's disease, by using the classifier on the latent space instead of the original higher dimensional space.

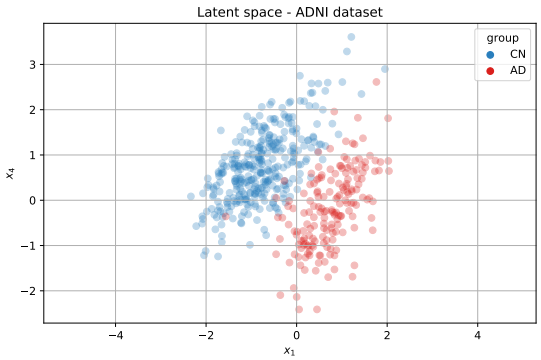


Figure 4.7 – Projection of training points on the dimension x_1 and x_4 of latent space for the ADNI dataset.

4.2.3 Medical interpretations

We give here a medical interpretation of the results obtained for the ADNI dataset. From the analysis of the optimised parameters of LSI model, we would like to understand the impact of AD on the clinical scores and biomarkers of the patients. To this end, we exploited the generative capabilities of the latent variable model and we simulated two synthetic patients, one cognitively normal and one affected by AD. Since we observed that patients tends to clusters according to the presence or not of the disease, we generated the synthetic patients using the latent representations determined by the centroids of the cluster of the healthy and of the AD patients. From these two latent points, we computed the slope and the intercept of the resulting time-series, which are given by:

$$\text{intercept} = Vx_n + \mu \tag{4.2}$$

$$\text{slope} = Wx_n + \omega \tag{4.3}$$

recalling the model equation eq. (3.1).
Figure 4.8 shows the value of the intercept for the 2 synthetic patients as a bar plot. If we focus on the clinical variables, we notice that the value of ADAS11, CDRSB, FAQ, RAVL.forgetting and RAVLT.perc.forgetting are higher in the AD patient with respect to the healthy one. These results match what we expect from the known medical effects of the disease.
For instance, the Alzheimer's Disease Assessment Scale - Cognitive 11-tasks version (ADAS11) [19] measures the severity of cognitive dysfunction in Alzheimer's: an higher ADAS11 means more dysfunctions. Similarly the Clinical Dementia Rating [21] tries to estimate the severity of dementia, in different cognitive categories: an high value may indicate the presence of dementia. The Functional

Assessment Questionnaire (FAQ) measures the ability of performing daily activities, higher means more dependence on others to perform those activities [22]. Rey's Auditory Verbal Learning Test aims to test the episodic memory of the patient, and it is performed by reading 15 words to the patients and ask him/her to recall them immediately and after a delay [23, 24]. As expected, the sick patients forgets more words than healthy ones after the delay (RAVLT.forgetting, RAVLT.perc.forgetting).
The remaining clinical scores have the opposite trend. For instance, the ability to immediately recall and learn the words diminishes in Alzheimer's patients (RAVLT.immediate, RAVLT.learning). Also the Mini Mental State Examination (MMSE), a simplified examination for the cognitive mental status, diminishes.

Other interesting considerations can be done for the brain volumes extracted from MRI images. The volume of the whole brain (WholeBrain) diminishes in the AD case. Also the brain regions known to be affected from the disease diminishes in volumes: Hippocampus, Middle temporal gyrus (MidTemp) and Entorhinal. The Ventricles, which are cavities inside the brain, are instead enlarging. In conclusion, the intercept, i.e. the punctual value, for the clinical and MRI volumes features changes from healthy to sick patients according to the expected medical effects of the disease.

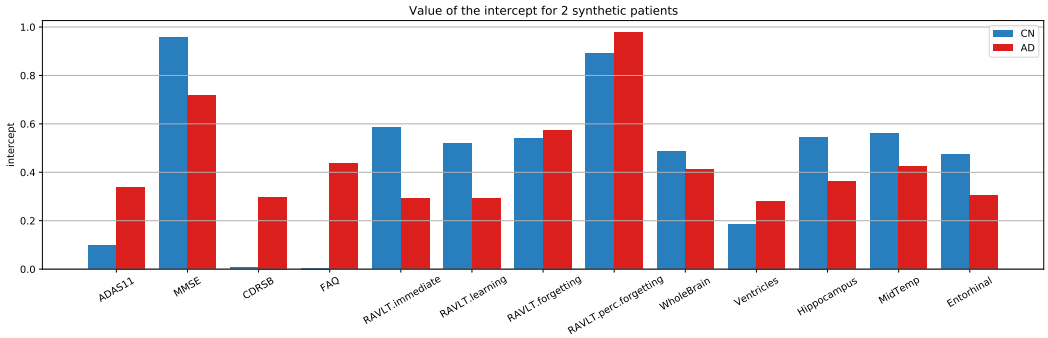


Figure 4.8 – Value of the intercept for the features of 2 synthetic patients. The values for the cognitively normal subject are plotted in blue, while the ones for the Alzheimer's patient are plotted in red.

We now consider the slope of the features for the 2 synthetic patients, fig. 4.9. Firstly, we notice is that in all cases the absolute value of the slope for AD is larger than for CN. If for instance we consider the MMSE, we notice that for a cognitive normal patient, the slope is slightly negative, meaning that, with ageing, the cognitive abilities are reducing. But the larger negative slope for the AD synthetic patient may indicate that, in the presence of the disease, the worsening is much faster. The same applies for all the other clinical scores: the

healthy subjects evolves toward the worsening of their cognitive and everyday life abilities, but in the case of AD patients the worsening is more evident and faster. As expected, all the brain volumes, except the Ventricles, are decreasing faster in the case of dementia. The Ventricle are on the opposite increasing faster. The only contradictory results is shown for RAVLT.forgetting, where we have a change of sign for the slope, yielding to a negative slope (reduction) for the AD case. Anyway, notice that the absolute value of the slope is very small, thus making it more susceptible to possible estimation errors.

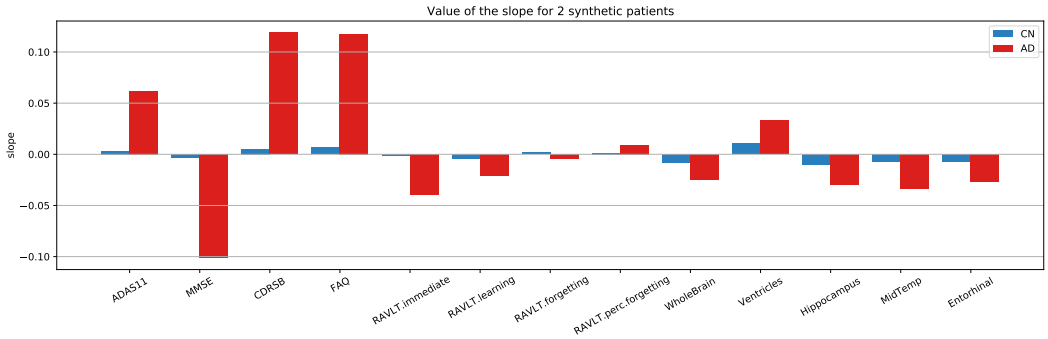


Figure 4.9 – Value of the slope for the features of 2 synthetic patients. The values for the cognitively normal subject are plotted in blue, while the ones for the Alzheimer's patient are plotted in red.

Chapter 5

Extension to multi-centric studies

In this chapter, we present an extension of the latent slope-intercept model to multi-centric studies, using a federated learning training scheme.

5.1 Motivation

The latent slope-intercept model we detailed in chapter 3 requires to be trained on a centralised dataset. The access to large amount of clinical data generated and stored in hospital or clinical centres becomes essential to provide good generalisation of the model.
However, due to privacy policies (such as the European GDPR [27]), raw data can not be shared across hospitals, nor with research centres. New learning strategies should then be developed to securely handle data distributed in different centres for training models.

5.2 Federated learning

In recent years, a new machine learning paradigm called federated learning (FL) has gained popularity to solve the issue of applying models to secured and sensitive decentralised data [28].
Traditional machine learning methods optimise the model parameters on a training dataset stored locally, on the device where the computations are performed. On the contrary, federated learning methods are specifically developed to train a model in a distributed and decentralised manner, with the data split across different clients.
In the biomedical context, every hospital or medical centre (the clients) locally trains on their private datasets a model provided by a researcher through a central server. Therefore, the clients return to the server the optimised parameters of their local models. Finally, the server combines those local models

to generate a global one. These steps are repeated for several rounds of communication clients-server, up to convergence: at each round, the clients use the current global model to initialize the local training. In this way, the final global model has been trained with more data (the union of all the private datasets) and so it will possibly generalise better. The great advantage of using FL is that the private medical data is never shared or disclosed to the central server, nor to the researcher: only model parameters are communicated.

In the literature, several aggregation strategies, i.e. procedures to combine the local models to get the global one, have been proposed. The standard aggregation strategy is federated averaging (FedAvg) [28], which essentially performs a weighted average of the local model parameters, based on the number of local samples. Algorithm 2 presents in details how the method works. We denote θ^r the global parameters of the model at round r, θ_c^r the parameters of the local model at round r, n_c the number of data points available in centre c and n the total amount of data across all the centres, $n = \sum_c n_c$.
Another strategy is FedProx [29], which is a generalisation of FedAvg that includes also a proximal term in the local function to optimise, so to avoid that the local models deviate too much from the global one. This methods yields to a more robust convergence, especially in the case of statistical heterogeneity of the datasets.

Algorithm 2: Federated Averaging training scheme

`server`: randomly initialise global parameters θ^0
foreach *learning round* $r = 0, \dots R-1$ **do**
  foreach *client* c **in parallel do**
    `server`: send the current global parameters θ^r
    `client` c: optimise the local parameters θ_c^{r+1} on the local dataset using SGD and θ^r as starting point
    `client` c: send the local parameters θ_c^{r+1} to the server
  end
  `server`: update the global parameters as $\theta^{r+1} = \sum_c \frac{n_c}{n} \theta_c^{r+1}$
end

FedAvg and FedProx are both designed specifically for training schemes based on Stochastic Gradient Descent (SGD), and so they are suited for instance for neural networks and deep learning architectures. On the contrary, our model is following a Bayesian approach and its optimisation scheme is

based on the Expectation-Maximisation.

Balelli, Silva, and Lorenzi [11] recently proposed a new fully Bayesian federated learning scheme for heterogeneous and distributed datasets, where parameters are optimised using the EM method.
The main idea behind the framework in [11] is to assume the existence of a hierarchical probabilistic structure, where the local parameters in each centre are sampled from a global distribution, and in turn the local data sampled from their local distribution, parametrised by the local parameters. The federated learning scheme developed in [11] can be adapted to propose a multi-centric extension to the LSI model presented in chapter 3.

5.3 Federated latent slope-intercept model

We consider C centres: each centre, indexed by c, owns a private dataset, composed of longitudinal data from N_c patients.

Recalling the notation of the centralised version of the model (see chapter 3 and appendix A), we set $\theta = \{W, V, \omega, \mu, \sigma^2\}$ as the ensemble of the unknown global model parameters, while θ_c denotes the local parameters for centre c.
We assume that for every $c = 1, \dots, C$, the parameters θ_c are described by a distribution $p(\theta_c|\theta)$ parametrised by θ, the global parameters.
We can set the distributions $p(\theta_c|\theta)$ to be Gaussian:

$$W_c \,|\, W \sim \mathcal{N}(W,\, \sigma_W^2) \tag{5.1}$$
$$V_c \,|\, V \sim \mathcal{N}(V,\, \sigma_V^2) \tag{5.2}$$
$$\omega_c \,|\, \omega \sim \mathcal{N}(\omega,\, \sigma_\omega^2) \tag{5.3}$$
$$\mu_c \,|\, \mu \sim \mathcal{N}(\mu,\, \sigma_\mu^2) \tag{5.4}$$

for all the parameters except σ_c^2, where a more reasonable choice may be an Inverse-Gamma distribution:

$$\sigma_c^2 \,|\, \sigma^2 \sim \text{Inverse-Gamma}(\alpha, \beta) \tag{5.5}$$

such that $\langle \sigma_c^2 \rangle = \sigma^2$.
The Inverse-Gamma distribution has a strictly positive support and it is therefore suited for σ_c^2 which must be non negative.

The training of the federated LSI model is performed both locally and globally. The local step is using the EM method to find a maximum a posteriori (MAP) estimate, instead of a maximum likelihood (ML) estimate.
In each centre, we aim to optimise the marginal likelihood of the data $p(y_n|\theta_c)$

(marginalised over the latent space) eq. (3.4), weighted by a prior $p(\theta_c|\theta)$ as follow:

$$\arg\max_{\theta_c} p(y_c|\theta_c)p(\theta_c|\theta) = \arg\max_{\theta_c} \ln p(y_c|\theta_c) + \ln p(\theta_c|\theta) \tag{5.6}$$

The prior $p(\theta_c|\theta)$ can be seen as a regularisation term, that forces the local distribution to not deviate too much from the global one.
At the server level, θ_c are aggregated using a maximum likelihood estimation on $p(\theta_c|\theta)$ to determine the global parameters θ. Notice that by choosing a Gaussian formulation for $p(\theta_c|\theta)$, the ML estimate for the parameters θ is the mean of the θ_c. The estimation for the Inverse-Gamma can be performed using the method proposed by [30].

The federated training procedure is summarised in Algorithm 3. The training can be implemented and run on one single machine, using different threads to simulate the various clients and keeping the private datasets separated. In this way, it is easy to test the convergence of the model and verify its correct functioning.

Algorithm 3: Federated latent slope-intercept model

`server`: randomly initialise global parameters θ
foreach *learning round* $r = 0, \dots R-1$ **do**
  foreach *client* c **in parallel do**
    `server`: send the current global parameters θ
    `client` c: optimise the local parameters θ_c on the local dataset using EM MAP on $p(y_c|\theta_c)p(\theta_c|\theta)$
    `client` c: send the local parameters θ_c to the server
  end
  `server`: optimise the global parameters θ using ML on $p(\theta_c|\theta)$
end

5.4 Towards real world application of multi-centric studies

After studying a theoretical framework for extending the latent slope-intercept model with a federated learning scheme (section 5.3), we devoted part of this work to the investigation of current software technologies for the effective deployment of federated learning in real scenarios.

For deploying a federated model in a real production environment, such as in hospitals and clinics, we need to satisfy several requirements: (a) to create a stable and easy-to-use software to install in the hospitals (the clients of the federated networks), where the actual training is taking place, (b) to create a central server that launches and coordinates the federated training and (c) to create an interface for the data scientists and researchers to easily deploy arbitrary machine learning models and train them with the data stored in the multiple centres, without getting access to the raw data.

FedBioMed [31, 32] is a framework developed in the Epione team at Inria and designed for federated learning in healthcare. The aim of this framework is precisely to provide the requirements listed above. During the work for this thesis, I contributed to the development of FedBioMed. In particular, I contributed to the complete code re-factoring of the framework, moving to the use of PySyft [33], a new promising Python library for federated learning.
We are planning to extend the framework to allow not only SGD based training (suited for neural networks and deep learning) but also general and arbitrary training schemes, like for instance the EM algorithm used in the federated latent slope-intercept model. With FedBioMed we are trying to create one of the first framework of this kind, ready for deployment in real hospitals and actually usable by researches and clinicians.

Chapter 6

Global disease progression model

The LSI model (chapter 3) makes the assumption that the time evolution is linear for each dimension. For the specific case of longitudinal data from Alzheimer's patients (chapter 4), this assumption can be reasonable since we consider follow-up visits that are usually few and close in time from the first visit (baseline).
But, if we want to consider instead a longer time evolution (for instance the entire history of the disease), this assumption is probably not holding. Indeed we expect that for a sick patient in a preliminary phase of the disease, the rate of variation will be small and it will abruptly accelerate going forward in the progression of the disease.
Moreover, it would be of great interest to estimate a global disease progression. So far, we have been studying individualised disease progressions, i.e. the evolution of the biomedical parameters for each specific individual. Now, we would like to find a global disease progression, so that the current medical status of each patient can be compared to it to understand the current stage of the disease in the global evolution.

More formally, we assume the existence of a global latent progression of the disease $x(s)$ along a common disease time reference s. Notice that s is not directly linked to the variable t_n which, in our experiments with the ADNI dataset, is the time (in years) from the first visit of patient n. Indeed, each patient can take its first visit at a different stage of the disease, and the disease itself can appear at different age in different patients.
From the generative equation eq. (3.1), we can write in expectation:

$$y(t, x(s)) = t(Wx(s) + \omega) + Vx(s) + \mu \tag{6.1}$$

by making explicit the dependency of y on t and $x(s)$.
We can then consider the eq. (6.1) as a first order Taylor approximation of the global disease evolution around $t \approx 0$.

For every $x(s)$, we recall in eq. (6.1) the intercept term $Vx(s) + \mu$ as the punctual value of y at time $t = 0$ and the slope term $Wx(s) + \omega$ as the rate of variation along the time t. From this, we write the global disease progression as a function of s, considering the value of the intercept term at $x(s)$:

$$y_*(s) := y(0, x(s)) = Vx(s) + \mu \tag{6.2}$$

and we impose its first derivative with respect to s to be equal to the slope term at $x(s)$:

$$\frac{dy_*(s)}{ds} := \frac{dy(t, x(s))}{dt} = Wx(s) + \omega \tag{6.3}$$

By applying the chain rule of calculus, from eq. (6.2) we can write:

$$\frac{dy_*(s)}{ds} = \frac{dy_*(s)}{dx}\frac{dx}{ds} = V\dot{x}(s) \tag{6.4}$$

and by imposing the condition in eq. (6.3) we obtain:

$$V\dot{x}(s) = Wx(s) + \omega \tag{6.5}$$

Finally, grouping eqs. (6.2) and (6.5), we get the following linear time-invariant system:

$$\begin{cases} \dot{x}(s) = V^+Wx(s) + V^+\omega \\ y_*(s) = Vx(s) + \mu \end{cases} \tag{6.6}$$

where $V^+ = (V^TV)^{-1}V^T$ is the pseudo-inverse of V.
Solving eq. (6.6) using the standard methods for linear dynamical systems, we get:

$$x(s) = e^{sV^+W}(x(0) + (V^+W)^{-1}V^+\omega) - (V^+W)^{-1}V^+\omega \tag{6.7}$$

$$y_*(s) = V(e^{sV^+W}(x(0) + (V^+W)^{-1}V^+\omega) - (V^+W)^{-1}V^+\omega) + \mu \tag{6.8}$$

Notice that the progression of the disease is exponential in s.

Chapter 7

Conclusions

In this project, we proposed the latent slope-intercept model, a latent variable model for the analysis of longitudinal data, based on Probabilistic Principal Component Analysis (PPCA).

Our tests focused in the first place on the application of the model to a synthetic dataset, created exploiting the generative capabilities of the model itself. A 5-folds cross validation has been applied to choose the best value for q, the dimension of the latent space and the only hyperparameter of the model. The optimal value for q we found matches with the value of q_{gen}, the ground truth dimension of the latent space. Moreover, the model was able to well reconstruct the training data starting from their projection on the latent space.

We then focused on the application on medical data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) [17]. By plotting the latent representations of the training data, we noticed that the points corresponding to a patient affected by Alzheimer's are distributed in a different region of the space with respect to the ones corresponding to healthy subjects. This differentiation has been learned by the model without ever providing it with the diagnosis of each patient.
We simulated two subject, one affected by AD and one cognitive normal, and we draw some medical conclusion on the estimated model parameters. In particular, we notice that the intercept (punctual value) of the biomarkers for AD patient have more pathological values than those of the healthy patient, consistently to the medical knowledge. Even more interestingly, the slopes of the biomedical parameters for AD patients are, in absolute value, higher than a healthy patient. This means that the model captured the fact that AD yields to a rapid worsening of the clinical status, in contrast to normal ageing.

We proposed an extension to the model, so that it can be trained with a federated learning scheme using data distributed in different centres. As we already stated, this is particularly relevant for medical data, which often cannot be transmitted outside of the hospital or clinic where they are captured. With this extension, the model can exploit the different private datasets for training, without requiring them to be moved in a centralised server.

Moreover, a global disease progression model has been proposed starting from the local individual progressions captured by LSI. With this generalisation, we get a solution of a dynamical system that allows us to draw an absolute disease trajectory over time in the latent space. From these, we expect to see an evolution from the cluster of healthy patients toward the portion of the space mapping AD patients.

The latent slope-intercept model presents several innovations with respect to currently available methods. First of all, it allows to analyse with a simple framework longitudinal data, of possibly different length in time. Moreover, due to the fact that the model is linear, it is very easily interpretable. Indeed, we were able to interpret the value and the speed of each biomarker and compare them for a sick and healthy patient. This is something very important in the medical field, and it is something that is usually not possible with more complex non-linear model, like for instance neural networks and deep learning, which are often treated as black boxes. Moreover, LSI presents the advantages of the Bayesian framework, such as providing a generative distribution, estimating the uncertainty of the results, and allowing to use automatic Bayesian model selection.
Finally, we recall that, beyond the biomedical motivation at the basis of this project, the proposed LSI model is general and of potential application to any kind of dataset.

As future work, we will implement in Python and test the two proposed extensions of the model: the federated version of LSI (chapter 5) and the global disease progression model (chapter 6).
Among other possible improvements of the LSI, we can indicate the extension to multi-view data, i.e. data coming from different sources (like MRI images, PET images, clinical scores) that up to now were simply concatenated and considered as a single view. Finally, we can exploit the capability of PPCA to handle missing data to perform a more complete analysis on the ADNI dataset, including incomplete exams that we excluded in the tests performed in this thesis.

Bibliography

[1] World Health Organization et al. "Global action plan on the public health response to dementia 2017–2025" (2017). URL: https://www.who.int/mental_health/neurology/dementia/action_plan_2017_2025/en/.

[2] Alzheimer's association. *Why Get Checked?* URL: https://www.alz.org/alzheimers-dementia/diagnosis/why-get-checked.

[3] Clifford R Jack Jr et al. "Hypothetical model of dynamic biomarkers of the Alzheimer's pathological cascade". *The Lancet Neurology* 9.1 (2010), pp. 119–128.

[4] Clement Abi Nader et al. "Alzheimer's Disease Modelling and Staging through Independent Gaussian Process Analysis of Spatio-Temporal Brain Changes". *Machine Learning in Clinical Neuroimaging (MLCN) workshop*. Granada, Spain, Sept. 2018. URL: https://hal.archives-ouvertes.fr/hal-01882450.

[5] Ian Goodfellow et al. "Generative adversarial nets". *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[6] Tero Karras, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4401–4410.

[7] Diederik Kingma and Max Welling. "Auto-Encoding Variational Bayes". International Conference on Learning Representations. 2014.

[8] Ian T Jolliffe. "Principal components in regression analysis". *Principal component analysis*. Springer, 1986, pp. 129–155.

[9] Michael E Tipping and Christopher M Bishop. "Probabilistic principal component analysis". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622.

[10] Luigi Antelmi et al. "Sparse Multi-Channel Variational Autoencoder for the Joint Analysis of Heterogeneous Data". *Proceedings of Machine Learning Research*. Proceedings of ICML 2019 97 (2019), pp. 302–311. URL: https://hal.inria.fr/hal-02395747.

[11] Irene Balelli, Santiago Silva, and Marco Lorenzi. *A Probabilistic Framework for Modeling the Variability Across Federated Datasets of Heterogeneous Multi-View Observations*. Accepted for publication, IPMI 2021.

[12] Scikit-learn team. *APIs of scikit-learn objects*. URL: https://scikit-learn.org/stable/developers/develop.html#apis-of-scikit-learn-objects.

[13] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. "The NumPy array: a structure for efficient numerical computation". *Computing in Science & Engineering* 13.2 (2011), p. 22.

[14] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. "Numba: a LLVM-based Python JIT compiler". *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. 2015, pp. 1–6.

[15] *pytest framework*. URL: https://docs.pytest.org/en/stable/.

[16] *pytest-benchmark plugin*. URL: https://pytest-benchmark.readthedocs.io/en/latest/.

[17] *Alzheimer's Disease Neuroimaging Initiative (ADNI)*. URL: http://adni.loni.usc.edu.

[18] Alzheimer's Disease Neuroimaging Initiative. *ADNI contributors*. URL: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

[19] Jacqueline K Kueper, Mark Speechley, and Manuel Montero-Odasso. "The Alzheimer's disease assessment scale–cognitive subscale (ADAS-Cog): modifications and responsiveness in pre-dementia populations. a narrative review". *Journal of Alzheimer's Disease* 63.2 (2018), pp. 423–444.

[20] Marshal F Folstein, Susan E Folstein, and Paul R McHugh. "Mini-mental state: a practical method for grading the cognitive state of patients for the clinician". *Journal of psychiatric research* 12.3 (1975), pp. 189–198.

[21] John C Morris. "The clinical dementia rating (cdr): Current version and scoring rules". *Young* 41 (1991), pp. 1588–1592.

[22] Robert I Pfeffer et al. "Measurement of functional activities in older adults in the community". *Journal of gerontology* 37.3 (1982), pp. 323–329.

[23] A Rey. "L'examen Clinique en Psychologie Paris: Presses Universitaires de France." (1964).

[24] Elaheh Moradi et al. "Rey's Auditory Verbal Learning Test scores can be predicted from whole brain MRI in Alzheimer's disease". *NeuroImage: Clinical* 13 (2017), pp. 415–427.

[25] Alzheimer's Disease Neuroimaging Initiative. *ADNI Data Inventory*. URL: http://adni.loni.usc.edu/data-samples/adni-data-inventory/.

[26] Alzheimer's Disease Neuroimaging Initiative. *ADNI General Procedures Manual*. URL: http://adni.loni.usc.edu/wp-content/uploads/2010/09/ADNI_GeneralProceduresManual.pdf.

[27] European Parliament and Council of European Union. *Regulation (EU) 2016/679*. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[28] Brendan McMahan et al. "Communication-efficient learning of deep networks from decentralized data". *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 1273–1282.

[29] Tian Li et al. "Federated optimization in heterogeneous networks". *arXiv preprint arXiv:1812.06127* (2018).

[30] A Llera and CF Beckmann. "Estimating an inverse gamma distribution". *arXiv preprint arXiv:1605.01019* (2016).

[31] Santiago Silva et al. "Fed-BioMed: A General Open-Source Frontend Framework for Federated Learning in Healthcare". *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*. Springer, 2020, pp. 201–210.

[32] *FedBioMed*. URL: https://fedbiomed.gitlabpages.inria.fr.

[33] Theo Ryffel et al. "A generic framework for privacy preserving deep learning". *arXiv preprint arXiv:1811.04017* (2018).

Appendix A

Latent slope-intercept model formulation

A.1 Model definition

For every subject n, we have:

$$\begin{aligned} y_n &= t_n \otimes (W x_n + \omega) + 1_{T_n} \otimes (V x_n + \mu) + \mathcal{E} \\ &= (t_n \otimes W + V^\dagger) x_n + t_n \otimes \omega + \mu^\dagger + \mathcal{E} \end{aligned} \tag{A.1}$$

where

- $t_n = [t_{n1} \dots t_{nT_n}]^T$ is a T_n vector of time instants, different for each subject n
- y_n are the samples for subject n. It is a vector $T_n d$, where T_n is the number of time instants and d is the number of features. It is the vectorisation (row-major) of a matrix Y_n, of dimensions $T_n \times d$, where each row represents a measurement performed at the same time instant:

$$y_n = vec(Y_n) = vec\left(\begin{bmatrix} y_{nt_{n1}1} \dots y_{nt_{n1}d} \\ \vdots \\ y_{nt_{nT_n}1} \dots y_{nt_{nT_n}d} \end{bmatrix}\right) = \begin{bmatrix} y_{nt_{n1}1} \\ \vdots \\ y_{nt_{n1}d} \\ \vdots \\ \vdots \\ y_{nt_{nT_n}1} \\ \vdots \\ y_{nt_{nT_n}d} \end{bmatrix}$$

- x_n is the latent variable, dimension q
- W is the projection matrix of the slope on the latent space, dimension $d \times q$
- V is the projection matrix of the intercept on the latent space, dimension $d \times q$
- ω is the average slope, dimension d

- μ is the average intercept, dimension d
- $\mathcal{E} \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_{T_n d})$ is an additive Gaussian noise, iid for every subject and for every time instant
- $V^\dagger = 1_{T_n} \otimes V$ and $\mu^\dagger = 1_{T_n} \otimes \mu$, for a more compact notation
- $\otimes$ is the Kronecker product

In particular, focusing at a generic time instant t, from eq. (A.1) we have:

$$\begin{aligned} y_{nt} &= t(Wx_n + \omega) + Vx_n + \mu + \epsilon \\ &= (tW + V)x_n + t\omega + \mu + \epsilon \end{aligned} \tag{A.2}$$

and $\epsilon \sim \mathcal{N}(0,\, \sigma^2 \mathbb{I}_d)$.

A.1.1 Likelihood and marginal likelihood

From eq. (A.1), we can derive the likelihood of the model to generate the data y_{nt}, given the latent representation x_n:

$$y_n | x_n \sim \mathcal{N}((t_n \otimes W + V^\dagger)x_n + t_n \otimes \omega + \mu^\dagger,\, \sigma^2 \mathbb{I}_{T_n d}) \tag{A.3}$$
likelihood

By considering the following prior on the latent variable x_n:

$$x_n \sim \mathcal{N}(0,\, \mathbb{I}_q) \tag{A.4}$$
prior

we can derive the marginal likelihood, i.e. the likelihood of the data y_n marginalised over the prior x_n, by applying a simple linear transformation of Gaussians:

$$y_n \sim \mathcal{N}(t_n \otimes \omega + \mu^\dagger,\, C_n) \tag{A.5}$$
marginal likelihood

where $C_n = \sigma^2 \mathbb{I}_{T_n d} + (t_n \otimes W + V^\dagger)(t_n \otimes W + V^\dagger)^T$

From eq. (A.5), we can notice that, for the same subject n, the observations y_{nt} from different time instants are not independent.

A.1.2 Posterior distribution

We want to derive $p\left(x_n | y_{nt_{n1}}, \ldots, y_{nt_{nT_n}}\right)$. We can do that applying Bayes rule:

$$p(x_n | y_{nt_{n1}}, \ldots, y_{nt_{nT_n}}) = \frac{p(y_{nt_{n1}}, \ldots, y_{nt_{nT_n}} | x_n) p(x_n)}{p(y_{nt_{n1}}, \ldots, y_{nt_{nT_n}})} \tag{A.6}$$

Since the likelihood and the prior are Gaussian, the posterior will be Gaussian as well because of conjugacy. Therefore, it is sufficient to derive its mean and variance, by

comparing what we get from the numerator of eq. (A.6) to the final form of the posterior we expect. The form of the posterior, omitting the terms not dependent on x_n, is given by:

$$\begin{aligned} x_n|y_{nt_{n1}},\dots,y_{nt_{nT_n}} &\sim exp\left\{-\frac{1}{2}(x_n-m)^T\Sigma^{-1}(x_n-m)\right\} \\ &\sim exp\left\{-\frac{1}{2}x_n^T\Sigma^{-1}x_n-\frac{1}{2}m^T\Sigma^{-1}m+x_n^T\Sigma^{-1}m\right\} \\ &\sim exp\left\{-\frac{1}{2}x_n^T\Sigma^{-1}x_n+x_n^T\Sigma^{-1}m\right\} \end{aligned} \tag{A.7}$$

As reported above, we don't have independence among different time instants of y_{nt}, but we have conditional independence of $y_{nt}|x_n$. We can exploit this fact and write:

$$p(y_{nt_{n1}},\dots,y_{nt_{nT_n}}|x_n)=p(y_{nt_{n1}}|x_n)\dots p(y_{nt_{nT_n}}|x_n)$$

So,

$$\begin{aligned} &\left(y_{nt_{n1}},\dots,y_{nt_{nT_n}}|x_n\right)\ (x_n) \\ &\sim exp\left\{-\frac{1}{2\sigma^2}\sum_{t\in t_n}\left(||y_{nt}-(tW+V)x_n-t\omega-\mu||^2\right)-\frac{1}{2}x_n^Tx_n\right\} \\ &\sim exp\left\{-\frac{1}{2\sigma^2}\sum_{t\in t_n}\left(x_n^T(tW+V)^T(tW+V)x_n-2x_n^T(tW+V)^T\left(y_{nt}-t\omega-\mu\right)\right)-\frac{1}{2}x_n^Tx_n\right\} \\ &\sim exp\left\{-\frac{1}{2}x_n^T\left(\frac{1}{\sigma^2}\sum_{t\in t_n}(tW+V)^T(tW+V)+\mathbb{I}\right)x_n+x_n^T\left(\frac{1}{\sigma^2}\sum_{t\in t_n}(tW+V)^T\left(y_{nt}-t\omega-\mu\right)\right)\right\} \end{aligned} \tag{A.8}$$

By comparing eqs. (A.7) and (A.8), we get

$$\begin{aligned} \Sigma^{-1}=\frac{1}{\sigma^2}\sum_{t\in t_n}(tW+V)^T(tW+V)+\mathbb{I}=\frac{1}{\sigma^2}\left(\sum_{t\in t_n}(tW+V)^T(tW+V)+\sigma^2\mathbb{I}\right) \\ \Sigma=\sigma^2\left(\sum_{t\in t_n}(tW+V)^T(tW+V)+\sigma^2\mathbb{I}\right)^{-1}=\sigma^2M_n^{-1} \end{aligned} \tag{A.9}$$

and

$$\begin{aligned} \Sigma^{-1}m&=\frac{1}{\sigma^2}\sum_{t\in t_n}(tW+V)^T\left(y_{nt}-t\omega-\mu\right) \\ m&=M_n^{-1}\sum_{t\in t_n}(tW+V)^T\left(y_{nt}-t\omega-\mu\right) \end{aligned} \tag{A.10}$$

Overall the posterior is given by:

$$x_n|y_{nt_{n1}},\dots,y_{nt_{nT_n}}\sim\mathcal{N}(M_n^{-1}\sum_{t\in t_n}(tW+V)^T\left(y_{nt}-t\omega-\mu\right),\sigma^2M_n^{-1}) \tag{A.11}$$

posterior

with $M_n=\sum_{t\in t_n}(tW+V)^T(tW+V)+\sigma^2\mathbb{I}=\tau_nW^TW+\eta_nW^TV+\eta_nV^TW+T_nV^TV+\sigma^2\mathbb{I}$ and $\eta_n=\sum_{t\in t_n}t,\quad\tau_n=\sum_{t\in t_n}t^2$.

A.1.3 Predictive distribution

We derive the predictive distribution $y_{nt_*}|y_{nt_{n1}}, \dots, y_{nt_{nT_n}}$, to compute predictions for future time instants. Combining the posterior eq. (A.11) and the likelihood for a new time instant t_* :

$$y_{nt_*}|x_n \sim \mathcal{N}((t_*W + V)x_n + t_*\omega + \mu,\, \sigma^2\mathbb{I})$$

we get the following predictive distribution:

$$y_{nt_*}|y_{nt_{n1}}, \dots, y_{nt_{nT_n}} \sim \mathcal{N}((t_*W + V)M_n^{-1}\sum_{t\in t_n}(tW + V)^T\,(y_{nt} - t\omega - \mu) + t_*\omega + \mu,$$
$$\sigma^2\mathbb{I} + \sigma^2(t_*W + V)M_n^{-1}(t_*W + V)^T) \tag{A.12}$$

predictive

by marginalising on x_n.

A.2 Parameters estimation

We would like to maximise the marginal likelihood eq. (A.5) with respect to the parameters of the model (W, V, μ, ω, σ^2). We can do that using the EM algorithm.
We start by deriving the complete log-likelihood of the parameters, which is given by:

$$\mathcal{L} = \sum_n \ln p(y_{nt_{n1}}, \dots, y_{nt_{T_n}}, x_n) = \sum_n \left(\sum_{t\in t_n} \ln p(y_{nt}|x_n) + \ln p(x_n)\right) =$$
$$= -\sum_n \left(\sum_{t\in t_n}\left(\frac{d}{2}\ln\sigma^2 + \frac{1}{2\sigma^2}\left\|y_{nt} - (tW + V)x_n - t\omega - \mu\right\|^2\right) + \frac{1}{2}\left\|x_n\right\|^2\right) =$$
$$= -\sum_n \left(\sum_{t\in t_n}\left(\frac{d}{2}\ln\sigma^2 + \frac{1}{2\sigma^2}\big(\left\|y_{nt} - t\omega - \mu\right\|^2 + x_n^T(tW + V)^T(tW + V)x_n + \right.\right.\right.$$
$$\left.\left.\left. -2x_n^T(tW + V)^T(y_{nt} - t\omega - \mu)\big)\right) + \frac{1}{2}\left\|x_n\right\|^2\right)$$

We compute its expectation with respect to $x_n|y_{nt}$:

$$\langle\mathcal{L}\rangle = -\sum_n \left(\sum_{t\in t_n}\left(\frac{d}{2}\ln\sigma^2 + \frac{1}{2\sigma^2}\big(\left\|y_{nt} - t\omega - \mu\right\|^2 + \mathrm{Tr}((tW + V)^T(tW + V)\langle x_n x_n^T\rangle)\right.\right.$$
$$\left.\left. -2\langle x_n\rangle^T(tW + V)^T(y_{nt} - t\omega - \mu)\big)\right) + \frac{1}{2}\mathrm{Tr}(\langle x_n x_n^T\rangle)\right) \tag{A.13}$$

expected complete log-likelihood

where the first and second order moments of $x_n|y_{nt}$ are given by:

$$\langle x_n\rangle = M_n^{-1}\sum_{t\in t_n}(tW + V)^T\,(y_{nt} - t\omega - \mu) \tag{A.14}$$
$$\langle x_n x_n^T\rangle = \sigma^2 M_n^{-1} + \langle x_n\rangle\langle x_n\rangle^T \tag{A.15}$$

The strategy is now to maximise the expected complete log-likelihood eq. (A.13) with respect to the model parameters. It was proven that doing that is equivalent to maximise the marginal likelihood eq. (A.5).

Estimation of $\widetilde{\mu}$

To estimate the optimal value for μ, we rearrange the expected log-likelihood eq. (A.13), excluding the terms not depending on μ:

$$\langle \mathcal{L} \rangle = -\sum_n \sum_{t \in t_n} \frac{1}{2\sigma^2} \big(\mu^T \mu - 2(y_{nt} - t\omega)^T \mu + 2\langle x_n \rangle^T (tW + V)^T \mu \big)$$

we differentiate it with respect to μ and set the derivative to 0:

$$\begin{aligned} \frac{\partial \langle \mathcal{L} \rangle}{\partial \mu} &= -\sum_n \sum_{t \in t_n} \frac{1}{2\sigma^2} \big(2\mu - 2(y_{nt} - t\omega) + 2(tW + V)\langle x_n \rangle \big) = 0 \\ &\implies \widetilde{\mu} = \frac{1}{\sum_n T_n} \sum_n \sum_{t \in t_n} (y_{nt} - (tW + V)\langle x_n \rangle - t\omega) \end{aligned} \tag{A.16}$$

Estimation of $\widetilde{\omega}$

We proceed in the same way for deriving $\widetilde{\omega}$:

$$\langle \mathcal{L} \rangle = -\sum_n \sum_{t \in t_n} \frac{1}{2\sigma^2} \big(t^2 \omega^T \omega - 2t(y_{nt} - \mu)^T \omega + 2t\langle x_n \rangle^T (tW + V)^T \omega \big)$$

$$\begin{aligned} \frac{\partial \langle \mathcal{L} \rangle}{\partial \omega} &= -\sum_n \sum_{t \in t_n} \frac{1}{2\sigma^2} \big(2t^2 \omega - 2t(y_{nt} - \mu) + 2t(tW + V)\langle x_n \rangle \big) = 0 \\ &\implies \widetilde{\omega} = \frac{1}{\sum_n \tau_n} \sum_n \sum_{t \in t_n} t \left(y_{nt} - (tW + V)\langle x_n \rangle - \mu \right) \end{aligned} \tag{A.17}$$

Estimation of $\widetilde{W}$

We rewrite the expected log-likelihood eq. (A.13) to simplify the computation of the derivative with respect to the entries of the matrix W. We also omit the constant terms with respect to W.

$$\begin{aligned} \langle \mathcal{L} \rangle = -\sum_n \sum_{t \in t_n} \frac{1}{2\sigma^2} \big(& t^2 \operatorname{Tr}(W^T W \langle x_n x_n^T \rangle) + t \operatorname{Tr}(W^T V \langle x_n x_n^T \rangle) + t \operatorname{Tr}(V^T W \langle x_n x_n^T \rangle) \\ & - 2t\langle x_n \rangle^T W^T (y_{nt} - t\omega - \mu) \big) \end{aligned}$$

We differentiate the previous equation with respect to W and set the derivative to 0, obtaining:

$$\begin{aligned} \frac{\partial \langle \mathcal{L} \rangle}{\partial W} &= -\sum_n \sum_{t \in t_n} \frac{1}{2\sigma^2} \big(2t^2 W \langle x_n x_n^T \rangle + 2tV \langle x_n x_n^T \rangle - 2t(y_{nt} - t\omega - \mu)\langle x_n \rangle^T \big) = 0 \\ \implies \widetilde{W} &= \Big(\sum_n \sum_{t \in t_n} t(y_{nt} - t\omega - \mu)\langle x_n \rangle^T - tV \langle x_n x_n^T \rangle \Big) \Big(\sum_n \tau_n \langle x_n x_n^T \rangle \Big)^{-1} \end{aligned} \tag{A.18}$$

with $\tau_n = \sum_{t \in t_n} t^2$

Estimation of $\widetilde{V}$

We proceed in the same way for deriving $\widetilde{V}$:

$$\langle \mathcal{L} \rangle = -\sum_n \sum_{t \in t_n} \frac{1}{2\sigma^2} \big(\mathrm{Tr}(V^T V \langle x_n x_n^T \rangle) + t\, \mathrm{Tr}(W^T V \langle x_n x_n^T \rangle) + t\, \mathrm{Tr}(V^T W \langle x_n x_n^T \rangle) \\ - 2 \langle x_n \rangle^T V^T (y_{nt} - t\omega - \mu) \big)$$

$$\frac{\partial \langle \mathcal{L} \rangle}{\partial V} = -\sum_n \sum_{t \in t_n} \frac{1}{2\sigma^2} \big(2V \langle x_n x_n^T \rangle + 2tW \langle x_n x_n^T \rangle) - 2(y_{nt} - t\omega - \mu) \langle x_n \rangle^T \big) = 0$$

$$\implies \widetilde{V} = \Big(\sum_n \sum_{t \in t_n} (y_{nt} - t\omega - \mu) \langle x_n \rangle^T - tW \langle x_n x_n^T \rangle \Big) \Big(\sum_n T_n \langle x_n x_n^T \rangle \Big)^{-1} \qquad \text{(A.19)}$$

Estimation of $\widetilde{\sigma^2}$

We proceed in the same way as for deriving $\widetilde{\sigma^2}$:

$$\frac{\partial \langle \mathcal{L} \rangle}{\partial \sigma^2} = -\sum_n \sum_{t \in t_n} \frac{d}{2\sigma^2} - \frac{1}{2(\sigma^2)^2} \big(\, \| y_{nt} - t\omega - \mu \|^2 + \mathrm{Tr}((tW + V)^T (tW + V) \langle x_n x_n^T \rangle) \\ - 2 \langle x_n \rangle^T (tW + V)^T (y_{nt} - t\omega - \mu) \big) = 0$$

$$\implies \widetilde{\sigma^2} = \frac{1}{d \sum_n T_n} \sum_n \sum_{t \in t_n} \big(\, \| y_{nt} - t\omega - \mu \|^2 + \mathrm{Tr}((tW + V)^T (tW + V) \langle x_n x_n^T \rangle) \\ - 2 \langle x_n \rangle^T (tW + V)^T (y_{nt} - t\omega - \mu) \big) \qquad \text{(A.20)}$$

Appendix B

Additional figures

We report here some additional figures from the experiments performed in chapter 4.

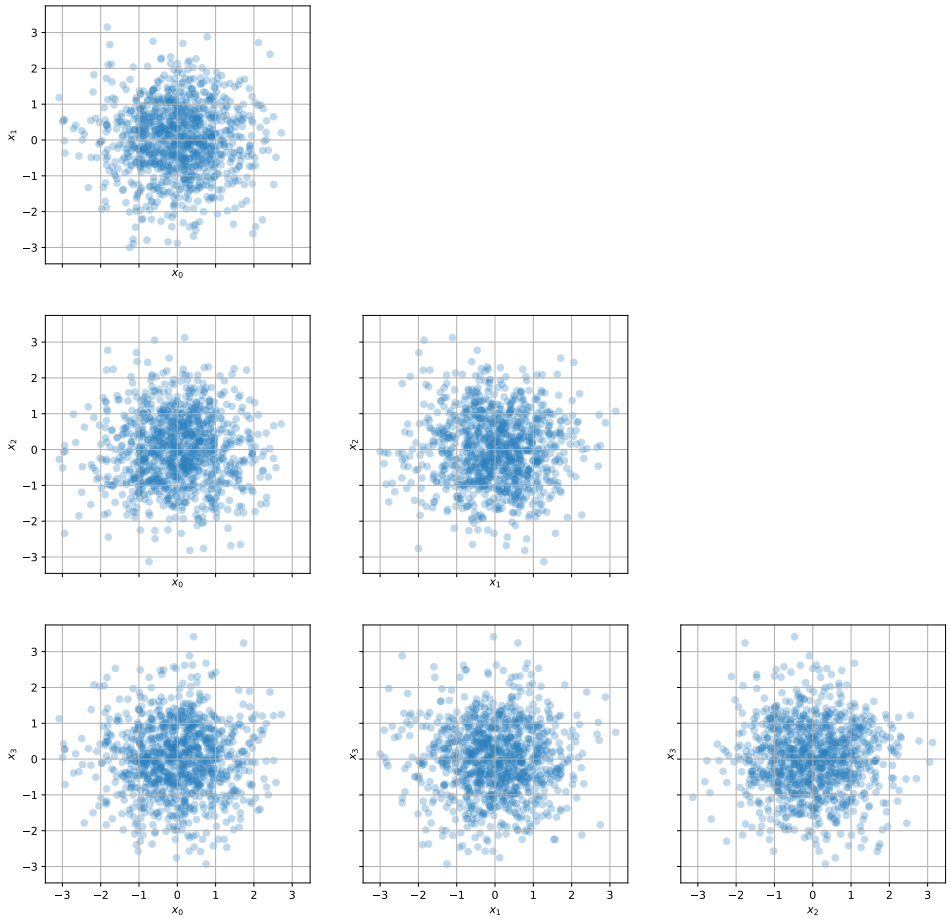


Figure B.1 – Pairwise scatter plots of the latent space of dimension $q = 4$ for the synthetic dataset. Notice that the latent dimensions are independent on each others and centred around the origin.

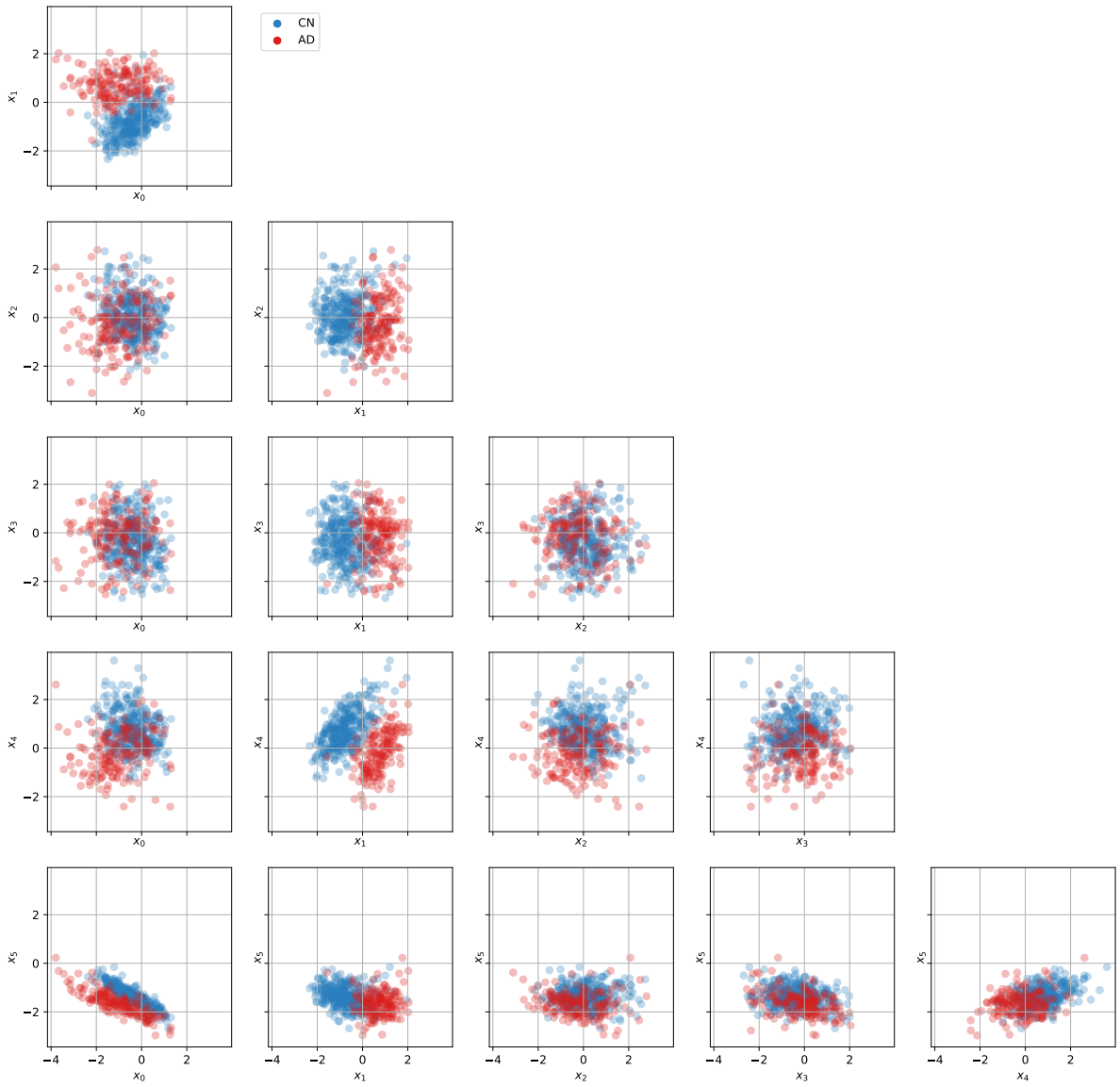


Figure B.2 – Pairwise scatter plots of the latent space of dimension $q = 6$ for the ADNI dataset. Notice that for some latent dimensions the AD and CN subjects form two separated clusters.