GRAPH-BASED MACHINE LEARNING ALGORITHMS FOR PREDICTING DISEASE OUTCOMES

Juliette VALENCHON

Master Thesis

Department of Electrical and Computer Engineering
McGill University
Montréal, Québec, Canada
March 2019

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering.

Abstract

Improving disease outcome prediction can greatly aid in the strategic deployment of secondary prevention approaches. We develop two methods to predict the evolution of diseases by taking into account personal attributes of the subjects and their relationships with medical examination results. Our approaches build upon a recent formulation of this problem as a graph-based geometric matrix completion task. The primary innovation is the introduction of multiple graphs, each relying on a different combination of subject attributes. Via statistical significance tests, we determine the relevant graph(s) for each medically-derived feature. In the first approach, we then employ a multiple-graph recurrent graph convolutional neural network architecture to predict the disease outcomes. In the second approach, we use a multiple-graph graph auto-encoder architecture to predict the disease outcomes. We demonstrate the efficacy of the two techniques by addressing the task of predicting the development of Alzheimer's disease for patients exhibiting mild cognitive impairment, showing that the incorporation of multiple graphs improves predictive capability. Moreover, in the second approach, the use of a graph auto-encoder also helps in increasing predictive capability.

Résumé

Améliorer la prédiction de maladies est certainement très bénéfique pour le développement d'approches de prévention secondaire. Nous proposons dans cette thèse deux méthodes pour prédire l'évolution de maladies en prenant en compte certaines caractéristiques personnelles des sujets et leur relation avec les résultats des examen médicaux. Nos approches sont fondées sur la récente formulation de ce problème comme étant une méthode géométrique de complétion de matrices basée sur un graphe. La principale innovation proposée est l'introduction de plusieurs graphes, chacun étant basé sur une combinaison différente de caractéristiques des patients. Par le biais de tests statistiques, nous trouvons pour chaque élément des examen médicaux le(s) graphe(s) au(x)quel(s) il est associé. Dans une première approche, nous employons une architecture basée sur des réseaux de neurones convolutifs sur plusieurs graphes pour prédire des maladies. Dans une seconde approche, nous utilisons un auto-encodeur construit sur plusieurs graphes pour la même tâche de prédiction de maladies. Nous testons ces deux méthodes pour la tâche de prédiction du développement de la maladie d'Alzheimer pour des patients atteints d'une déficience cognitive légère et nous montrons que l'incorporation de plusieurs graphes aide à augmenter la capacité de prédiction des deux algorithmes. De plus, dans la deuxième approche, l'utilisation d'un auto-encodeur construit sur plusieurs graphes aide aussi à augmenter la capacité de prédiction de maladies.

Acknowledgements

I wish to express my sincere thanks and gratitude to my supervisor Professor Mark Coates for having accepted me within his team, for his support, his continuous follow-up and his very helpful advice, scientifically speaking but also for the writing of my thesis. He transmitted me critical knowledge that have been essential in my work as a research student and will also be very helpful in my future work.

I would also like to thank Professor Milica Popovic for her technical and moral support during all my Master and in particular during the first year of my Master when I was following the breast cancer detection project.

I would like to thank Florence Robert-Regol (Master's student) and Soumyasundar Pal (PhD candidate) for the numerous discussions that we had on many interesting topics such as those on graph convolutional neural networks or other discussions related to graph signal processing and machine learning techniques for graph-structured data. I would also like to thank them for their help while I was having issues with my work, both theoretically and during the implementation step. I would also like to thank Laure Abecassis for her feedback when I was writing my thesis.

I would like to thank all the members of the Computer Networks Lab for providing an inviting and enriching environment. I thank them for their kindness and motivation throughout this journey at McGill University.

Lastly, I would like to thank the administration of the Electrical and Computer Engineering department for the guidance they provided during all my stay at McGill University.

Contents

List of Figures

List of Tables

1 Introduction

Preventing a disease with early intervention rather than waiting for a diagnosis and then performing a treatment after is the modern approach to healthcare. In some areas like cardiovascular disease or neurosurgery, computer-based tools are already being used by doctors to improve prediction. Improving disease outcome prediction can greatly aid in the strategic deployment of secondary prevention approaches. Secondary prevention tries to halt or slow the progress of a disease for people that are already sick but only on the early stages of the disease.

1.1 Motivation

Prediction of disease outcomes can be challenging for a doctor as the reasons for a disease may not be well-known and trying to see an evolution might mean regular check-ups which are expensive, in terms of both time and monetary cost. Traditionally, a risk calculator is used to assess the possibility of disease development. It is based on fundamental information (demographics or medical conditions for example). Risk calculators are created based on statistical analysis of clinical data. However, these risk calculators do not perform well and have a low accuracy [1], [2]. An example of such a calculator is given in the Framingham study [1] where a risk calculator is developed for long-term cardiovascular disease. The accuracy of the prediction for hospitalization is only 56% [2]. These models with a low accuracy are not helpful for disease outcome prediction. In recent years, there have been intensive efforts to develop and apply machine learning methods to predict disease outcomes. Compared to traditional approaches, machine learning methods use a large number of variables which help them in improving results. In order to develop such models, data is required to train the model. In the case of cardiovascular disease, Dai et al. [3] achieve an accuracy of 82% for the same false alarm of 30% on the task from the Framingham study with a machine learning algorithm and with more medical factors, improving by 26% the results obtained with a traditional approach. Moreover, another study from the Francis Crick Institute [4] also demonstrates that a machine learning model performs better than models designed

by medical experts at predicting risk of death in patients with heart disease. Machine learning helps in increasing the accuracy of these models but also allows us to identify new important variables for the prediction that doctors had not considered. Machine learning techniques were also applied to neurosurgical outcome prediction [5]. In [5], Senders et al. review thirty studies that used machine learning algorithms for an outcome prediction task after neurosurgery. Some algorithms achieved an accuracy 15% higher than the one obtained with logistic regression [6]–[9]. Some other studies reviewed in [5] show that machine learning methods outperform established prognostic indices [10], [11] and clinical experts [12], [13].

We focus on the prediction of Alzheimer's disease but the work presented here can be applied to other diseases. Alzheimer's disease (AD) is an irreversible disease which destroys brain cells and according to Alzheimer's Disease International [14], someone develops dementia in the world every three seconds. The number of people living with dementia in 2015 is estimated to be 46.8 million and is expected to double every 20 years, reaching 75 million in 2030. Several tools exist to determine if a person with memory problems has possible AD. For example, questions about overall health can be asked to a family member, memory, neuropsychological or standard medical tests can be conducted or brain scans can be performed [15]. Tests should be conducted every 6 to 12 months for people with memory problems. However, the diagnosis is uncertain. Indeed, AD can only be definitively diagnosed after death by linking clinical measures with an examination of brain tissues in an autopsy. Early and accurate diagnosis is important as an early treatment in the disease process may help improve the quality of life of patients for some time even though no cure is available for AD. Several medications for memory decline, changes in language, thinking ability and motor skills exist.

Mild Cognitive Impairment (MCI) is a clinical diagnosis that represents a potential intermediate stage between normal stage and dementia. The tests to diagnose MCI are similar to those used for AD [16]. Approximately 15 to 20% of people that are 65 or older have MCI. Patients with MCI are in a stage where the disease could evolve to AD or not. Predicting the conversion from MCI to AD is very important as knowing the probable progression of the disease early can greatly aid in the strategic deployment of secondary prevention approaches. Thus, applying machine learning models to the prediction of conversion from MCI to AD may help to potentially detect patterns that are not obvious to a doctor. The goal of our work is to develop methods to predict the evolution of a patient from MCI to AD based on multi-modal data from an array of medical examinations and scans.

1.2 Thesis Organization and Contributions

Below we describe the organization of the thesis and summarize the main technical contributions.

- *Chapter 2 - Background*
  We present the background material required for this thesis. We start with a review of statistical methods for longitudinal data based on generalized linear models. Then, we describe the matrix completion problem, on which both of the developed approaches are based, and Graph Convolution Neural Networks, an architecture used in both approaches. Lastly, we give medical background to better understand the used dataset for the prediction of conversion from MCI to AD.

- *Chapter 3 - Literature review*
  We first describe statistical methods used for the study of Alzheimer's disease. Then, we develop the machine learning and graph-based methods applied to the task of prediction of conversion from MCI to AD. Finally, we provide a short comparison of the different type of methods for the disease outcome prediction task.

- *Chapter 4 - Multiple-Graph Recurrent Graph Convolutional Neural Network Architectures for predicting disease outcomes*
  We present the first architecture that we developed for predicting disease outcomes. We pose the problem as a matrix completion problem and solve it with a recurrent graph convolutional neural network, using multiple graphs in order to take into account subject-specific information.

- *Chapter 5 - Multiple-Graph Graph Auto-Encoder architectures for predicting disease outcomes*
  We present the second architecture developed for predicting disease outcomes. Here, we also pose the problem as a matrix completion problem but solve it with a graph auto-encoder strategy.

- *Chapter 6 - Conclusion*
  We provide a summary of the main contributions of the thesis and discuss the outcomes and observed results.

1.2.1 Contributions

- *Chapter 4 - Multiple-Graph Recurrent Graph Convolutional Neural Network Architectures for predicting disease outcomes*

Prof. Mark Coates provided guidance with the research plan and experimental procedure. I designed the architecture and introduced a novel technique to take into account multiple graphs. I conducted the experiments to test the architecture developed.

- *Chapter 5 - Multiple-Graph Graph Auto-Encoder architectures for predicting disease outcomes*
  Prof. Mark Coates provided guidance with the research plan and experimental procedure. I designed the architecture and introduced significant changes to the auto-encoder method to adapt it to our problem. I conducted the experiments to test the architecture developed.

1.2.2 Publications

- *Chapter 4 - Multiple-Graph Recurrent Graph Convolutional Neural Network Architectures for predicting disease outcomes*
  J. Valenchon and M. Coates, "Multiple-Graph Recurrent Graph Convolutional Neural Network Architectures for predicting disease outcomes", to appear in *Proc. 2019 IEEE Int. Conf. Acoustics, Speech and Signal Process. (ICASSP)*, Brighton, UK, May 2019.
  This paper relates to the work described in Chapter 4.

2 Background Material

This chapter provides a summary of background material relevant to this thesis, drawing on literature from the statistical and machine learning fields as well as the medical domain. Although the data analysis techniques presented in this thesis derive from a machine learning perspective, it is important to discuss statistical methods.

Section 2.1 presents an overview of statistical techniques for performing longitudinal data analysis, with a focus on the generalized linear model. We focus on longitudinal data because of the nature of the prediction task. In Chapter 4 and 5, we formulate the disease outcome prediction task as a matrix completion task. With this in mind, Section 2.2 describes the matrix completion problem, focusing on geometric approaches. Section 2.3 provides background material on graph convolutional neural networks. Section 2.4 introduces background material on the different imaging modalities used to generate the dataset we analyze to attempt to predict the onset of Alzheimer's disease.

2.1 Statistical methods for longitudinal data analysis

In developing methods for analyzing longitudinal data and predicting disease outcomes, we can consider statistical approaches [17]. One-way analysis of variance (ANOVA) and multivariate ANOVA (MANOVA) can be used to compare group means. For example, one can assess whether there is a significant difference in the means of a particular risk factor for the group that does not progress to a disease versus the group that does. These techniques have limitations, however. In particular, they struggle in the face of missing or irregularly-timed data, usually requiring undesirable data imputation. More importantly, the ANOVA/MANOVA models do not permit incorporation of time-varying predictors, which usually play an essential role in capturing disease dynamics.

Extensions of Generalized Linear Models provide a more flexible framework. In the subsequent sections we review the definition of a generalized linear model and then describe extensions that have been introduced for longitudinal data analysis.

2.1.1 Generalized Linear Models

A generalized linear model (GLM) is a regression model for independent responses that can be either discrete or continuous. Let $Y_i \in \mathbb{R}$ be the i^{th} response and $x_i \in \mathbb{R}^p$ the $p \times 1$ vector of explanatory variables (covariates) for the i^{th} response. The index i ranges from 1 to K. The goal is to describe the dependence of the mean response $\mu_i = \mathbb{E}(Y_i) \in \mathbb{R}$ on the covariates. A GLM is described by Eqs. (2.1) and (2.2). In these expressions, $g : \mathbb{R} \to \mathbb{R}$ is a known link function and $\beta \in \mathbb{R}^p$ is the model parameter to be inferred. The known variance function $v : \mathbb{R} \to \mathbb{R}$ describes how the variance $var : \mathbb{R} \to \mathbb{R}$ depends on the mean, and $\phi \in \mathbb{R}$ is a constant dispersion parameter.

$$g(\mu_i) = x_i^T \beta. \tag{2.1}$$

$$var(Y_i) = v(\mu)\phi. \tag{2.2}$$

The link function $g : \mathbb{R} \to \mathbb{R}$ and the variance function $v : \mathbb{R} \to \mathbb{R}$ can be any function such as the identity, logit or log functions. Many methods have been developed to infer β [18].

2.1.2 Extensions of GLMs to longitudinal data

GLMs cannot be applied directly to longitudinal data because the responses are correlated, thus violating one of the core model assumptions. There are three prominent extensions of GLMs that allow application to longitudinal data analysis [19]. These include marginal models, transition (Markov) models and mixed-effects models. The difference between these three approaches lies in how correlation is modelled.

In marginal models, regression and within-subject correlation are modeled separately. In transition models and in mixed-effects models, they are modeled jointly. In order to estimate the parameters for a marginal or a transition model, the Generalized Estimating Equation (GEE), a partial likelihood method, is commonly used. This approach involves fewer nuisance parameters than a full likelihood method. A full likelihood approach is normally adopted for mixed-effects models.

Below we briefly review the three different models for statistical longitudinal data analysis. More detail is provided for mixed-effects models because, among statistical methods, the mixed-effects model most flexibly accommodates the challenges of neurodegenerative diseases and associated longitudinal data [17]. It is preferred by the US Food and Drug Administration (FDA) for observational and clinical studies.

The summary provided here is based on material in [19]. In this subsection, longitudinal data is characterized by an $n_i \times 1$ vector of repeated responses for the i-th subject Y_i. The index i ranges from 1 to K and the vector can be expressed as $Y_i = (Y_{i1}, Y_{i2}, \ldots, Y_{in_i})^T \in \mathbb{R}^{n_i}$. At time t, one observes the response $Y_{it} \in \mathbb{R}$ and a $p \times 1$ vector of covariates $x_{it} \in \mathbb{R}^p$.

Marginal models

A marginal model assumes that the marginal expectation of Y_{it}, denoted $\mu_{it} = \mathbb{E}(Y_{it})$, is related to x_{it} by $g(\mu_{it}) = x_{it}^T\beta$ where $g : \mathbb{R} \to \mathbb{R}$ is a known link function and $\beta \in \mathbb{R}^p$ the model parameter to be inferred and that the marginal variance is linked to the marginal expectation by $\text{var}(Y_{it}) = v(\mu_{it})\phi$ where $v : \mathbb{R} \to \mathbb{R}$ is a known function and $\phi \in \mathbb{R}$ a constant dispersion parameter as in the GLM model. Compared to the standard GLM, a key change is in the model of the covariance matrix because Y_{is} and Y_{it} are correlated. The covariance is parametrized as $\text{cov}(Y_{is}, Y_{it}) = c(\mu_{is}, \mu_{it}, \alpha)$ where $c : \mathbb{R}^3 \to \mathbb{R}$ is considered to be a known function and $\alpha \in \mathbb{R}$ an additional parameter.

An interpretation at the population level is given by marginal regression coefficients. $\beta \in \mathbb{R}^p$ describes the effects of the covariates on the marginal expectation of the Y variables.

Transition (Markov) models

Compared to marginal models, transition models try to address both the regression objective and the within-subject correlation simultaneously. The hypotheses are not specified on the marginal expectation and covariance but instead on the conditional expectation $\mu_{it}^c = \mathbb{E}(Y_{it}|Y_{it-1}, \ldots, Y_{i1})$ and the conditional variance $\text{var}(Y_{it})^c = \text{var}(Y_{it}|Y_{it-1}, \ldots, Y_{i1})$. The conditional expectation is modeled as

$$g(\mu_{it}^c) = x_{it}^T\beta + \sum_{j=1}^{v} \alpha_j f_j(Y_{it-1}, \ldots, Y_{i1}), \tag{2.3}$$

where $g : \mathbb{R} \to \mathbb{R}$ and $\{f_j : \mathbb{R}^{t-1} \to \mathbb{R}, j = 1, \ldots, v\}$ are known functions and $\beta \in \mathbb{R}^p$ and $\alpha = (\alpha_1, \ldots, \alpha_v)^T \in \mathbb{R}^v$ are the parameters to be inferred. The assumption on the conditional variance of Y_{it} is $\text{var}(Y_{it})^c = v(\mu_{it}^c)\phi$ where $v : \mathbb{R} \to \mathbb{R}$ is a known function and $\phi \in \mathbb{R}$ a constant dispersion parameter as in the GLM model.

Mixed-effects models

Compared to transition models where the regression coefficients are interpretable only at a population level, mixed-effects models attempt to interpret the coefficients at both population and individual levels. This is done by introducing random effects at the subject level $b_i \in \mathbb{R}^q$ $(i = 1, \ldots, K)$ in addition to fixed effects at the population level $\beta \in \mathbb{R}^p$. In general, the covariate vector $z_{it} \in \mathbb{R}^q$ for random effects b_i is chosen to be a subset of the covariate vector $x_{it} \in \mathbb{R}^p$ for fixed effects β.

Gibbons et al. [20], Laird et al. [21] and Davidian [22] present a variety of linear and nonlinear mixed-effects regression models and discuss their application to longitudinal data analysis. One of the simplest models is the linear mixed-effects model from [23] with Eq. (2.4) where $\epsilon_i \in \mathbb{R}^{n_i}$ is an error term; $\epsilon_1, \ldots, \epsilon_K, b_1, \ldots, b_K$ are independent.

$$Y_{it} = \beta^T x_{it} + b_i^T z_{it} + \epsilon_{it}, \quad \epsilon_i = \begin{pmatrix} \epsilon_{i1} \\ \vdots \\ \epsilon_{in_i} \end{pmatrix} \sim \mathcal{N}_{n_i}(0, \Sigma_i), \quad b_i \sim \mathcal{N}_q(0, D). \tag{2.4}$$

In order to move to a matrix formulation of the linear mixed-effects model, let us denote $Y_i = (Y_{i1}, Y_{i2}, \ldots, Y_{in_i})^T \in \mathbb{R}^{n_i}$, $X_i = (x_{i1}^T, \ldots, x_{in_i}^T)^T \in \mathbb{R}^{n_i \times p}$ and $Z_i = (z_{i1}^T, \ldots, z_{in_i}^T)^T \in \mathbb{R}^{n_i \times q}$. Eq. (2.4) directly reads as Eq. (2.5).

$$Y_i = X_i\beta + Z_i b_i + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}_{n_i}(0, \Sigma_i), \quad b_i \sim \mathcal{N}_q(0, D). \tag{2.5}$$

A matrix formulation of the linear mixed-effects model for $Y = (Y_1, Y_2, \ldots, Y_K)^T \in \mathbb{R}^n$ where $n = \sum_{i=1}^K n_i$ is given in Eq. (2.6) where $X = (X_1, X_2, \ldots, X_K)^T \in \mathbb{R}^{n \times p}$, $Z = diag(Z_1, Z_2, \ldots, Z_K) \in \mathbb{R}^{n \times Kq}$, $b = (b_1, \ldots, b_K)^T \in \mathbb{R}^{Kq}$, $\epsilon = (\epsilon_1, \ldots, \epsilon_K) \in \mathbb{R}^n$, $\tilde{D} = diag(D, \ldots, D) \in \mathbb{R}^{Kq \times Kq}$, $R = diag(\Sigma_1, \ldots, \Sigma_K) \in \mathbb{R}^{n \times n}$ and $O_{m \times n}$ is a $m \times n$ matrix only filled with 0.

$$Y = X\beta + Zb + \epsilon, \quad \begin{pmatrix} b \\ \epsilon \end{pmatrix} \sim \mathcal{N}_{Kq+n}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tilde{D} & O_{Kq \times n} \\ O_{n \times Kq} & R \end{pmatrix}\right). \tag{2.6}$$

Davidian presents a Bayesian formulation of both the linear and non-linear mixed-effects models [22]. The Bayesian formulation incorporates a prior on the distribution of β in the form $\mathcal{N}(\beta^*, H)$ and $D^{-1} \sim$ Wishart. β^*, H and the parameters of the Wishart distribution are assumed to be known.

Semiparametric and non-parametric forms of the mixed-effects model are described in Davidian et al. [24] and Quintana et al.[25]. The nonlinear mixed-effects model is generalized to a semiparametric model by allowing f to depend on a completely unspecified function of time and on the elements of β_i. The extension to multivariate responses is also developed in Davidian et al. [24].

For both linear and nonlinear models, the mixed-effects model can be extended to more than two levels. For example, instead of only having a population and a subject level, we can also have a clinic level. This extension is described in Gibbons et al. [20] and Davidian [22].

Model comparison

Of the three models outlined above, the marginal model and mixed-effects model are most commonly used for analysis of disease longitudinal data [17]. Both models allow either time-invariant or time-varying predictors and handle irregularly timed and missing data without the need for explicit imputation. Both provide mechanisms for assessing the regression relationship between covariates and repeated responses. However, the marginal model does not allow one to choose the correlation structure of the repeated responses. Furthermore, hypothesis testing cannot be performed on correlation parameters. Mixed-effects models use random-effects to describe subject-specific trends over time which provides greater flexibility in modelling the correlation structure of the repeated response. The mixed-effects models are more complex and they rely on correct specification of the mean and correlation structure of the repeated responses for valid hypothesis testing conclusions.

2.1.3 Statistical inference methods

We would like to infer the parameters $\beta \in \mathbb{R}^p$ from the marginal model, $\beta \in \mathbb{R}^p$ and $\alpha \in \mathbb{R}^v$ from the transition model and $\beta \in \mathbb{R}^p$ and $b_i \in \mathbb{R}^q$ $(i = 1, \ldots, K)$ from the mixed-effects models. In this section, we describe two of the most common methods for statistical inference — full and partial likelihood methods. We also briefly discuss the Bayesian inference method and recent developments to reduce the computational complexity of inference for linear mixed-effects models.

Partial likelihood methods

The full likelihood method involves many nuisance parameters. For the marginal and the transition model, partial likelihood methods are attractive alternatives. A common strategy is to construct a generalized estimating equation (GEE) and use it to estimate the parameters. A GEE relies on the specification of the first two

moments rather than the full likelihood. Parameter estimates are consistent with GEE as the number of subjects goes to infinity even if the covariance structure of Y_i is incorrectly specified. The equation to estimate β is $S_\beta(\beta, \alpha) = 0$. Here S_β is the quasi-score function defined in Eq. (2.7). The equations to estimate α are $S_\beta(\beta, \alpha) = 0$ and $S_\alpha(\beta, \alpha) = 0$, with S_α defined in Eq. (2.8).

$$S_\beta(\beta, \alpha) = \sum_{i=1}^{K} (\frac{\partial \mu_i}{\partial \beta})^T cov^{-1}(Y_i)(Y_i - \mu_i), \quad \mu_i = (\mu_{i1}, \dots \mu_{in_i})^T. \tag{2.7}$$

$$S_\alpha(\beta, \alpha) = \sum_{i=1}^{K} (\frac{\partial \eta_i}{\partial \alpha})^T (w_i - \eta_i), \tag{2.8}$$

$$w_i = (r_{i1}r_{i2}, r_{i1}r_{i3}, \dots, r_{in_{i-1}}r_{in_i}, r_{i1}^2, \dots, r_{in_i}^2), \tag{2.9}$$

$$r_{ij} = Y_{ij} - \mu_{ij}, \tag{2.10}$$

$$\eta_i = \mathbb{E}[w_i; \beta, \alpha]. \tag{2.11}$$

Moreover, models with GEE are more restrictive in their assumptions regarding missing data than full-likelihood models.

Full likelihood methods

Full likelihood methods are more computationally expensive than partial likelihood methods. On one hand, a partial likelihood method rewrites the likelihood as the product of one term that depends on β and one that depends on α so that the parameters β and α are inferred with two different formulas. On the other hand, a full likelihood method only uses one formula to find the parameters. Full likelihood methods are used to infer parameters of mixed-effects models. Two inference methods are commonly used, one based on maximum likelihood (ML) and the other one on restricted maximum likelihood (ReML).

In the case of linear mixed-effects models for longitudinal data, the matrix form of the problem reads as Eq. (2.6), as described earlier in the linear mixed-effects part. When D and R are known, the standard estimators for β and b are the generalized least squares estimators $\hat{\beta} = (X^T V^{-1} X)^{-1} X^T V^{-1} Y$ and $\hat{b} = \tilde{D} Z^T V^{-1} (Y - X\hat{\beta})$ where $V = R + Z\tilde{D}Z^T$. When D and R are not known, the ML estimators for β and b are obtained by maximizing the log-likelihood corresponding to the marginal density of y for β and D. The variable part of the log-likelihood is stated in Eq. (2.12). The ML estimators for the variance components are biased and the ReML method

adds a term in the log-likelihood to correct that bias (Eq. (2.13)).

$$l_{ML}(\beta, D|Y) = -\frac{1}{2}\log|V| - \frac{1}{2}(Y - X\beta)^T V^{-1}(Y - X\beta). \tag{2.12}$$

$$l_{ReML}(\hat{\beta}, D|Y) = -\frac{1}{2}\log|X^T V^{-1} X| + l_{ML}(\hat{\beta}, D|Y). \tag{2.13}$$

The goal is to maximize Eq. (2.12) or Eq. (2.13) in order to obtain estimates for β and D. Two commonly employed methods are the Newton-Raphson (NR) algorithm [26]–[28] and the Expectation-Maximization (EM) algorithm [27]–[29]. The NR algorithm is an algorithm to find the value x such that $f(x) = 0$. Here, as we are trying to maximize the log-likelihood, the equation to solve for the ML method to find an estimate of β is $\frac{\partial l_{ML}(\beta, D|Y)}{\partial \beta} = 0$. We have the same equations for the ReML method and for the other parameters we are trying to estimate. The NR algorithm is an iterative algorithm that updates the estimate of the parameter x at each step via the equation $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$.

Lindstrom et al. [27] present an efficient and computationally simple way to implement the NR algorithm and the EM algorithm with matrix decompositions for linear mixed-effects models. The two algorithms are also described by Gumedze et al. [28]. A likelihood method for nonlinear mixed-effects model is described by Wu et al. [29]. An EM algorithm is used for the likelihood estimation.

Bayesian inference

Davidian [22] describes the Bayesian inference method. It is based on employing sampling methods to approximate the posterior distribution of the parameters given the data. The Gibbs sampler is used to address the computational difficulties involved in the necessary high-dimensional integration for linear and nonlinear mixed-effects models. Bayesian inference for nonlinear mixed-effects models is described by Lachos et al. [30].

Recent developments for inference methods

For linear mixed-effects models, Gao et al. [31] present an approach based on the method of moments that is more effective for problems with two random effects and one fixed effect than a likelihood or Bayes approach. The usual methods scale badly to large datasets, with a cost that grows superlinearly in the sample size. The method proposed in Gao et al. [31] has a cost that scales linearly in the problem size and it achieves similar estimation accuracy. Tan et al. [32] present the computational complexity of the different inference algorithms, focusing on the case of

high-dimensional linear mixed-effects models (where the number of covariates p and the sample size n are related as $p \gg n$). For this setting, EM and NR algorithms are too computationally expensive because of the matrix inversions. For the ReML algorithm, the computational complexity is $O(n^2p)$. Using a method of moments approach it can be reduced to $O(n(p+q)^4)$, with q being the number of random effects [33], but the resultant method has no convergence guarantee. The algorithm proposed by Tan et al. [32] is scalable with sublinear computational complexity in p and is guaranteed to converge. The computational cost is $O(\frac{n^2(k+\log p)\log k}{\epsilon^2})$, where k is the rank of the covariance matrix and ϵ is the target approximation error.

2.2 Matrix completion

The goal of matrix completion is to recover a matrix M of shape $m \times n$ where we only know p entries and $p \ll mn$. Formulated as such, the problem is impossible to solve without additional information. However, in many cases, M is known to be structured and the low-rank or the approximately low-rank approximation can be made. This approximation is employed when the matrix entries only depend upon a small number of factors. For example, in the case of recommender systems, the preferences of a user are primarily related to few factors (age, passion for example) so the approximately low-rank assumption can be made.

The low-rank matrix completion problem is given by Eq. (2.14) where X is the matrix to complete and Ω is the set of known entries m_{ij} in matrix M.

$$\min_X \quad rank(X) \quad s.t. \quad x_{ij} = m_{ij}, \forall (i,j) \in \Omega. \tag{2.14}$$

The low-rank matrix completion problem, when described as a rank minimization problem as in Eq. (2.14), is NP-hard. Exact solutions given by all known algorithms require time doubly exponential in the dimension of the matrix, in theory and in practice.

An alternative to rank minimization is given by Candès and Recht [34]. They replace the rank operator by the nuclear norm. The nuclear norm is a convex function defined as the sum of the singular values of the matrix. Under some assumptions given in [34], Eq. (2.15) has a unique low-rank matrix solution and for most problems, Eq. (2.15) is equivalent to Eq. (2.14).

$$\min_X \quad ||X||_* \quad s.t. \quad x_{ij} = m_{ij}, \forall (i,j) \in \Omega. \tag{2.15}$$

If we want the problem to be more robust to noise, the equality constraint of Eq. (2.15) is replaced by a penalty in the objective function. Eq. (2.16) highlights the new form of the problem where $\circ$ is the element-wise Hadamard product, $||.||_F$ is the Frobenius norm, $||A||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$ for an $m \times n$ matrix A and μ controls the balance between fidelity to known values and minimization of the rank.

$$\min_X \quad ||X||_* + \frac{\mu}{2}||\Omega \circ (X - M)||_F^2. \tag{2.16}$$

For semidefinite matrices, the nuclear norm is equivalent to the trace, so Eq. (2.15) can be rewritten as Eq. (2.17).

$$\min_X \quad trace(X) \quad s.t. \quad x_{ij} = m_{ij}, \forall (i,j) \in \Omega. \tag{2.17}$$

Eq. (2.17) can be extended to all kinds of matrices by:

$$\min_{X,W_1,W_2} \quad trace(W_1) + trace(W_2) \quad s.t. \quad x_{ij} = m_{ij}, \forall (i,j) \in \Omega, \tag{2.18}$$

$$\begin{bmatrix} W_1 & X \\ X^* & W_2 \end{bmatrix} \geq 0. \tag{2.19}$$

Another alternative to rank minimization is geometric matrix completion. In this approach, graphs are used to encode relationships between rows and columns. All graphs are assumed to be given as inputs. This translates to the addition of regularization terms in the objective function. This approach was introduced by Kalofolias et al. [35]. Let us assume that the rows and columns of M are given on vertices of graphs. The rows are characterized by the graph $\mathcal{G}_r$ and the columns by the graph $\mathcal{G}_c$. $\mathcal{G}_r = (V_r, E_r, A_r)$ and $\mathcal{G}_c = (V_c, E_c, A_c)$ are undirected weighted graphs where $V_r = \{1, \ldots, m\}$ and $V_c = \{1, \ldots, n\}$ are the vertices and E_r and E_c the edges weighted with non-negative values represented by adjacency matrices A_r (size $m \times m$) and A_c (size $n \times n$). In order to take into account the graph structures in the matrix completion task, two smoothness terms are added to the objective function, one per graph.

$$\min_X \quad \frac{\mu}{2}||X||_* + ||\Omega \circ (X - M)||_F^2 + \frac{\mu_r}{2}||X||_{\mathcal{G}_r}^2 + \frac{\mu_c}{2}||X||_{\mathcal{G}_c}^2. \tag{2.20}$$

We denote by $||X||_{\mathcal{G}} = \text{trace}(X^T \Delta X)$ the Dirichlet norm with respect to a graph $\mathcal{G}$ represented by adjacency matrix A, degree matrix D with $D_{ii} = \sum_j A_{ij}$ and graph Laplacian $\Delta = I - D^{-1/2} A D^{-1/2}$, for identity matrix I. μ, μ_r and μ_c control the balance between fidelity to known values and smoothness with respect to the graphs.

It is also possible to only take into account the graphs and remove the nuclear norm term as tested for recommender systems by Kalofolias et al. [35] and theoretically written by Monti et al. [36]. In that case, the alternative to rank minimization is given by Eq. (2.21).

$$\min_X \quad \frac{\mu}{2}||\Omega \circ (X - M)||_F^2 + ||X||_{\mathcal{G}_r}^2 + ||X||_{\mathcal{G}_c}^2. \tag{2.21}$$

All the alternatives to rank minimization given previously by Eqs. (2.15), (2.16), (2.17), (2.19), (2.20) and (2.21) are convex optimization problems so a unique robust solution exists. A solution for this problem is to use the factorized form of $X = WH^T$ where W and H are respectively $m \times r$ and $n \times r$ matrices with $r \ll min(m, n)$. By construction, $rank(X) \leq r$ and this goes with the low-rank assumption. This approach is described in Monti et al. [36] and Ramlatchan et al. [37].

2.2.1 Examples of architecture to solve a matrix completion task

We review here two architectures developed to solve the matrix completion task for recommender systems. The first one is developed by Monti et al. [36] and the second one by Van Den Berg et al. [38].

Separable Recurrent Graph Convolutional Neural Network

One solution to solve the minimization problem described in Eq. (2.21) is given in Monti et al. [36] where Monti et al. proposed a method named separable Recurrent Graph Convolutional Neural Network (sRGCNN). This method combines a graph convolutional neural network (GCNN) and a recurrent neural network (RNN) to construct a graph diffusion process to identify a solution of the geometric matrix completion problem. Background material concerning GCNNs is provided in Section 2.3 and the GCNN used by Monti et al. is the one from Defferrard et al. [39] with the Chebyshev decomposition. The second part of the architecture involves an RNN and more specifically an LSTM. This part of the architecture helps for the diffusion process and computes an update of the matrix to complete. RNNs are networks with loops in them so information can persist. LSTMs are capable of learning long-term dependencies, something RNNs are not able to do in practice. The architecture is described by the formulas given in Eqs. (2.22), (2.23), (2.24) and (2.25). W_f, W_i, W_o and W_c are weight matrices updated with backpropagation. b_f, b_i, b_o and b_c are bias vectors updated with backpropagation. x_t is the input, h_t is the ouput

and h_{t-1} is the output at the previous time step. σ and tanh are non linearities. $[a, b]$ represents the concatenation of a and b.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \quad i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \quad o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{2.22}$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c), \tag{2.23}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t, \tag{2.24}$$

$$h_t = o_t \circ \tanh(C_t). \tag{2.25}$$

The matrix completion procedure in Monti et al. [36] involves initialization of a matrix X_0 and then iterative training of (i) a graph CNN to perform an embedding of X_t; and (ii) a recurrent neural network that processes the embedding to calculate an update $\delta\widetilde{X}$ to obtain $X_t = X_{t-1} + \delta X_{t-1}$. The parameters of the graph CNN and the RNN are trained to minimize a loss function of the form Eq. (2.21).

Graph Convolutional Matrix Completion

Another solution for a matrix completion task is developed by Van den Berg et al. [38] for recommender systems. The Graph Convolutional Matrix Completion (GC-MC) approach does not use graphs as a regularizer, which is different from all the previously introduced approaches. The goal is to complete a matrix M (size $m \times n$) composed of ratings of n items from m users belonging to a discrete set $\mathcal{R} = \{1, \ldots, R\}$, R being the maximal rating value. Each row of the matrix is a different user and each column a different item. A missing value is encoded by 0. Instead of using only one graph where each node represents a user, a bipartite graph between the users and the items is used. The rating value for one item given by one user is used on the bipartite graph edge between this item and this user. The architecture used to complete the matrix M is a graph auto-encoder. The problem is reinterpreted as a link prediction task. R adjacency matrices M_r of size $m \times n$ are created, one for each rating. M_r contains 1 for its element where the original rating is r. The encoder is a function $[U, V] = f(M, M_1, ...M_R)$ and the decoder is $\tilde{M} = g(U, V)$. The goal is to minimize the reconstruction error between M and $\tilde{M}$. The root-mean square error or the cross entropy can be used for the loss function.

The structure used for the encoder in Van den Berg et al. [38] is a graph convolutional encoder. It uses a message passing algorithm to construct an embedding for each user and for each item. A different embedding is derived for each rating level. Let us consider item j and user i. The message passing step for the message from

item j to user i is Eq. (2.26). c_{ij} is a normalization constant ($|N_i|$ or $\sqrt{|N_i||N_j|}$, N_i being the set of neighbors of node i). W_r is an edge-type specific parameter matrix and x_j is the initial feature vector of node j. The formula is the same for the message passing step from user to item. After obtaining all the messages for user i, the messages are summed in order to obtain a first user embedding h_i (Eq. (2.27)) that leads to the final user embedding u_i (Eq. (2.28)) where W is a weight matrix updated by backpropagation. *accum* is an accumulation operation (concatenation of vectors or sum). σ is an element-wise activation function such as ReLU. The same procedure is repeated for the item embedding v_j.

$$\mu_{j \to i,r} = \frac{1}{c_{ij}} W_r x_j, \tag{2.26}$$

$$h_i = \sigma[accum(\sum_{j \in N_{i,1}} \mu_{j \to i,1}, ..., \sum_{j \in N_{i,R}} \mu_{j \to i,R})], \tag{2.27}$$

$$u_i = \sigma(W h_i). \tag{2.28}$$

A bilinear decoder is used when we have the user and item embeddings. The decoder treats each rating class as separate. The formulas for the decoder are Eqs. (2.29) and (2.30) where Q_r ($r \in R$) are weight matrices, each one specific to one rating r, updated by backpropagation.

$$p(\tilde{M}_{ij} = r) = \frac{\exp(u_i^T Q_r v_j)}{\sum_{s \in R} \exp(u_i^T Q_s v_j)}, \tag{2.29}$$

$$\tilde{M}_{ij} = g(u_i, v_j) = \mathbb{E}_{p(\tilde{M}_{ij}=r)}[r] = \sum_{r \in R} r p(\tilde{M}_{ij} = r). \tag{2.30}$$

2.3 Graph Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have proven to provide significant improvements in solving machine learning problems where the underlying structure of the data can be represented by a low-dimensional regular grid such as images [40],

speech [41] or video [42]. However, many datasets without this underlying structure exist and CNNs cannot be applied to non-Euclidean or high-dimensional irregular domains represented by graphs. Recently, novel techniques for applying convolutional neural networks to graph-structured data have emerged. Spectral and spatial approaches have been proposed for applying CNNs to graph-based data.

2.3.1 Spectral approaches

Spectral graph theory proposes mathematical tools to study graphs [43], [44]. The first model developed by Bruna et al. [45] generalizes convolutions to graphs by creating spectral filters based on the graph Laplacian. This approach is also used by Henaff et al. [46] and extended to large-scale classification problems. Defferrard et al. [39] also use the same approach but propose spectral filters that are strictly localized in K hops from the central vertex. Moreover, a reduction of the computational complexity is obtained by using the Chebyshev decomposition. A simplification is proposed by Kipf and Welling [47] with their Graph Convolutional Network (GCN) where the convolution is defined by only taking the first-order approximation (only one hop) of the localized spectral filters defined in Defferrard et al. [39]. Another generalization of convolution to the spectral domain is done by Levie et al. [48] with Cayley polynomials. The spectral filters computed with this architecture focus on narrow-band frequencies. However, all these spectral graph CNNs are designed for undirected graphs. Monti et al. [49] introduce MotifNet, an extension of graph CNNs to directed graphs by using local graph motifs.

Many tasks are being solved with the spectral formulation of graph CNNs. Sukhbaatar et al. [50] use one of the simplest formulations of a graph CNN to learn the communication between multiple agents to solve multiple tasks like traffic control. Marcheggiani et al. [51] extend the GCN from Kipf and Welling [47] for syntactic tasks by adding gates that allow the model to decide which edges are more relevant to the task in question. This architecture is created for semantic role labeling (finding the structure of a sentence). Both of these architectures [50], [51] are used by Bresson et al. [52] to create a Gated Graph CNN suitable for a graph of arbitrary size. Anirudh et al. [53] perform classification of Autism Spectrum Disorder by using a bootstrapped version of Kipf and Welling's [47] graph CNNs where the nodes are functions of imaging features and the graph is build from the non-imaging features. They use a randomized ensemble of population graphs which are used to create features from graph CNNs and then a consensus strategy is used

for classification.

Spectral Graph Convolution

Most of the spectral approaches are based on the graph Fourier transform so we review the theory behind the generalization of CNNs to data on graphs [36], [39], [46]–[48], [53]. The graph analogue of the Fourier domain is the spectral decomposition of the discrete graph Laplacian.

We would like to process signals defined on an undirected weighted graph $\mathcal{G} = (V, E, A)$ where $V = \{1, \dots, n\}$ are the vertices and E the edges with weights represented by adjacency matrix $A \in \mathbb{R}^{n \times n}$. Using the adjacency matrix A, we calculated a diagonal degree matrix D with entries $D_{ii} = \sum_j A_{ij}$ and a graph Laplacian $\Delta = I - D^{-1/2} A D^{-1/2}$, for identity matrix I. The graph Laplacian Δ is a real symmetric positive semidefinite matrix so by the spectral theorem, there exists a complete set of orthonormal eigenvectors $\{u_l\}_{l=0}^{n-1}$, the graph Fourier modes, associated to real nonnegative eigenvalues $\{\lambda_l\}_{l=0}^{n-1}$, the frequencies of the graph. By defining $U = [u_1, \dots, u_{n-1}] \in \mathbb{R}^{n \times n}$ and $\Lambda = diag([\lambda_0, \dots, \lambda_{n-1}]) \in \mathbb{R}^{n \times n}$, we have $\Delta = U \Lambda U^T$. A graph Fourier transform of $x \in \mathbb{R}^n$ is defined as $\hat{x} = U^T x \in \mathbb{R}^n$.

The convolution operator of graph $*_{\mathcal{G}}$ is not defined in the vertex domain because there is no meaningful translation operator in this domain. It is defined in the Fourier domain as $x *_{\mathcal{G}} y = U((U^T x) \circ (U^T y))$. A signal x is filtered by g_θ as $y = g_\theta(\Delta)x = g_\theta(U \Lambda U^T)x = U g_\theta(\Lambda) U^T x$. A non-parametric filter would be defined as $g_\theta(\Lambda) = diag(\theta)$ where $\theta \in \mathbb{R}^n$ is a vector of Fourier coefficients. However, non-parametric filters are not localized in space and the learning complexity is in $O(n)$, the dimensionality of the data. These issues can be overcome by using a polynomial filter $g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k$ where $\theta \in \mathbb{R}^K$ is a vector of polynomial coefficients. In that case, spectral filters represented by a K^{th} order polynomial of the Laplacian are K-localized and the learning complexity is $O(K)$, similarly to CNNs. However, even with polynomial filters, evaluating the expression $U g_\theta(\Lambda) U^T x$ still takes $O(n^2)$ operations. Defferrard et al. [39] propose to decrease the number of operations by parametrizing $g_\theta(\Lambda)$ as a polynomial function that can be computed recursively from Λ. A traditional approach in graph signal processing to approximate kernels is the Chebyshev expansion [54]. The Chebyshev polynomial of degree j is defined as $T_j(\lambda) = 2\lambda T_{j-1}(\lambda) - T_{j-2}(\lambda)$ with $T_1(\lambda) = \lambda$ and $T_0(\lambda) = 1$. A filter can be parametrized as Eq. (2.31) where $\hat{\Lambda} = 2\lambda_n^{-1} \Lambda - I$ is a diagonal matrix of scaled eigenvalues that lie in the interval [-1,1]. Thus, with this filter, the signal

x is filtered by g_θ as $y = g_\theta(\Delta)x = \sum_{k=0}^{K-1} \theta_k T_k(\hat{\Delta})x$ where $\hat{\Delta} = 2\lambda_n^{-1}\Delta - I$ is the rescaled Laplacian such that its eigenvalues are in the interval [-1,1]. By denoting $x_k = T_k(\hat{\Delta})x$, we have the following recurrence relation $x_k = 2\hat{\Delta}x_{k-1} - x_{k-2}$ with $x_0 = x$ and $x_1 = \hat{\Delta}x$. The number of operations is now $O(K|E|)$, compared to $O(n^2)$ previously.

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\hat{\Lambda}). \tag{2.31}$$

Collecting the feature vectors from all nodes in the graph as the rows of a matrix X, the layers of a graph CNN [39], [47] are of the form:

$$H^{(1)} = \sigma(\widetilde{A}_{\mathcal{G}} X W^{(0)}), \tag{2.32}$$

$$H^{(l+1)} = \sigma(\widetilde{A}_{\mathcal{G}} H^{(l)} W^{(l)}). \tag{2.33}$$

Here $W^{(l)}$ are the weights of the neural network at layer l, $H^{(l)}$ are the output features from layer $l-1$, and σ is a non-linear activation function. The matrix $\widetilde{A}_{\mathcal{G}}$ is an operator derived from the observed graph and determines how the output features are mixed across the graph at each layer. In Kipf et al. [47], $\widetilde{A} = D^{-1/2}(I + A)D^{-1/2}$; in Defferrard et al. [39], $\widetilde{A} \approx [T_0(\Delta) \dots T_{K-1}(\Delta)]$ is a learnable operator constructed from a Chebyshev expansion of a K-th order polynomial of the graph Laplacian and here $\widetilde{A}_{\mathcal{G}} H^{(l)} W^{(l)} = (\widetilde{A}^T H^{(l)})^T W^{(l)} \approx \sum_k^{K-1} T_k(\Delta) H^{(l)} W_k^{(l)}$ where $W^{(l)} = [W_0^{(l)} \dots W_{K-1}^{(l)}]^T$ are the weights of the neural network at layer l.

2.3.2 Spatial approaches

Spatial approaches generalize convolution by using the graph's spatial structure. For a node, a spatial convolution is defined as an inner product between the model's parameters and the values associated to spatially close neighbors of this node. However, in the case of graph-structured data, there is no definition of the spatially close neighbors.

In the same paper where Bruna et al. [45] present a spectral approach of graph CNNs, they also propose a spatial generalization of CNNs to graph-structured data by using multi-scale clustering to define the network architecture. The layers of the network are defined via hierarchical clustering of the node set. The convolution is defined per cluster and the architecture does not use weight sharing.

Another generalization of convolution to graph-based data with a spatial approach is presented by Atwood et al. [55] with their diffusion convolution neural networks (DCNNs). They build a latent representation by scanning a diffusion process across each node. DCNNs can perform node, graph or edge classification. For

the entity that is being classified, the first step is to build a diffusion-convolutional representation by performing random walks on the graph to select spatially close neighbors. The diffusion-convolutional representation is an $H \times F$ real matrix defined by H hops of graph diffusion over F features. The graph diffusion process is defined by the transition matrix P_t that gives the probability of jumping from node i to node j in one hop. P_t^k gives the probability of jumping from node i to node j in k hops. The first step is to perform $P_t^* X_t$, the multiplication of the H power of P_t ($P_t^* = [P_t, \ldots, P_t^H]^T$) by the feature matrix X_t. In order to classify one node, the diffusion-convolutional representation is defined by the rows of this matrix $P_t^* X_t$ associated with the node to classify. In the case of graph classification, the diffusion-convolutional representation is the mean over the nodes of the matrix $P_t^* X_t$. The diffusion-convolutional representation is associated to an $H \times F$ weight matrix. The convolution is realized by doing an inner product between the weight matrix and the diffusion-convolutional representation. This approach is computationally expensive.

To overcome the computational expense of the model of Atwood et al. [55], Hechtlinger et al. [56] propose another generalization of CNNs that is computationally efficient and effective. Similarly to the convolution proposed by Atwood et al. [55], the convolution is also created with random walks on a graph to select the top p closest neighbors for every node. Instead of using the transition matrix, they use another matrix $Q_{ij}^{(K)}$ built from the transition matrix that represents the expected number of visits from node i to node j in K hops. The K^{th} closest neighbor of node i is chosen to be the one that has the largest value in the i^{th} row of $Q^{(K)}$. For each node, the convolution is defined as the inner product between the weights and the node's values of the selected p closest neighbors. The novelty of this convolution is in the handling of different graph structures as its input. This is due to the fact that the neighbors are selected based on their relative position to the node.

In the definition of spatial convolution done by Atwood et al. [55] and Hechtlinger et al. [56], the size of the convolution is fixed. To overcome that issue, Monti et al. [57] propose a mixture model CNN that uses a parametric model to find the closest neighbors. Simonovsky et al. [58] propose a spatial approach to handle graphs of varying size and connectivity. Instead of using diffusion, filter weights are conditioned on edge labels and they are dynamically generated for each input sample. Such et al. [59] propose a new architecture where the filters are polynomial functions of the graph adjacency matrix and they introduce vertex filters to learn features from both edges and vertices. This architecture can be used for both homogeneous and heterogeneous data and for datasets with multiple graphs. Niepert et al. [60] extend CNNs to arbitrary graphs by extracting locally connected regions

from graphs.

However, recent architectures such as [39], [45], [47], [60], [61] do not scale to large graphs or have only been applied in transductive settings where it is required to know the entire graph for training. To overcome these issues, Hamilton et al. [62] propose an extension of GCNs [47] to an inductive setting where it is possible to generate node embeddings for unseen data and to generalize node embeddings across graphs with the same form of features. To do so, Hamilton et al. propose an approach that first uniformly samples a fixed-size neighborhood and then learns several functions that aggregate node feature information such as node degrees by taking into account the node's local neighborhood. Moreover, they also propose a framework to generalize GCNs to use trainable aggregation functions instead of basic convolutions. Based on the same observation that the GCN from Kipf and Welling [47] needs to know the topology and features of the test data during training, Chen et al. [63] interpret GCNs [47] as integral transforms of embedding functions under probability measures. This view of graph convolution allows them to use samples of vertices to compute the loss and the gradient and to reduce computational expense.

Spatial approaches of graph CNNs are also used in numerous applications. One use of spatial convolution is given in Duvenaud et al. [61] for the extraction of molecular fingerprints from molecules with layers which are local filters applied to all nodes and their neighbors. Verma et al. [64] use a spatial formulation of graph CNNs where they map local graph patches and filter weights using the features in the previous network layers for 3D shape analysis. Simonovsky et al. [58] use graph CNNs for point cloud classification. A data-efficient GCN algorithm is developed in Ying et al. [65] where they created a network that can process a graph $10,000\times$ larger than those encountered in typical applications of GCNs. The convolutions are performed by sampling a node's neighborhood and dynamically constructing a computation graph from this neighborhood. The sampling is done by using short random walks. Moreover, random walk similarity measures are used to weight the importance of node features.

2.4 Medical background

A dataset on Alzheimer's disease (AD) was created by the Alzheimer's Disease Neuroimaging Initiative (ADNI) with subjects from North America [66]. It is a

publicly available dataset. This initiative was launched to develop clinical, imaging, genetic and biochemical biomarkers for the early detection and tracking of Alzheimer's disease. ADNI started in 2004 under the leadership of Dr. Michael W. Weiner. It was funded as a private-public partnership with $27 million contributed by 20 companies and two foundations through the Foundation for the National Institutes of Health and $40 million from the National Institute on Aging. The initial five-year study (ADNI-1) was extended by two years in 2009 by a Grand Opportunities grant (ADNI-GO), and in 2011 and 2016 by further competitive renewals of the ADNI-1 grant (ADNI-2, and ADNI-3, respectively).

The TADPOLE dataset, based on the ADNI dataset, is composed of the following modalities: Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), diffusion tensor imaging (DTI) and the cerebrospinal fluid (CSF). Data derived from genetic tests or cognitive test are not included. DTI information is not included as too many values are missing. The information provided here is based on the TADPOLE website [67]. All the images modalities (MRI and PET) are already preprocessed into Regions Of Interest (ROIs) and the measures done on each modality are given in Table 2.1. We will describe in this Section the different medical modalities used in the dataset.

Modality	Measures
MRI	volume, cortical thickness and surface area.
FDG-PET	average measure of cell metabolism.
AV45-PET	average measure of amyloid-beta load in the brain.
AV1451-PET	average measure of tau load in the brain.
CSF	amyloid and tau levels in the CSF.

TABLE 2.1: Different measures of the different modalities in the TADPOLE dataset.

2.4.1 Structural MRI measures

MRI is a 4D modality that allows us to analyze the anatomy and the physiological processes of the brain. Atrophy is an important characteristic to measure using a structural MRI as it can highlight the evolution of AD. Atrophy measures the loss of volume in one region that is caused by the death of neurons in the brain. It can be measured by estimating the quantity of gray matter (GM) and white matter (WM) of the brain. The GM is the brain tissue that consists of nerve cells and the WM defines the fibres connecting the different GM. The brain splits into GM, WM and CSF. Fig. 2.1 shows this splitting for one slice of MRI. Quantification of atrophy

with MRI is a very important biomarker as it is widely available and non-invasive. According to Jack et al. [68], atrophy indicates the progression of MCI to AD for a person because it becomes abnormal in close temporal proximity to the onset of the cognitive impairment.

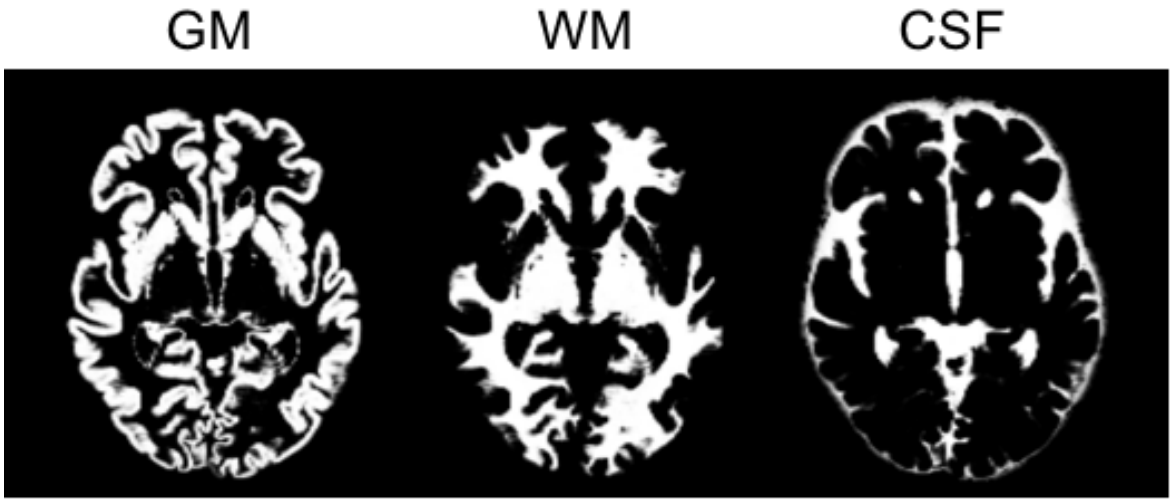


FIGURE 2.1: Parcellation of a slice of a brain MRI into GM, WM and CSF. *Reproduced from [69]*

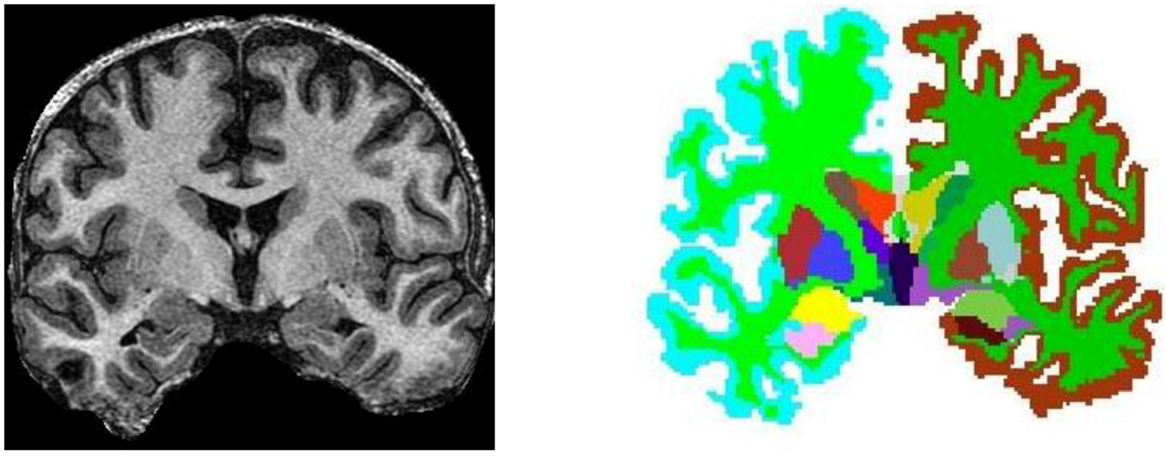


FIGURE 2.2: a) A coronal T1-weighted brain MRI.
b) Brain structures segmented by FreeSurfer.
Reproduced from Mahmoudi et al. [70]

In the TADPOLE dataset, markers of atrophy are measured in three different ways for each Region Of Interest (ROI): volume, cortical thickness and surface area. Here, a ROI is a 3D sub-region of the brain. The brain is subdivided into 130 regions. A total of 346 values of volume, cortical thickness and surface area are available. The preprocessing and segmentation of the MRI is performed using the FreeSurfer

software [71]. The parcellation step is illustrated in Fig. 2.2, where we can only see the subdivision for one slice of the MRI. Fig. 2.3 provides an indication of how the brain is partitioned into ROIs, although this figure shows only a small fraction of the 130 regions.

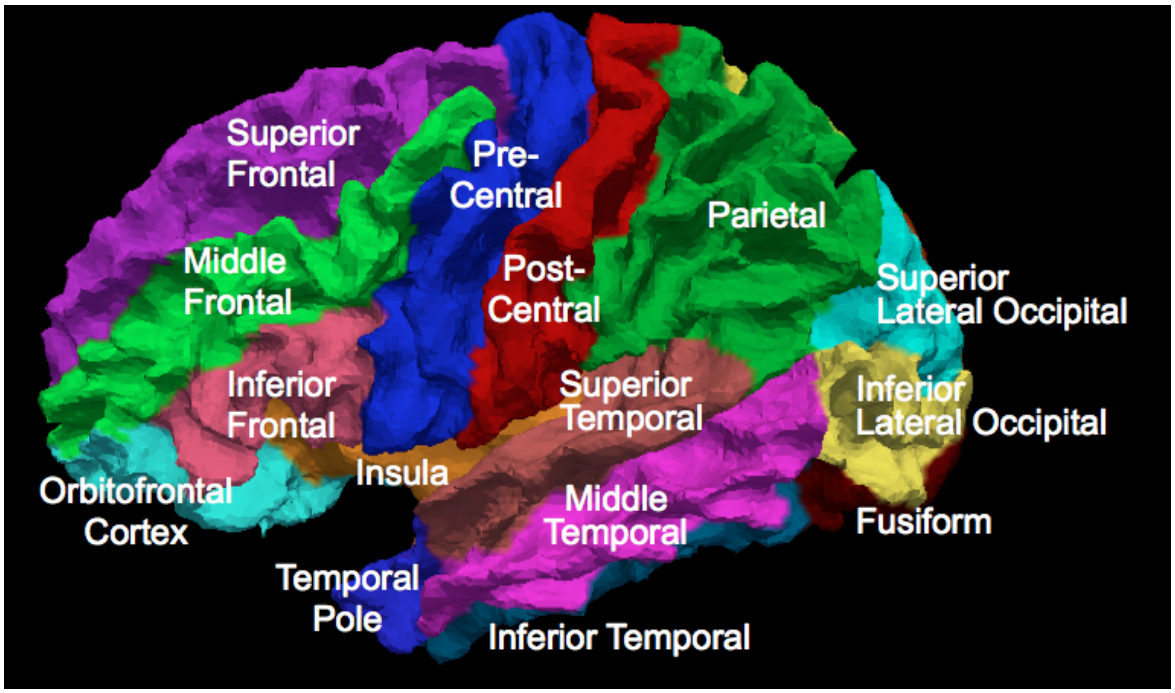


FIGURE 2.3: Different ROIs of the brain labeled with FreeSurfer.
Reproduced from [72]

Two different pipelines can be used in order to derive the atrophy measures from the MRIs: a cross-sectional pipeline and a longitudinal pipeline. The cross-sectional pipeline only uses the data from one visit and each visit is considered as independent. The longitudinal pipeline used the information from all the visits of a subject. The longitudinal measures are more robust, but more values are missing in the TADPOLE dataset.

2.4.2 Positron Emission Tomography (PET)

The PET modality allows to observe metabolic processes in the entire body and in particular, for the study of Alzheimer's disease, in the brain. A radioactive tracer is injected into the region that is under study. Gamma rays are emitted by the tracer and then are detected by the PET system. The tracer is introduced in the body of a molecule to spread throughout the brain. Usually, the molecule with the tracer binds to abnormal proteins (amyloid-beta and tau). We would like to know if someone has these abnormal proteins as their presence may be related to Alzheimer's

disease [73]–[78]. A three-dimensional image illustrating the concentration of the tracer is constructed by computer analysis.

There are three different types of PET scans, depending on the cellular and molecular processes that are being measured:

- *Fluorodeoxyglucose (FDG) PET*: FDG is an analogue of glucose. FDG PET highlights neurodegeneration and can be used to measure cell metabolism. FDG PET images are used in the diagnosis of dementia [79], [80]. FDG PET represents 90% of the PET scans in standard medical care.

- *AV45 PET*: Levels of abnormal proteins such as amyloid-beta can be measured through AV45 PET. Proteins like amyloid-beta need to be properly folded in order to execute their biological function. Accumulation of amyloid-beta in the brain is present for patients with AD [73]. Misfolded amyloid-beta is thought to eventually lead to neurodegeneration and cognitive decline [74]–[76].

- *AV1451 PET*. Levels of abnormal tau proteins can be measured through AV1451 PET. Tau proteins can be abnormally hyperphosphorylated and can accumulate in the neuron's transport system and cause it to degenerate, leading to the neuron's death [77], [78].

An example of PET scans can be seen in Fig. 2.4 where we can observe that the concentration of the proteins depends on age but also on the state of the patient (if the patient has AD or not).

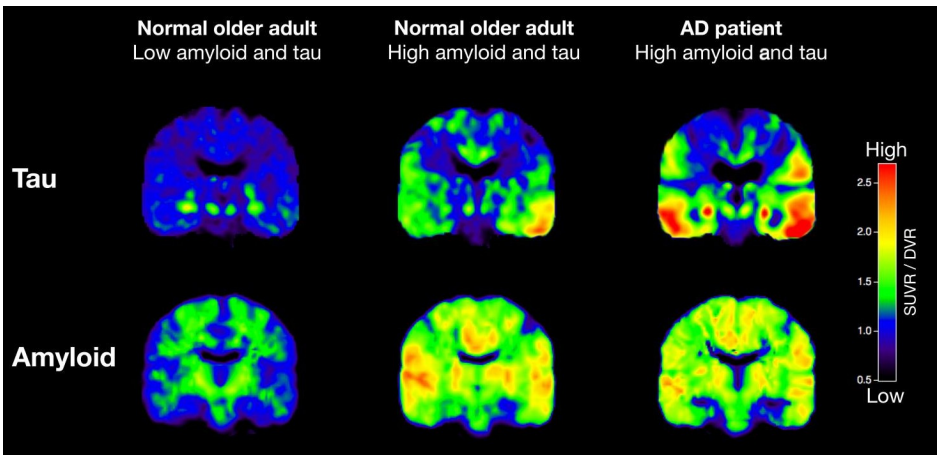


FIGURE 2.4: AV45 and AV1451 PET scans. *Reproduced from [81]*

PET measures are important because they provide information about molecular processes in the brain. These processes are usually the first to become abnormal for someone that can convert to AD. Thus they are important early markers of the

disease. These PET measures might be indicative of whether a healthy control will eventually progress to MCI or not. As was the case for the MRI measures, PET measures are derived for each ROI in the TADPOLE dataset.

PET scans are non-invasive but patients are exposed to ionizing radiation while doing a PET scan, limiting the number of scans possible to take in a specific time interval. PET scans also have a much lower spatial resolution compared to MRI scans. Moreover, the PET scanner is extremely expensive. For AV1451 PET, another disadvantage is its novelty: it is still under research, and few subjects in the TADPOLE dataset have undergone an AV1451 PET scan. In our analysis, the FDG-PET and the AV1451 PET measures are removed from the dataset because there are too many missing values. We only process the measures derived from AV45 PET scans.

2.4.3 Cerebrospinal Fluid (CSF)

The CSF is a liquid that goes around the brain and spinal cord. It acts as a cushion or buffer for the brain and spinal cord, providing basic mechanical and immunological protection to the brain inside the skull. CSF picks up needed supplies from the blood and gets rid of waste products from brain cells. Bacteria and viruses that can attack the brain can be present in the CSF. By taking a sample of the CSF, the doctor can diagnose some illnesses. A sample of the CSF can be taken from patients invasively, by inserting a needle in the spinal cord.

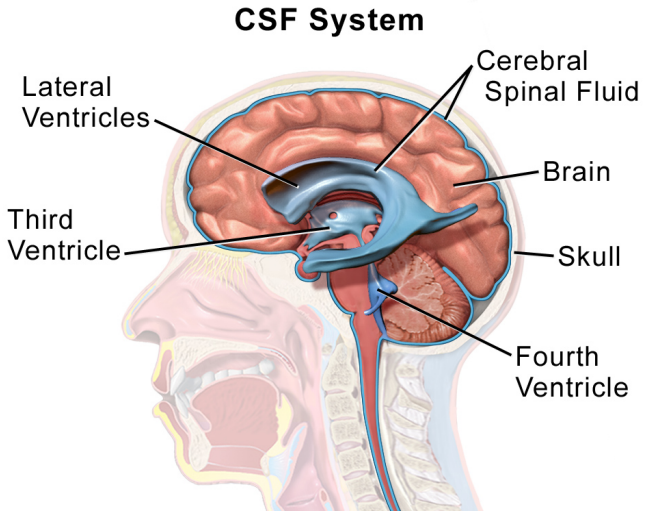


FIGURE 2.5: CSF fluid. *Reproduced from [82]*

Measures of CSF are very important for dementia research. In the CSF, the concentration of abnormal proteins such as amyloid-beta and tau is a strong indicator of Alzheimer's disease [83]–[85]. Abnormal levels of concentrations of these proteins

are some of the earliest signs of Alzheimer's disease and can indicate abnormalities many years before symptom onset.

However, the CSF measures have some limitations. One of the key limitations is that the lumbar puncture is highly invasive and thus not performed in many studies, although a substantial fraction of the ADNI subjects did agree to undergo the procedure. The CSF measures are also not specific to any particular part of the brain and only provide global concentration measures. In the TADPOLE dataset, we have the concentration of three different proteins in the CSF: amyloid-beta, tau and phosphorylated tau.

3 Literature review

Predicting the conversion from MCI to AD is very important as knowing the probable progression of the disease early can greatly aid in the strategic deployment of secondary prevention approaches. In recent years there have been intensive efforts to develop and apply machine learning methods to predict disease outcomes. The learning algorithms can potentially detect patterns that are not obvious to a doctor.

Many algorithms have been applied to data collected for the study of progression of Alzheimer's disease (AD) since helping cure Alzheimer's disease is of worldwide concern. The Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset is a publicly available dataset with data collected for the early detection and tracking of AD. We first look into statistical methods that were used for the study of Alzheimer's disease in Section 3.1. Then, we focus on machine learning techniques in Section 3.2 and graph-based methods in Section 3.3 that were used for the important task of prediction of conversion from MCI to AD. Finally, we do a short comparison of the different type of methods for the disease outcome prediction task in Section 3.4.

3.1 Statistical methods for the study of Alzheimer's disease

Statistical methods for longitudinal datasets are often used to analyze data but less often for classification. Mixed-effects and marginal models have been applied to the study of progression of Alzheimer's disease. We focus on mixed-effects models as they have demonstrated superior performance.

3.1.1 Mixed-effects models

Mixed-effects models have been mainly used to analyze the ADNI data and assess temporal progression of some measurements [86]–[90]. Most studies try to find predictors of the disease. They extract information from different medical measurements (e.g., MRI, PET, CSF) and try to fit a linear mixed-effects model to analyze

the evolution of the information with time as a covariate.

Schiratti et al. [89] analyze the temporal progression of the ADAS13 scores (the 13^{th} item of the Alzheimer's Disease Assessment Scale) for MCI subjects who progressed to AD. This study does not highlight the difference between MCI converters and non converters.

Group differences and parameter evolution are studied by Bernal-Rusiel et al. [86] and Li et al. [88]. Bernal-Rusiel et al. [86] assess group differences between trajectories of two biomarkers (Mean thickness within the entorhinal cortex averaged across hemispheres (ECT) and the total hippocampas volume (HV)). Six fixed-effects are used: time-from-baseline, age, sex and several medical characteristics of the subject (such as the carriage of the APOEϵ4 allele). Li et al. [88] predict the long-term trends of seven outcomes which are modalities of the ADNI dataset known to be related to AD such as CSF tau, CSF amyloid-beta or the MRI volume of hippocampus. They do not try to predict conversion from MCI to AD but they do examine how different biomarkers and parameters evolve with age for healthy and Alzheimer's subjects. The fixed-effect covariates used to explain the difference between the AD subjects and the healthy ones are age, carriage of the APOEϵ4 allele, sex and education.

Contrary to all of the studies described above, Ziegler et al. [87] and Bernal-Rusiel et al. [90] use images and not biomarkers as outcomes. Ziegler et al. [87] use MRI voxels as outcomes to detect group differences. A voxel-wise linear mixed model is used on the gray matter density at a single voxel. Bernal-Rusiel et al. [90] propose a spatial extension of the linear mixed-effects model to study cortical thickness maps. This model is proven to perform better than the voxel-wise linear mixed model. Instead of employing a voxel or vertex-wise approach, the image is split into Regions Of Interest (ROIs) and the temporal covariance structure is assumed to be shared for all the voxels of each ROI. The different voxels are linked by a simple parametric covariance structure. The goal is to find which cortical regions are different between AD and healthy subjects. The fixed-effects from [86] are used in [90].

Several methods have been used to estimate the parameters of the model by computing the likelihood. Bernal-Rusiel et al. [86] either use the EM algorithm or Newton-Raphson based procedures, Ziegler et al. [87] the EM algorithm, Li et al. [88] a Markov Chain Monte Carlo method and Schiratti et al. [89] the Monte Carlo Markov Chain Stochastic Approximation EM.

3.1.2 Marginal models with GEEs

Li et al. [91] and Zhang et al. [92] use a marginal model on MRIs for prediction and region selection. Zhang et al. [92] choose to use a GEE approach because it only requires the first two marginal moments and a working correlation structure for the scalar response variable. By contrast, a mixed-effects model requires specification of a distribution for the parameters, which is a difficult task for a tensor covariate.

3.2 Machine learning methods for the prediciton of conversion from MCI to AD

Machine learning techniques have been widely used for the task of prediction of conversion from MCI to AD [93]–[98]. The input is only formed with data from subjects when they were diagnosed with MCI. We call this baseline data. The feature extraction step is one of the main differences between the state-of-the-art methods: most of the methods use Regions Of Interest (ROIs) which are known to be linked to AD to reduce the dimension of the data. These methods are handcrafted as the features extracted from the MRIs are based on theoretical knowledge about the regions. Another group of methods use automatic procedures to learn useful features from the raw data. We split the methods into two groups: those which process raw data and those which use handcrafted features.

3.2.1 Using raw data

All of the methods that process raw data use Convolutional Neural Networks (CNNs). Korolev et al. [93] use known architectures (VGG [99] and ResNet [100]) and adapt them to 3D images for the input. MRIs are used as the input. An accuracy of only 56% is obtained for VoxCNN (the extension of VGG [99] to 3D images) and 52% for ResNet (extension of ResNet [100] to 3D images). Arco et al. [94] use the searchlight approach [101] to extract the features from the gray matter (GM) and white matter (WM) maps of the MRI. They add two cognitive scores from the ADNI dataset to the features to perform classification with an SVM. They use data from one or two sessions when possible. The accuracy is respectively 84.3% and 82.05% for a prediction six and twelve months ahead. Choi et al. [95] use a deep convolutional neural network, trained with data for the classification task of AD vs healthy controls (HC) in order to predict the conversion from MCI to AD. The two modalities used are FDG and AV-45 PET. Both images are used as inputs of the 3D CNN which then becomes a 4D CNN. Data augmentation (left-right flipped) is used to augment

the training set. An accuracy of 84.2% is obtained for a prediction within 3 years.

3.2.2 Using handcrafted features

Two different methods are proposed with handcrafted features, one based on an auto-encoder and the other one based on a neural network. Suk et al. [97] first perform a handcrafted feature selection by taking the patch-based volume from the GM tissues for the 93 Regions Of Interest (ROIs) which are known to be linked to AD. For each of these regions, the mean intensity from the PET images is also taken as a feature. Then, a stacked auto-encoder is used for feature extraction, followed by sparse regression for feature selection and a multi-kernel SVM for classification. The accuracy is 83.3% for an eighteen-month ahead prediction. Lu et al. [98] use the same handcrafted features as [97] for MRI and FDG-PET images. Then, a multimodal and multiscale deep neural network is used for classification. An accuracy of 82.4% was obtained in identifying the individuals with MCI who will convert to AD at 3 years prior to conversion and a combined accuracy of 86.4% was obtained for conversion within 1 to 3 years.

3.3 Graph-based methods for the prediction of conversion from MCI to AD

Only recently have graph-based learning methods started to appear for disease outcome prediction. Previously, state-of-the-art approaches employed more traditional classification approaches including random forests, support vector machines [94], [102] and convolutional neural networks [95]. Parisot et al. [103] were the first to propose a graph-based learning algorithm for disease outcome prediction [103]. Then, Vivar et al. [96] proposed another graph-based method.

Parisot et al. [103] employed a graph convolutional neural network on MRI data preprocessed in 138 regions of interest where volumes are taken as features. They used 3^{rd} order Chebyshev polynomials in the graph convolutional layer. The system is composed of 5 convolutional layers followed by a ReLU activation layer and then a fully-connected output layer. This output layer is followed by a softmax layer for the classification. The cross-entropy loss is used. They used longitudinal data but two samples from the same person are used as different examples, they are just linked by the adjacency matrix.

Age and sex are used as links for the graph. The formula for the adjacency matrix A for subjects v and w is given in Eq. (3.1) with $M = \{M_h\}$ the set of H non imaging measures (here, age and sex). Sim is a similarity measure between subject v and w. Here, $Sim = \lambda$ with $\lambda > 1$ if two samples are from the same subject, otherwise it is equals to 1. For categorical data (such as sex), $\rho = \delta$, δ being the Kronecker delta. For quantitative data (such as the age), $\rho(M_h(v), M_h(w)) = 1$ if $|M_h(v) - M_h(w)| < \theta$ and $\rho(M_h(v), M_h(w)) = 0$ otherwise. λ and θ are hyperparameters. In [103], Parisot et al. took $\lambda = 10$ and $\theta = 2$. The reported averaged accuracy is 77% and the Area Under the receiver operating characteristic Curve (AUC) is 85%.

$$A(v, w) = Sim(v, w) \sum_{h=1}^{H} \rho(M_h(v), M_h(w)). \tag{3.1}$$

Vivar et al. [96] also propose a graph-based method to predict MCI to AD conversion. Multimodal data composed of numeral values already extracted from MRI, PET, CSF and DTI from the TADPOLE challenge, described in Appendix A, is used. They propose to solve the multi-modal disease classification as a geometric matrix completion problem.

Vivar et al. [96] use the algorithm from [36] to do matrix completion. This built on the work of Thung et al. [104] who used the low-rank matrix completion approach developed in Goldberg et al. [105] for jointly performing imputation of missing values and transductive classification.

The initial approach used by Monti et al. [36] is described in Section 2.2. The goal is to complete the matrix M composed of the concatenation of the $m \times n$ matrix of features X and of the $m \times 1$ vector of labels Y. Each row of M represents one of the m subjects. The first step is a single value decomposition of the matrix M into W (size $m \times r$) and H (size $n \times r$), r being the rank chosen for the decomposition.

A graph on the subjects is used on the rows of W. W is updated with the sRGCNN algorithm described in Algorithm 1. The GCNN layer is the one by Defferrard et al. [39] described in Section 2.3. H has no graph and is only updated by backpropagation.

The loss function is described in Eq. (3.2) where γ_a, γ_b, γ_c, γ_d and γ_e are hyperparameters. Ω_a is the indicator matrix of the known feature values, the concatenation of a $m \times n$ matrix filled with 1 when the feature is available in M and 0 otherwise and a $m \times 1$ vector of 0. The classification term is a binary cross-entropy term $l_{\Omega_b}(Z, M) = -(y \log(p) + (1 - y) \log(1 - p))$, where Ω_b is the indicator matrix of

known outcomes, the concatenation of a $m \times n$ matrix filled with 0 and a $m \times 1$ vector of 1, p is the classification output vector and y the label vector. $||.||_F$ is the Frobenius norm, $||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$ for an $m \times n$ matrix A and the Dirichlet norm with respect to a graph $\mathcal{G}_i$ with adjacency matrix A_i, degree matrix D_i and graph Laplacian $\Delta_i = I - D_i^{-1/2} A_i D_i^{-1/2}$, for identity matrix I is $||X||_{\mathcal{G}_i} = \text{trace}(X^T \Delta_i X)$. The first four terms are related to the matrix completion problem, the first term being a constraint on the graph structure on W, the second and third being constraints on the matrix and the fourth being for the matrix completion problem. The last term is for the classification.

The graph is built with data from the patients (age and sex). Vivar et al. [96] used the graph from [103] for the TADPOLE dataset but removed the similarity measure. The formula for the adjacency matrix A becomes Eq. (3.3). [96] reports an accuracy of 87% and an AUC of 95%.

$$l(\theta) = \frac{\gamma_a}{2}||W||^2_{\mathcal{G}_r} + \frac{\gamma_b}{2}||W||^2_F + \frac{\gamma_c}{2}||H||^2_F + \frac{\gamma_d}{2}||\Omega_a \circ (M - WH^T)||^2_F + \gamma_e(l_{\Omega_b}(M, X)). \tag{3.2}$$

$$A(v, w) = \sum_{h=1}^{H} \rho(M_h(v), M_h(w)). \tag{3.3}$$

Algorithm 1 sRGCNN algorithm modified by Vivar et al. [96]

procedure sRGCNN(W, H, A)
  Initialization
  Initialize weights (Glorot [106]) and biases (zero)
  for k in number iterations **do**
    $\hat{W} = GCNN(W, A)$
    $\delta W = LSTM(\hat{W})$
    $W = W + \delta W$
    $M = WH^T$
  Compute loss function (3.2)
  Backpropagation Update weights, biases and H

3.4 Comparison of these methods

As a conclusion of this section, we compare the different types of techniques in the case of prediction of conversion from MCI to AD. The main reason for retaining a statistical approach is for its ease of understanding and interpreting parameters. However, mixed-effects models are computationally expensive and do not work if there are too many effects. We also need to know the covariance structure and

the distribution of the parameters. The strengths of a mixed-effects model are its handling of imbalanced longitudinal data with missing data-points and imperfect timing. Using a mixed-effects model for our problem would mean using a non-linear model as we want to perform classification. A logistic mixed-effects model could be tried because of the task we want to achieve and it might be more powerful than a linear model as the outcome is categorical. The outcome is the classification decision and the covariates biomarkers or voxels. This implies many effects if we want to take into account all the voxels. No paper using a statistical approach [86]–[92] used voxels of MRIs as covariates. In all of the models which were described before, the number of effects is small (always fewer than ten). Mixed-effects models have not been used previously for the prediction of conversion from MCI to AD.

Compared to statistical methods, machine learning methods have been widely used in order to predict the conversion from MCI to AD. One of the strengths of machine learning is its handling of high-dimensional data. However, it is harder to interpret a machine learning model than a statistical one. The results achieved are promising, especially those reported by Vivar et al [96] (although we have struggled to reproduce these results in our experiments). Using a graph to link the subjects is one of the main strengths of the approach used in Vivar et al. [96] compared to other state-of-the-art methods. Only a few graph-based learning methods [96], [103] have been used for the task of prediction of the conversion from MCI to AD.

4 Multiple-Graph Recurrent Graph Convolutional Neural Network Architectures for predicting disease outcomes

4.1 Introduction

Extracting the most information from medical datasets can greatly aid in the strategic deployment of secondary prevention approaches. Machine learning algorithms can potentially discover patterns that are not obvious to a doctor. The prediction accuracy can be improved by using as much information as possible, including, for example, the age and sex of a subject. These types of subject attributes can impact both the medically-derived features and the disease outcome that is the prediction target. For example, women are more likely to develop Alzheimer's disease (AD) than men [107], [108]. The MRI-derived brain volumes of cortical subregions are potential predictors, and larger values are observed for men [109], [110].

Recently, prediction techniques have been developed based on graph convolutional neural networks (CNNs) and graph-based geometric matrix completion [96], [103]. These methods connect subjects by constructing a single graph based on attributes such as age and sex. Graph-based learning approaches such as those developed in Defferard et al. [39], Kipf et al. [47] and Monti et al. [36] are then employed to process the medical features for each subject and perform the prediction. The geometric matrix completion approach also eliminates the need for imputation of missing features. In both [103] and [96], a single graph is used for all features. In contrast, we develop an architecture that processes multiple graphs; our algorithm associates different features to different graphs by fitting a general linear model (GLM) and assessing the significance of each regression coefficient. Since it builds on the algorithm in Monti et al. [36], our work is related to graph-based matrix completion techniques described in Section 2.2 and graph convolutional neural networks described in Section 2.3. Most of the graph-based algorithms employ a single

graph. Kipf et al. discuss the possibility of using multiple graph [47]; Such et al. and Monti et al. explicitly use multiple graphs [36], [59]. Although multiple graphs are employed, each graph is used to process all features at each node. As a result, the graph neural network must learn an embedding from a higher dimensional feature space using many variables that are unlikely to be related to the graph used for processing.

In general, it is not the case that every feature is dependent on each of the attributes used to construct the graph. For example, for Alzheimer's disease prediction, intracranial volume is dependent on sex but does not vary significantly with age (see Figure 4.3). The major innovation in this chapter is the use of multiple attribute graphs for graph CNN matrix completion. Via a general linear model and statistical significance tests, we identify an appropriate association of specific features to each graph. Our approach is the first to employ multiple feature-specific adjacency matrices for learning using convolutional graph neural networks.

Section 4.2 provides a formal statement of the problem. Section 4.3 provides our approach and algorithm and Section 4.4 presents the results of the application of our approach to the prediction of Alzheimer's disease development.

4.2 Problem statement

We consider the following prediction task for disease outcomes. Let $X \in \mathbb{R}^{m \times n}$ be the feature matrix, m being the number of subjects and n the number of features. The features are assumed to be derived from medical examinations. X may have missing values. Let $Y \in \{0,1\}^{m \times 1}$ be a vector denoting the disease outcomes for the m subjects. Some of these are unknown and these are the focus of the prediction task.

Let $\mathcal{G}_i = \{V_i, E_i, A_i\}$ be a graph on the subjects with edges derived by a similarity metric from a subject attribute s_i. The attribute can be categorical, or real- or integer-valued. V_i denotes the vertices, $V_i = \{1, ..., m\}$, E_i the edges, $E_i \subseteq V_i \times V_i$, and $A_i \in \{0,1\}^{m \times m}$ the adjacency matrix. We assume that there are P such graphs derived from different combinations of subject attributes and thus capturing different relationships between subjects. Taking into account the features X and the relationships formed by the similarities of the attributes s_i and captured by G_i, $i = 1, \ldots, P$, our task is to predict the unknown disease outcomes in Y and impute the missing values in the matrix X.

4.3 Methodology

In the task of disease outcome prediction, most of the datasets have missing values and inaccurate measurements. Formulating the task as matrix completion as in Goldberg et al. [105] allows us to jointly perform transductive classification and imputation of missing values. To do this, we form a matrix $Z = [X, Y]$ and apply a matrix completion algorithm. The initial matrix M contains all the information from the dataset.

We commonly have knowledge of attributes that can be used to identify relationships or similarities between subjects. Attributes such as age and sex often impact the probability of a disease outcome and the likelihood of a feature derived from a medical examination. In trying to recover a matrix with missing values and unknown disease outcomes, it is reasonable to assume that there is smoothness with respect to a graph that connects individuals who share similar attributes (close in age, same sex). Once such a graph has been constructed, geometric graph completion can be performed; Vivar et al. [96] use the algorithm from [36] to do so.

The problem with the approach outlined above is that in general it is not the case that every medically-derived feature is dependent on all of the attributes used to construct the graph. For example, for prediction of progression to Alzheimer's disease, Vivar et al. [96] construct a weighted adjacency matrix that includes an edge between people of the same sex and those of similar age. Many of the features in the matrix have no dependency on age; requiring such features to be smooth with respect to age imposes an undesirable penalty in the optimization and results in incorrect information diffusion throughout the graph. With regard to imputation, if one is estimating a missing value that is sex-dependent, but not age-dependent, it is better to use all of the values from subjects with the same sex and not bias the imputation by processing values from the other sex.

In our proposed approach we construct multiple graphs based on the available attributes and associate a feature with one or more of these graphs by fitting a general linear model (GLM) with the attributes as the independent variables and the features as the dependent variables. We then assess the significance of the regression coefficients. The features with a statistically significant non-zero value for attribute a_i are included in a subset Z_i of Z that is associated with each graph $\mathcal{G}_i$. The GLM is fit using ordinary least squares and we assess significance of coefficients using multiple ANOVA and post-hoc t-tests. In this procedure, controlling the Type I

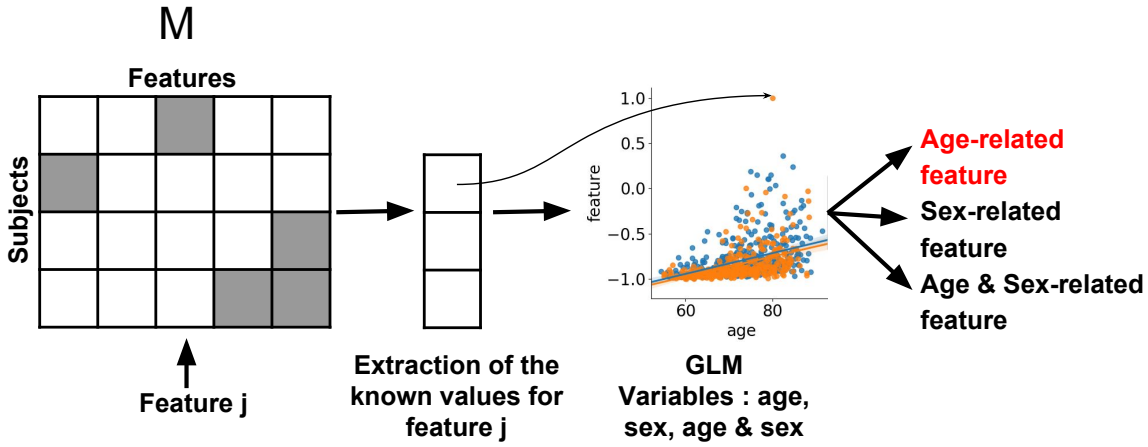


FIGURE 4.1: Process to decide the feature dependence.

error is not as significant a concern as is usually the case in regression procedures. Erroneous association of a feature with a specific attribute graph leads to an additional smoothness penalty that should not be included, but in most cases this has a minor effect on the overall inference procedure. Improvement in prediction outcomes is achieved by ensuring that the majority of features with no dependence on an attribute are excluded from the subset.

4.3.1 Multiple-Graph Recurrent Graph Convolutional Neural Network (MG-RGCNN)

We develop an architecture based on the Recurrent Graph Convolutional Neural Network (RGCNN) from [36]. We adapt it to take into account the multiple graphs and the prediction task. The GCNN layer as described in Section 2.2 computes features from the initial matrix Z using multiple graph convolutional neural networks based on the graphs $\mathcal{G}_i$. Each graph $\mathcal{G}_i$ is associated to some subject's attribute(s) and each subset Z_i of Z is composed of the features that depend on the same subject's attribute(s). Each node of each $\mathcal{G}_i$ is a subject and is represented by the row of Z_i corresponding to that subject. If the graph is associated to a categorical attribute (such as sex), two people are connected if they have the same attribute. If the graph is associated to a quantitative attribute (such as age), two people are connected if they have a close value in the attribute, chosen by a threshold depending on the application. The P different GCNN outputs are concatenated and provided to the recurrent neural network. The algorithm is described in Algorithm 2.

The parameters of the multiple graph CNNs and the RNN are trained using a loss function that has a Dirichlet norm penalty for each graph. The Dirichlet norm

with respect to a graph $\mathcal{G}_i$ with adjacency matrix A_i, degree matrix D_i and graph Laplacian $\Delta_i = I - D_i^{-1/2} A_i D_i^{-1/2}$, for identity matrix I is $||X||_{\mathcal{G}_i} = \text{trace}(X^T \Delta_i X)$. It would also be possible to consider a weighted sum of Dirichlet norms, where each weight is dependent on the number of features associated with the subset. As we are addressing a classification problem in addition to imputation of missing entries, we also add a binary cross-entropy term $l_{\Omega_b}(Z, M) = -(y \log(p) + (1-y)\log(1-p))$, where Ω_b is the indicator matrix of known outcomes, the concatenation of a $m \times n$ matrix filled with 0 and a $m \times 1$ vector of 1, p is the classification output vector and y the label vector. Ω_a is the indicator matrix of the known feature values, the concatenation of a $m \times n$ matrix filled with 1 when the feature is available in M and 0 otherwise and a $m \times 1$ vector of 0, and $||.||_F$ is the Frobenius norm, $||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$ for an $m \times n$ matrix A. We added an l2 regularization term to avoid overfitting. $(W_1, \ldots, W_q)$ represents the q weight matrices used in the architecture. μ_i, μ and γ_{l2} are parameters controlling the balance between the different loss terms.

$$l(\theta) = \sum_{i=1}^{P} \frac{\mu_i}{2} ||Z_i||^2_{\mathcal{G}_i} + \frac{1}{2} ||\Omega_a \circ (Z - M)||^2_F + \mu l_{\Omega_b}(Z, M) + \gamma_{l2} \sum_{i=1}^{q} W_i. \tag{4.1}$$

Algorithm 2 RGCNN with P graphs (MG-RGCNN)

procedure MG-RGCNN($M = [X, Y], \{A_i\}, \{Z_i\}$)
  Initialization
  $Z = [X, Y_{train}]$
  Initialize weights (Glorot [106]) and biases (zero)
  for k in number iterations **do**
    for i in $1, ..., P$ **do**
      $v_i = GCNN(Z_i, A_i)$
    $v_{tot} = \text{concatenation}([v_i], i = 1, \ldots, P)$
    $\delta Z = LSTM(v_{tot})$
    $Z = Z + \delta Z$
  Compute loss function (4.1)
  Backpropagation Update weights and biases

4.3.2 Application to Alzheimer's disease

We apply the proposed MG-RGCNN to the prediction of conversion from Mild Cognitive Impairment (MCI) to Alzheimer's disease (AD). MCI is a clinical diagnosis that represents a potential intermediate stage between normal stage and dementia. Patients with MCI are in a stage where the disease could evolve to AD or not.

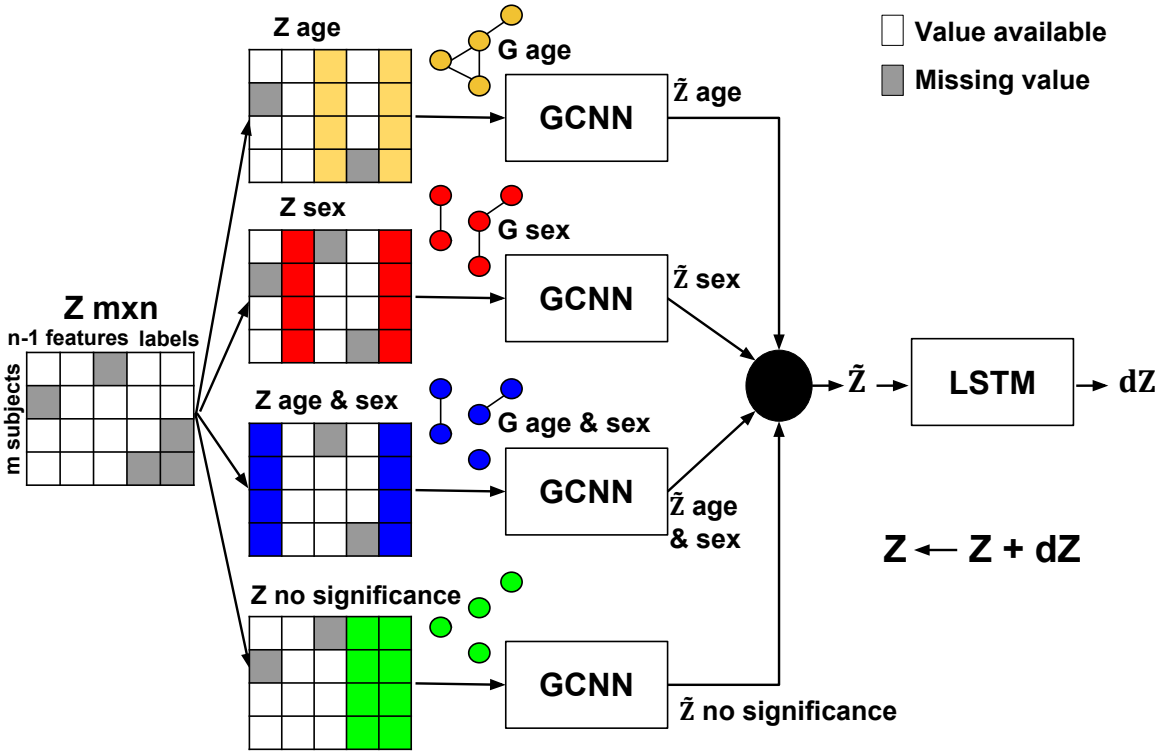


FIGURE 4.2: MG-RGCNN architecture for the application of prediction of conversion from MCI to AD. The initial matrix Z is divided into 4 subsets Z_{age}, Z_{sex}, $Z_{age\&sex}$ and Z_{ns} respectively associated with each graph $\mathcal{G}_{age}$, $\mathcal{G}_{sex}$, $\mathcal{G}_{age\&sex}$ and $\mathcal{G}_{ns}$. The highlighted columns are the columns of features that are associated with the attribute(s) that the graph is built on and that are kept in Z_i.

Predicting the conversion from MCI to AD is very important as knowing the probable progression of the disease early can greatly aid in the strategic deployment of secondary prevention approaches. The architecture is described in Fig. 4.2. We built 4 graphs: $\mathcal{G}_{age}$ for age-related features, $\mathcal{G}_{sex}$ for sex-related features, $\mathcal{G}_{age\&sex}$ for age and sex-related features and $\mathcal{G}_{ns}$ for features that are neither related to age or sex. Each node of each graph represents a subject. At each node of each graph, we have the subject's values of the features that are related to the subject's characteristic(s) associated to the graph.

These four graphs lead to four adjacency matrices A_{age}, A_{sex}, $A_{age\&sex}$ and A_{ns}, A_{ns} being the identity. The age adjacency matrix A_{age} is constructed by including an edge between subject r and s if $|\text{age}(s) - \text{age}(r)| < 2$. The sex adjacency matrix A_{sex} includes an edge if $\text{sex}(s) = \text{sex}(r)$. The age and sex adjacency matrix $A_{age\&sex}$ adds an edge when both conditions are satisfied.

We tried both the simplified graph CNN from [47] (MG-RGCNN GCN) and the graph CNN from [39] (MG-RGCNN GCNN). The GCN from Kipf and Welling [47] has a reduced computational cost compared to the GCNN from Deferrard et al. [39].

The loss function reads as Eq. (4.2) where each Z_i is a subset of Z associated with graph $\mathcal{G}_i$. μ_{age}, μ_{sex}, $\mu_{age\&sex}$, μ_{ns}, μ and γ_{l2} are parameters to control the trade-off between the different loss terms.

$$l(\theta) = \frac{\mu_{age}}{2}||Z_{age}||^2_{\mathcal{G}_{age}} + \frac{\mu_{sex}}{2}||Z_{sex}||^2_{\mathcal{G}_{sex}} + \frac{\mu_{age\&sex}}{2}||Z_{age\&sex}||^2_{\mathcal{G}_{age\&sex}} + \frac{\mu_{ns}}{2}||Z_{ns}||^2_{\mathcal{G}_{ns}} + ||\Omega_a * (Z - M)||^2_F + \mu l_{\Omega_b}(Z, M) + \gamma_{l2}\sum_{i=1}^{q} W_i. \quad (4.2)$$

In order to compare the one graph and the multiple graph architecture, we also apply the sRGCNN defined in Section 3.3 by Vivar et al. [96] to our dataset. In that case, we use the graph from Parisot et al. defined in Section 3.3. The loss function is defined as Eq. (4.3) where γ, γ_W, γ_H, γ_e and γ_{l2} are parameters to control the trade-off between the different loss terms. $\mathcal{G}_r$ is the graph on the rows of W.

$$l(\theta) = \gamma||W||^2_{\mathcal{G}_r} + \gamma_W||W||^2_F + \gamma_H||H||^2_F + ||\Omega_a \circ (M - WH^T)||^2_F + \gamma_e(l_{\Omega_b}(M, Z)) + \gamma_{l2}\sum_{i=1}^{q} W_i. \quad (4.3)$$

Based on the procedure adopted by Parisot et al. [103] for the ABIDE dataset, we added a similarity metric on the graphs. Indeed, Parisot et al. do not use it on the ADNI dataset because they are already using a similarity measure for the longitudinal aspect of their dataset. The similarity metric takes into account feature values to put a larger weight on the edge between two people that have similar values. We use a correlation distance between the features of two subjects $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^n$ defined as Eq. (4.4) where $\overline{v}$ is the mean value of v and $x.y$ is the dot product of x and y. For the MG-RGCNN GCN, we tried the graphs with (MG-RGCNN GCN similarity) and without (MG-RGCNN GCN) the similarity metric.

$$dist(u, v) = 1 - \frac{(u - \overline{u}).(v - \overline{v})}{||(u - \overline{u})||_2||(v - \overline{v})||_2}. \quad (4.4)$$

4.4 Results

We apply the proposed MG-RGCNNs to the TADPOLE dataset [111], a dataset for the prediction of conversion from Mild Cognitive Impairment (MCI) to Alzheimer's

disease (AD). The TADPOLE dataset and the preprocessing steps used are described in Appendix A.

MCI is an intermediate stage between the normal stage and dementia. Patients with MCI are in a stage where the disease could evolve to AD (MCI converters, MCIc) or not (MCI non converters, MCInc). In forming predictions, we use the baseline data acquired from the first examination of a subject. We include subjects that were diagnosed as MCI in their baseline scan and that have converted to AD 48 months later (MCIc) or that have remained stable for the course of the study (MCInc).

We excluded features where more than 50% of the values were missing. After the preprocessing steps described in Appendix A, we have 779 subjects, 296 MCIc and 483 MCInc, and 563 features. For the included subjects, 21% of feature values are missing.

We added the label (disease outcome = MCIc or MCInc) column as the last column of the matrix. Each column of M is normalized so that the values lie between -1 and 1, by scaling based on the minimum and maximum observed values in the column. Some of the elements of M may be missing. We set these to a value of zero.

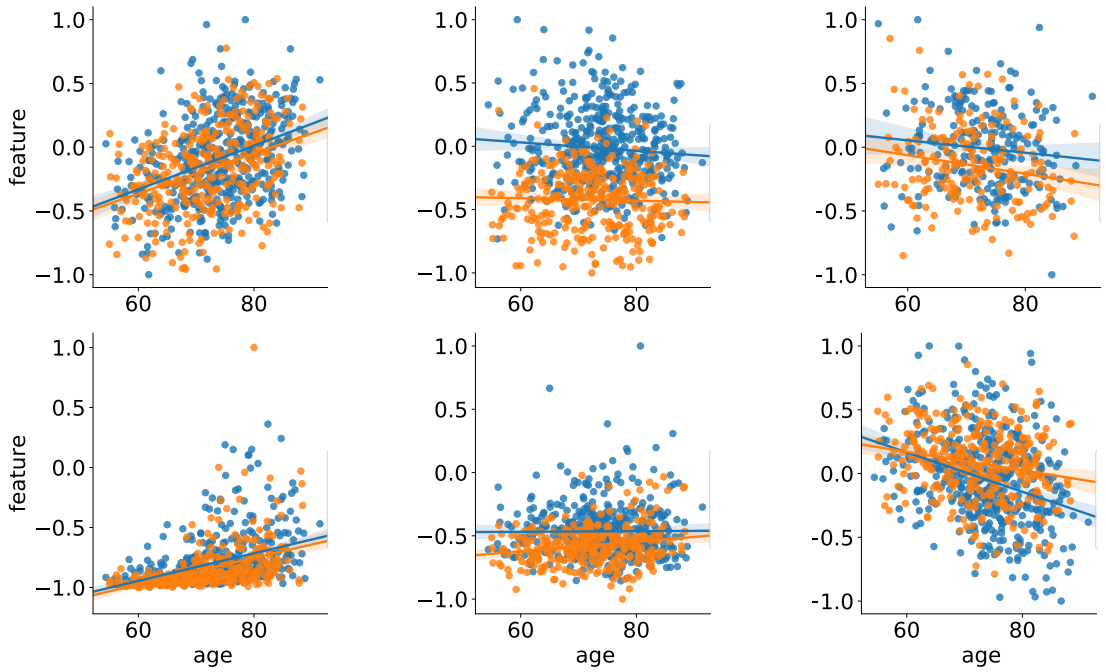
4.4.1 Graph construction

AD and MRI-derived brain volume features are known to be related to age and sex [107]–[110], so these attributes are used to construct the graphs, as in Parisot et al. [103] and Vivar et al. [96].

We conducted the GLM analysis using the three variables age, sex, and age&sex, employing a significance threshold for the p-values of 0.05. The process is highlighted in Fig. 4.1 with the example of an age-related feature.

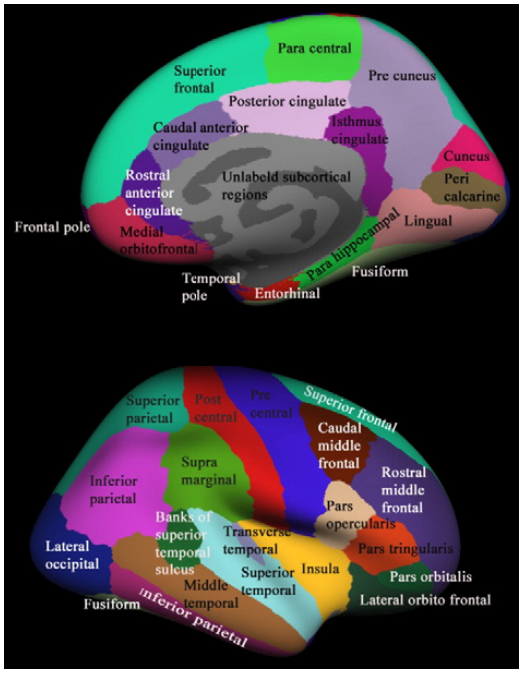
As expected, different features have different relationships with age and sex, as illustrated by the examples in Fig. 4.3. The analysis leads to 452 age-related features, 188 sex-related features, 123 age&sex-related features, and 89 features with no relationship to age or sex.

Usually, we would apply a correction to account for the fact that we are performing multiple hypothesis tests. Here our main goal is to exclude features for which there is no strong evidence of a relationship; spurious inclusion of some variables is less of a concern.

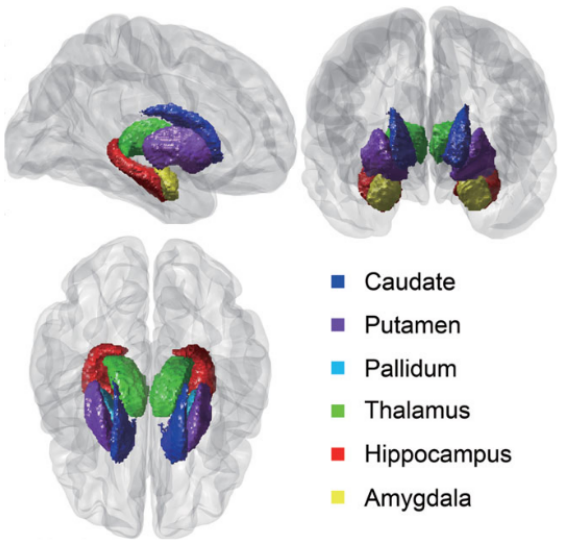


Age-related features. **Sex**-related features. **Age & Sex** features.

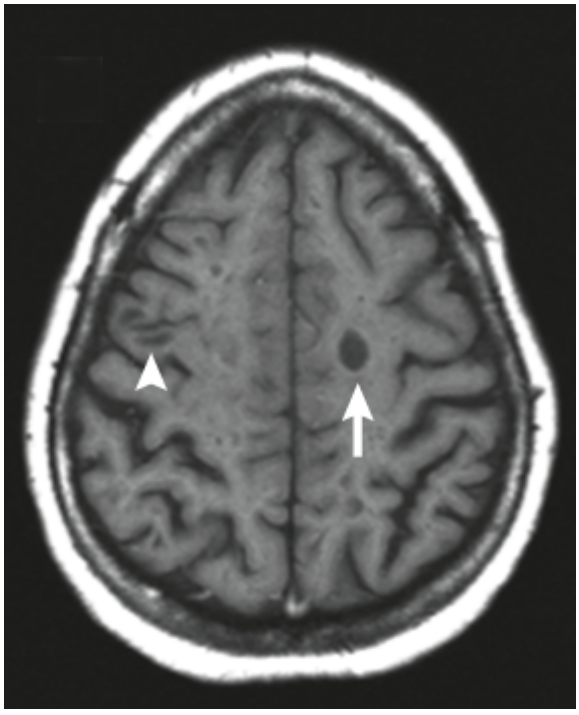
FIGURE 4.3: Relationships of age and sex (Men and Women) with six different features in the case of Alzheimer's disease. The age-related features are the left caudal anterior cingulate cortical thickness standard deviation (top) and the hypointensities volume (bottom); the sex-related features are intracranial volume (top) and the left caudate volume (bottom); the age & sex-related features are the raw volume value for the right pars orbitalis (top) and the cortical thickness average of the left pars orbitalis (bottom).



Anatomical subregions of cerebral cortex. *Reproduced from [112]*



3D view of subcortical segmentations. *Reproduced from [113]*



Hypointensities. *Reproduced from [114]*

FIGURE 4.4: Brain regions studied in Fig. 4.3

Li et al. [113] and Taki et al. [115] describe models also based on GLMs to measure the age and sex dependence in regions of the brain for healthy subjects. The volume values of subcortical regions such as thalamus, caudate or hippocampus are used to measure the dependence of these regions.

The results from Li et al. [113] are with a dataset with healthy subjects from 19 to 70. The same brain volumes are measured in the TADPOLE dataset in two different modalities, one in the UCSF FreeSurfer dataset, via MRI, and the other in the UC Berkeley dataset, via AV45-PET.

We compare in Table 4.1 the dependencies of five volumes of interest that are highlighted in Fig. 4.5. The dependencies are very different because the age range of the subjects is different. In our study, we only have subjects from 54 to 92 years old whereas Li et al. have subjects from 17 to 70 years old. When looking at the plots of the volume values as a function of age in Li et al. [113], we can see that the dependence changes when we only take the values when the age is greater than 54. For example, caudate becomes only sex dependent and thalamus only age dependent. Putamen, amygdala and hippocampus remain with the same dependence.

Brain regions	**Our results**	**Li et al. [113]**
Hippocampus	Age	Age & sex
Amygdala	Age	Sex
Caudate	Sex	Age, sex
Putamen	Age, sex	Age, sex
Thalamus	Age	Age, sex

TABLE 4.1: Results of the study of feature dependencies with age and sex.

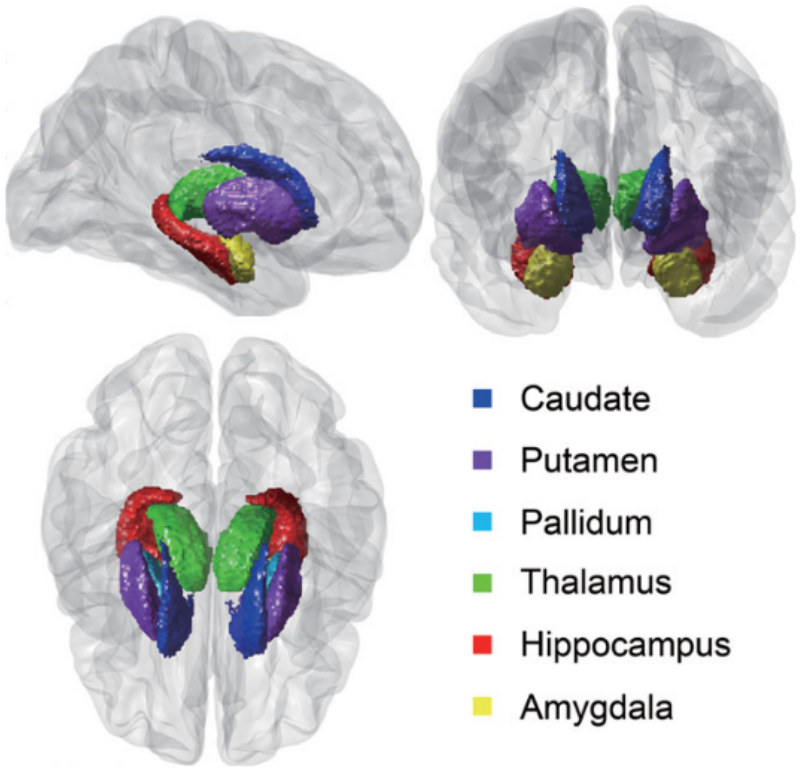


FIGURE 4.5: 3D view of brain regions studied in Table 4.1. *Reproduced from [113].*

Thus the GLM results of feature dependence with age and sex prove that different features have different relationships with age and sex. This justifies our focus on an architecture taking into account the feature differences by using a multiple-graph architecture.

4.4.2 Optimization of the hyperparameters

We develop a Python and Tensorflow implementation of the algorithm, building on the matrix completion code provided by Monti et al. [36]. We split the data between a training (60%), validation (20%) and test set (20%). We put the same percentage of MCIc and MCInc in each set.

Optimization is realized with RBFOpt [116]. In order to ensure that classification performance generalizes, we optimize the hyperparameters over a validation set that is different from the test set. We want to maximize the validation AUC. We take the best validation AUC over 1000 iterations as the value of the function we want to optimize.

For the sRGCNN

We first have to optimize the hyperparameters of the sRGCNN. The hyperparameters are:

- the weight parameters in front of the loss terms in the loss function (Eq. (4.3)): γ, γ_W, γ_H, γ_e and γ_{l2},
- the number of hidden units h,
- the learning rate l,
- the number of Chebyshev polynomials used for the GCNN c.

Trying to optimize 8 hyperparameters at the same time is extremely computationally expensive. To reduce the overhead, we assume that we should optimize the 5 parameters of the loss function together as they are closely related but we can optimize the other parameters individually as they are not closely related. We first optimize the parameters of the loss function and proceed to optimize the other three parameters. Depending on the number of values to optimize and on the bounds, we run the optimization for I different tries of values for the hyperparameters.

Hyperparameter to optimize	**Bounds**	**Fixed parameters**	I	**Identified value**	**AUC**
Loss coefficients	1-100	$h = 36$ l=0.00089 c=18	200	γ_{l2}, γ, γ_H, γ_W, γ_e=17, 100, 68, 68, 70	76.8
Number of hidden units h	1-50	γ_{l2}, γ, γ_H, γ_W, γ_e= 17, 100, 68, 68, 70 l=0.00089 c=18	50	h=17	77.1
Learning rate l	5×10^{-3}- 5×10^{-5}	γ_{l2}, γ, γ_H, γ_W, γ_e= 17, 100, 68, 68, 70 h=17 c=18	100	l=0.0003	81.2
Number of Chebyshev polynomials c	1-50	γ_{l2}, γ, γ_H, γ_W, γ_e= 17, 100, 68, 68, 70 h=17 l=0.0003	50	c=18	81.2

TABLE 4.2: Results of the optimization of the hyperparameters for the sRGCNN architecture. The AUC reported is the one on the validation set.

For the MG-RGCNN

As the loss function is different for the MG-RGCNN, we cannot use the values found for the sRGCNN. Indeed, we now have one coefficient for each graph norm as highlighted by Eq. (4.2). We used the simplified version with the GCN layers, the MG-RGCNN GCN, in order to reduce the computational complexity.

The hyperparameters are:

- 6 parameters in front of the loss terms in the loss function (Eq. (4.2)): μ_{age}, μ_{sex}, $\mu_{age\&sex}$, μ_{ns}, μ and γ_{l2},
- nb of hidden units h,
- learning rate l.

As for the sRGCNN, we first optimize the 6 parameters of the loss function and proceed to optimize individually the number of hidden layers and the learning rate. Depending on the number of values to optimize and on the bounds, we run the optimization for I different tries of values for the hyperparameters.

Hyperparameter to optimize	**Bounds**	**Fixed parameters**	I	**Identified value**	**AUC**
Loss coefficients	1-100	h=36 l=0.001	500	γ_{l2}, μ_{age}, μ_{sex}, $\mu_{age\&sex}$, μ_{ns}, μ=1, 84, 100, 29, 82, 84	73.2
Number of hidden units h	1-100	γ_{l2}, μ_{age}, μ_{sex}, $\mu_{age\&sex}$, μ_{ns}, μ l=1, 84, 100, 29, 82, 84, 0.001	100	h=51	75.4
Learning rate l	5×10^{-3}-5×10^{-5}	γ_{l2}, μ_{age}, μ_{sex}, $\mu_{age\&sex}$, μ_{ns}, μ, h=184, 100, 29, 82, 84, 51	100	l=0.0008	77.2

TABLE 4.3: Results of the optimization of the hyperparameters for the MG-RGCNN architecture. The AUC reported is the one on the validation set.

4.4.3 Experiment results

In this section, we fix the parameters to the values found in the optimization step. We train the architecture on the training set and we compute the test AUC at the iteration where the AUC is maximal on the validation set. We do that for different initializations of the splitting of the train, validation and test sets. We do an average over all the different initializations. We test the different architectures on the TADPOLE dataset.

A Random Forest (RF), a linear SVM and a Multi-Layer Perceptron (MLP) were tested on the dataset to see the performance of these standard methods compared to the methods that we propose here. As we had missing values, we decided to fill them with the mean value of the corresponding missing feature. Then we apply the algorithm. For the RF, the number of estimators is a hyperparameter so as previously done for the sRGCNN, we optimize the validation AUC in order to find the best value for the number of estimators. We identified a value of 80. For the MLP, we put one hidden layer and optimize the validation AUC in order to find the best number of hidden units. We identified a value of 50 hidden units.

We also replicated the architecture from Parisot et al. [103]. As it is not an architecture that deals with missing values, we had to fill the missing values and performed it the same way we did it for the standard methods. As the dataset is different than the one used in their paper, we optimized the hyperparameters (number of hidden layers L, number of hidden units per hidden layer h, l2 regularization parameter γ, learning rate lr, dropout rate d) with RBFOpt during 500 runs of the algorithm with different values for the hyperparameters. We found $L = 3$, $h = 76$, $\gamma = 10^{-6}$, $lr = 5 \times 10^{-4}$ and $d = 0.832$. We did 200 iterations for one run to train the architecture as mentioned in the paper. We used the graph construction variables from the paper which are the same as the one used in the sRGCNN from Vivar et al. [96].

The results on the TADPOLE dataset are given in Table 4.4. As the results are close and can overlap, we computed the Wilcoxon signed-rank test (Table 4.5). This test compares the different test AUC values obtained for the same initialization to determine if the performance difference is statistically significant over the 100 different initialization. We conducted the test to compare the results of sRGCNN (1), MG-RGCNN GCN similarity (2), MG-RGCNN GCNN similarity (3), random forest (4), linear SVM (5), multi-layer perceptron (6) and the architecture from Parisot et al. (7).

Algorithm - Corresponding number	Mean AUC ± std
sRGCNN [96] - (1)	0.7191 ± 0.0556
MG-RGCNN GCN	0.693 ± 0.040
MG-RGCNN GCN similarity - (2)	0.698 ± 0.041
MG-RGCNN GCNN similarity - (3)	0.739 ± 0.044
Random Forest - (4)	0.771 ± 0.031
Linear SVM - (5)	0.690 ± 0.027
Multi-Layer Perceptron - (6)	0.736 ± 0.037
Parisot et al [103] - (7)	0.767 ± 0.036

TABLE 4.4: Mean test AUC in the different cases presented for the TADPOLE dataset.

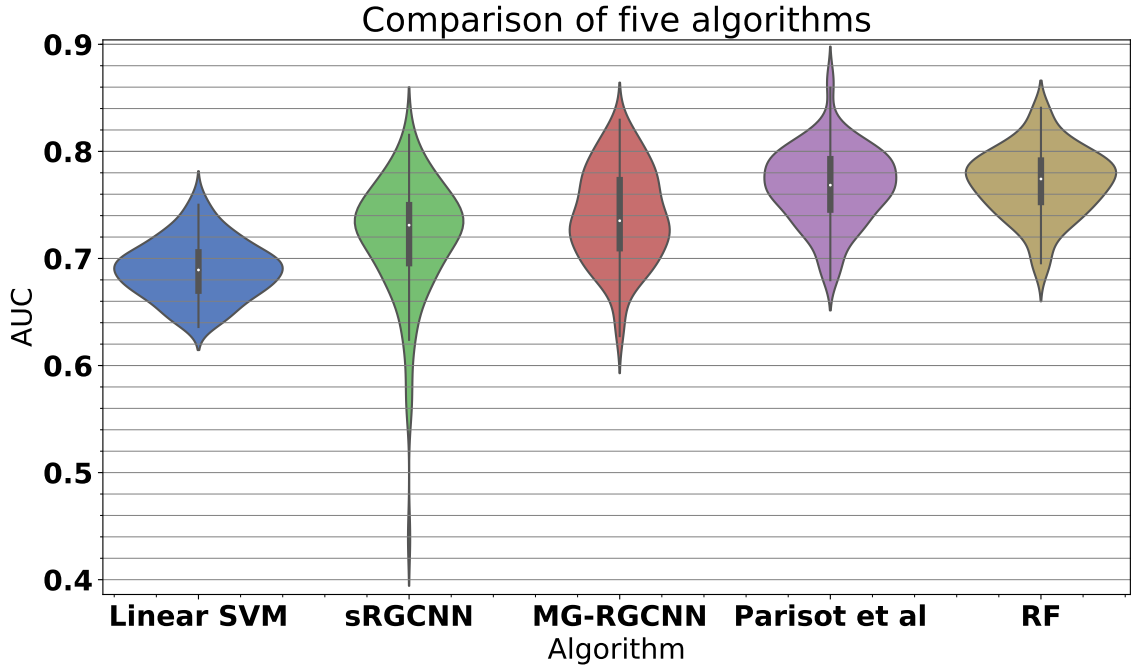


FIGURE 4.6: Violin plots of the distribution of the AUC over the 100 different train/validation/test initializations for linear SVM, sRGCNN, MG-RGCNN, Parisot et al. and random forest.

Using the GCNN layer from Defferard et al. [39] as done in the sRGCNN from Vivar et al. [96] helps in improving the AUC values from almost 4% compared to the architecture using the GCN layer from Kipf and Welling [47]. We can also see that the simplified version with the GCN layer is performing worse than the version with the GCNN as the p-value for the Wilcoxon test is 2.61×10^{-11}. The only difference is that it is more computationally expensive and takes four time more time to run the architecture with the GCNN rather than the one with the GCN.

There is an improvement of 2% on the mean AUC compared to Vivar et al. [96] with the MG-RGCNN GCNN similarity. With the Wilcoxon test results, we can see that the MG-RGCNN GCNN similarity outperforms the sRGCNN as the p-value for

the Wilcoxon test is 4.11×10^{-3}. The random forest and the architecture of Parisot et al. [103] outperform all the other algorithms. For both these architectures, the matrix completion task is not being performed. These algorithms are only performing the classification task. The missing values are imputed by a mean of the known values for this feature. This outperformance could be due to the fact that the dataset is not large and there are only 21% of missing values which is not enough to interfere with the classification results.

	1	2	3	4	5	6	7
1		6×10^{-5}	4×10^{-3}	5×10^{-15}	2×10^{-8}	2×10^{-2}	2×10^{-14}
2	6×10^{-5}		3×10^{-11}	6×10^{-18}	1×10^{-1}	1×10^{-11}	8×10^{-18}
3	4×10^{-3}	3×10^{-11}		3×10^{-11}	1×10^{-13}	3×10^{-1}	1×10^{-8}
4	5×10^{-15}	6×10^{-18}	3×10^{-11}		6×10^{-18}	3×10^{-14}	2×10^{-1}
5	2×10^{-8}	1×10^{-1}	1×10^{-13}	6×10^{-18}		7×10^{-16}	6×10^{-18}
6	1×10^{-2}	1×10^{-11}	3×10^{-1}	3×10^{-14}	7×10^{-16}		5×10^{-14}
7	7×10^{-14}	8×10^{-18}	1×10^{-8}	2×10^{-1}	6×10^{-18}	5×10^{-14}	

TABLE 4.5: Wilcoxon scores for the TADPOLE dataset. 1: sRGCNN, 2: MG-RGCNN GCN similarity, 3: MG-RGCNN GCNN similarity, 4: random forest, 5: linear SVM, 6: multi-layer perceptron, 7: the architecture from Parisot et al.

Parameter	**Value**
Split train/validation/test	0.6/0.2/0.2
Number of iterations	1000
Number of different train/validation/test initialization	100

TABLE 4.6: Table of fixed hyperparameters to run each different algorithm.

4.5 Conclusion

We introduce a multiple-graph architecture based on a graph-based geometric matrix completion method to predict disease outcomes for datasets with missing values. We use a statistical significance test to determine the subsets of the features that are relevant to each of the graphs. This leads to an improvement of 2% on the mean AUC compared to Vivar et al. [96]. The MG-RGCNN algorithm helps in performing better classification as it takes into account more accurately the feature dependencies with age and sex and allows to better recover the missing values. However, it is being outperformed by the random forest and the GCNN-based algorithm designed by Parisot et al. [103], architectures where the missing values are imputed by

a mean of the known values for this feature. This could be due to the fact that the dataset is not large and there are only 21% of missing values which is not enough to interfere with the classification results. Moreover, the AUC on the training set is close to 1 which is probably indicative of overfitting. The architecture is perhaps too powerful for the provided data. One way to alleviate this might be to introduce perturbances in the data.

The obtained results are considerably lower than those reported by Vivar et al. [96]. This may be due to the preprocessing of the dataset as it is not explained in much detail in the paper. It can also be due to the fact that there is an unfortunate overlap in the training and test data in the train/test split specified in the TADPOLE challenge overlap. Some entities are present in both training and test sets which biased the results. In the split that we used, there is no overlap of the training, validation and test set. We are not using the train/test split specified in the TADPOLE challenge.

5 Multiple-Graph Graph Auto-Encoder architectures for predicting disease outcomes

5.1 Introduction

Based on the model developed by Vivar et al. [96] where using an architecture first developed for matrix completion to do classification resulted in promising results, we decided to look for architectures used for matrix completion that had better results on the matrix completion task than those obtained by Monti et al. [36]. This leads us to focus on the Graph Convolutional Matrix Completion (GC-MC) developed by Van den Berg et al. [38] that achieved better results than the sRGCNN from Monti et al. on the task of recommender systems. Instead of computing the embeddings for the users and the items based on the rating values available, we compute the embeddings based on personal information about the subjects and on the relationships of this information with the features. We add the labels as one column of the matrix to complete and add a cross-entropy term for the labels in the loss function. The major contribution is on the transition from an architecture for categorical data to one for continuous data. Our approach is the first one to use a graph auto-encoder for bipartite graphs for disease outcome prediction.

Section 5.2 provides a formal statement of the problem. Section 5.3 provides our approach and algorithm. Section 5.4 presents the results of the application of our approach to the prediction of Alzheimer's disease development.

5.2 Problem statement

We consider the following prediction task for disease outcomes. Let $X \in \mathbb{R}^{m \times n}$ be the feature matrix, m being the number of subjects and n the number of features. The features are assumed to be derived from medical examinations. X may have missing values. Let $Y \in \{0,1\}^{m \times 1}$ be a vector denoting the disease outcomes for

the m subjects. Some of these are unknown and these are the focus of the prediction task.

Let $\mathcal{G}_i = \{U_i, V_i, E_i\}$ be a bipartite graph on the subjects and the features with edges derived by a similarity metric from a subject attribute s_i. The attribute can be categorical, or real- or integer-valued. U_i and V_i denote the vertices, $U_i = \{1, ..., m\}$ is a subset of the subjects, $V_i = \{1, ..., n\}$ is a subset of the features and E_i are the edges. We assume that there are P such graphs derived from different combinations of subject attributes and thus capturing different relationships between subjects and features. Taking into account the features X and the relationships formed by the similarities of the attributes s_i and captured by G_i, $i = 1, \ldots, P$, our task is to predict the unknown disease outcomes in Y and impute the missing values in the matrix X.

5.3 Methodology

In the task of disease outcome prediction, most of the datasets have missing values and inaccurate measurements. Formulating the task as matrix completion as in [105] allows us to jointly perform transductive classification and imputation of missing values. To do this, we form a matrix $Z = [X, Y]$ and apply a matrix completion algorithm. The initial matrix M contains all the information from the dataset.

We commonly have knowledge of attributes that can be used to identify relationships or similarities between subjects. Attributes such as age and sex often impact the probability of a disease outcome and the likelihood of a feature derived from a medical examination. In trying to recover a matrix with missing values and unknown disease outcomes, it is reasonable to interpret every element of the matrix as an edge weight. The edge exists in one of multiple bipartite graphs. The graph it is located in depends on the attributes of the subject and the feature. Our task is to infer the weight. If there is an edge in a graph between a specific subject and a certain feature (e.g hippocampus volume) then the weight associated with that edge is the subject's value for that feature.

After building the bipartite graphs, we have to infer the edge weights. In order to do so, we build embeddings for each feature and each subject based on the known edges. We do so for each one of the graphs. It is reasonable to assume that we can compute an embedding for a certain feature from all the known edges from this feature. Subjects that are related to the same feature in one graph are supposed to be similar and thus should have similar values. In the same way, we compute

embeddings for the subjects. Subjects that are similar should have a similar embedding. On the one hand, when reconstructing the edges between one subject and one feature, subjects that are similar should have a similar edge weight. On the other hand, subjects that are different should have different subjects' embeddings and thus different edge weights.

The bipartite graphs are built based on attributes of the subject and the feature. The relationships between the features and the attributes of the subjects are found by the same technique as the one developed in Section 4.3. For a binary attribute $a_b \in \{0,1\}$ such as sex, two groups of subjects are created, one where the subjects have 0 for this attribute and another one where the subjects have 1 for this attribute. These two groups lead to two graphs where each bipartite graph is between the subjects of one group and the features related to that attribute. For a continuous attribute $a_c \in [a_{min}, a_{max}]$ such as age, we choose a value l such that we split $[a_{min}, a_{max}]$ into l intervals. For each interval, we group together the subjects that have a value for a_c in the given interval. These l groups lead to l bipartite graphs where each graph is between the subjects of one group and the features related to that attribute. A group of subjects is defined by an attribute of a subject or by an intersection of attributes and by a range of attribute values. A group of features is defined by an attribute of a subject or by an intersection of attributes.

5.3.1 Multiple-Graph Graph Auto-Encoder (MG-GAE)

We propose an architecture based on a graph auto-encoder. We adapt the architecture from [38] to take into account the subjects' attributes, the fact that the values in the matrix are continuous and the prediction task. We have a matrix M of dimensions $m \times n$, where m is the number of subjects and n is the number of features. The n-th column of this matrix is the outcome vector y. The architecture is described in Fig. 5.1 where the encoding is done for subject i and feature j. The relationships taken here are age, sex and age & sex but it is possible to take other relationships into account.

Each column of M is normalized so that the values lie between -1 and 1, by scaling based on the minimum and maximum observed values in the column. Some of the elements of M may be missing. We set these to a value of zero. We construct two identity matrices $I_{m\times m}$ and $I_{n\times n}$. Each row corresponds to a subject (feature).

We have K binary indicator matrices $S^{(i)}$ that correspond to a relationship between a group of subjects and a group of features. For example, the matrix $S^{(1)}$

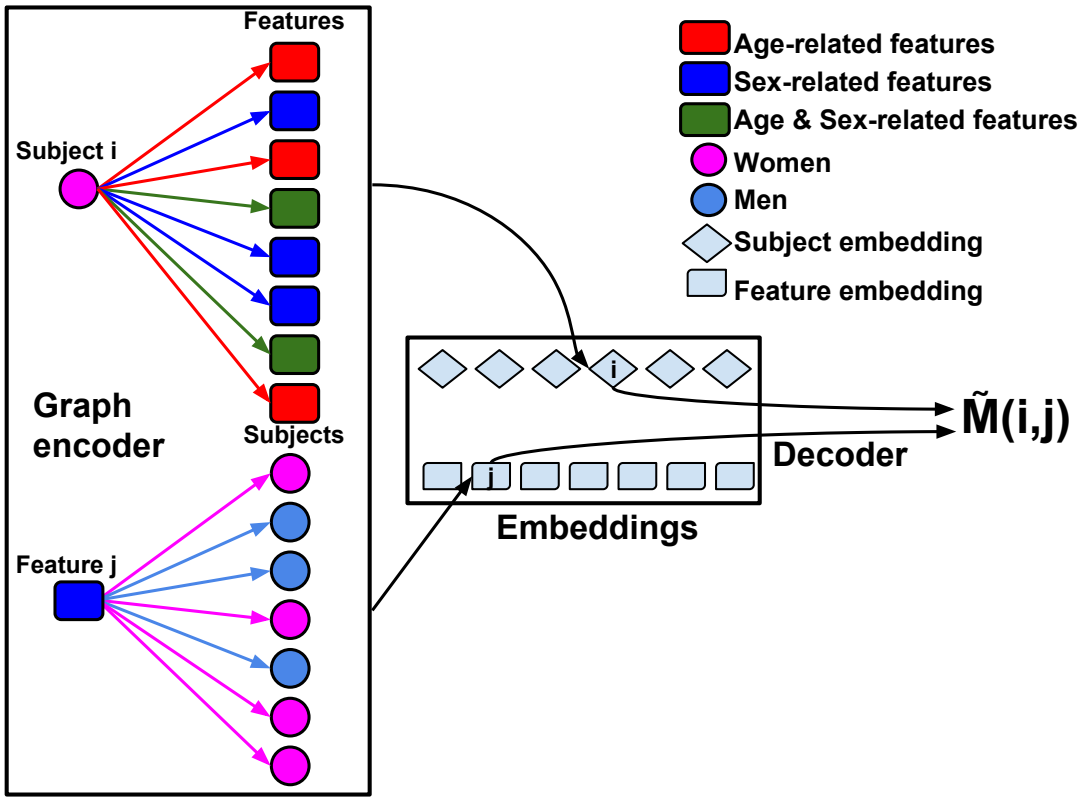


FIGURE 5.1: Graph auto-encoder process for subject i and feature j.

might indicate subjects that are in the age range 80-85 and features that are age-dependent. We define $A^{(i)} = M \odot S^{(i)}$ as the element wise product of the normalized data matrix M and the i-th indicator $S^{(i)}$. An example of several bipartite graphs is given in Fig. 5.2 where the values of $A^{(i)}$ are on the edges. If the value corresponding to an edge is null then there is no edge. For each of the i indicator matrices, we identify two real-valued weight matrices $W^{(i)}$ (dimensions $m \times o$) and $V^{(i)}$ (dimensions $n \times o$). o is the subject and the feature-specified embedding dimension.

The initial embedding $Q^{(i)}$ of the subject is obtained by multiplying $I_{m\times m}$ by $W^{(i)}$, so that $Q^{(i)} = I_{m\times m}W^{(i)} = W^{(i)}$. It is clear that the o-dimensional embedding of the j-th subject is just the j-th row of the weight matrix. Similarly, the initial embedding $P^{(i)}$ of the features is obtained by multiplying $I_{n\times n}$ by $V^{(i)}$, so that $P^{(i)} = I_{n\times n}V^{(i)} = V^{(i)}$. It is clear that the o-dimensional embedding of the j-th feature is just the j-th row of the weight matrix $V^{(i)}$.

We then define $Z^{(i)} = A^{(i)}P^{(i)}$ to form an $m \times o$ dimensional matrix. If we inspect this equation, we see that the j-th row is a weighted sum of the embeddings $p_s^{(i)}$. The weights correspond to the entries of $A^{(i)}$ corresponding to the j-th subject. We have $z_j^{(i)} = \sum A_{js}^{(i)} p_s^{(i)}$. Similarly we define $B^{(i)} = A^{(i)T}Q^{(i)}$ to form an $n \times o$ dimensional matrix. If we inspect this equation, we see that the j-th row is a weighted sum of

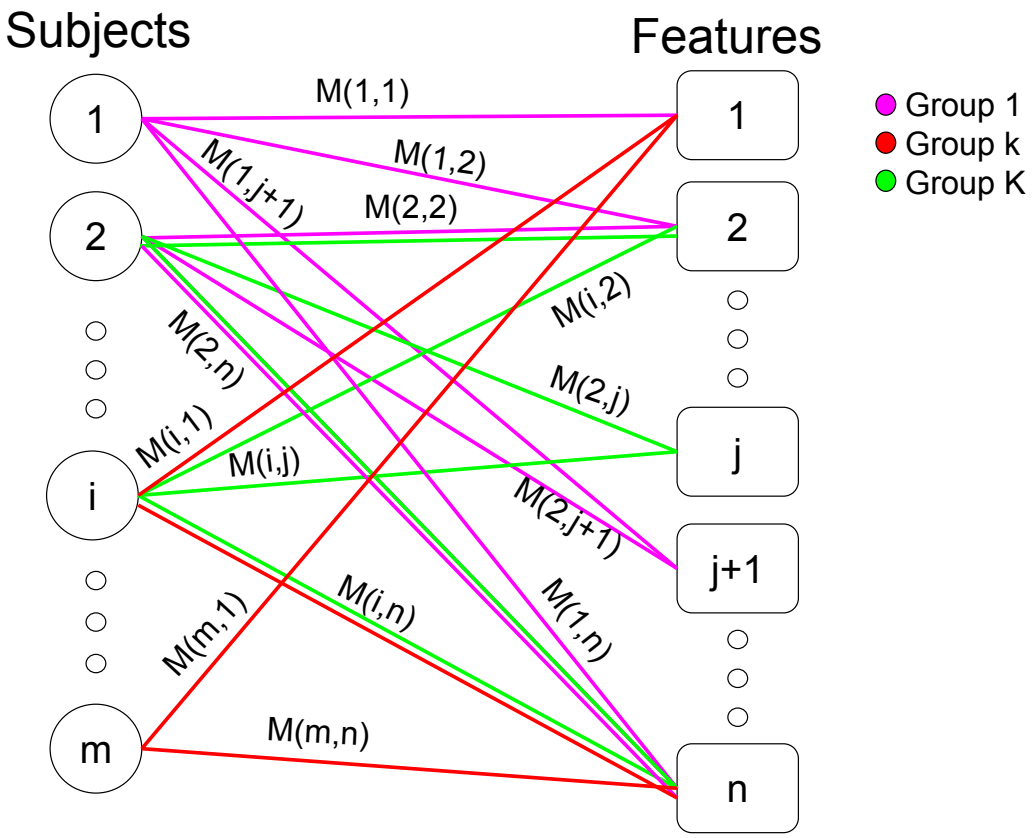


FIGURE 5.2: Three bipartite graphs corresponding to different attributes of the subjects. The three colors represent three different bipartite graphs that act between different groups of subjects and features. Group 1 is for example a group of subjects that have an age between 70 to 75. Subjects 1 and 2 have an age between 70 to 75 and feature 1, 2, $j+1$ and n are age-related features. $M(2,1)$ is missing hence the missing edge.

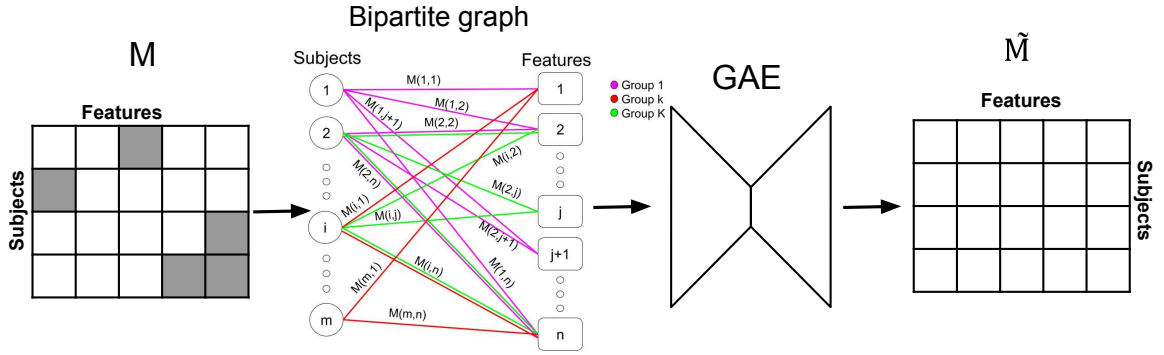


FIGURE 5.3: Depiction of the architecture. M is the input and the grey elements are missing values. $\tilde{M}$ is the output. GAE is the Graph Auto-encoder.

the embeddings $q_s^{(i)}$. The weights correspond to the entries of $A^{(i)}$ corresponding to the j-th feature. We have $b_j^{(i)} = \sum A_{sj}^{(i)} q_s^{(i)}$. We use the term "GCN layer" to refer to the operation performed to create $Z^{(i)}$ and $B^{(i)}$. We concatenate the $Z^{(i)}$ and $B^{(i)}$ to form matrices Z and B. At this stage, a non-linearity (e.g. ReLU) is applied to Z and B. Finally, we have linear transforms $Z' = ZG$ and $B' = BH$ to produce matrices of dimension $m \times d$ and $n \times d$. G and H are learnable parameters. We can add a non-linearity to Z' and B'. We use the term "dense layer" to refer to the operation performed to create Z' and B'.

For the decoder, we identify one weight matrix W_d (dimension $d \times d$). The decoder performs the operation $Z'W_dB'^T$ to produce an output of dimension $m \times n$. We can add a non-linearity to the produced output.

The loss function is defined as Eq. (5.1) where the first term is for the matrix completion task and the second one for the classification task. The matrix completion term is the Frobenius norm $||.||_F$ of the difference between the values of the known feature entries of the input matrix M and the reconstructed matrix $\tilde{M}$. The Frobenius norm is $||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$ for an $m \times n$ matrix A. Ω_a is the indicator matrix of the known feature values, the concatenation of a $m \times n$ matrix filled with 1 when the feature is available in M and 0 otherwise and a $m \times 1$ vector of 0. The classification term is a binary cross-entropy term $l_{\Omega_b}(Z, M) = -(y \log(p) + (1 - y) \log(1 - p))$, where Ω_b is the indicator matrix of known outcomes, the concatenation of a $m \times n$ matrix filled with 0 and a $m \times 1$ vector of 1, p is the classification output vector and y the label vector. The last one is an l2 regularization term where $(W_1, \ldots, W_q)$ represents the weight matrices of the architecture. γ and β control the balance between the different loss terms.

$$l = ||\Omega_a * (M - \tilde{M})||_F + \gamma l_{\Omega_b}(M, \tilde{M}) + \beta \sum_{i=1}^{q} W_i. \quad (5.1)$$

5.3.2 Application to Alzheimer's disease

We apply the proposed MG-GAE to the prediction of conversion from MCI to AD with the same dataset as the one used in Chapter 4, the TADPOLE dataset. The first step is to choose the support matrices. In the TADPOLE dataset, we have subjects from 54 to 92 years old. For each of the support matrices S_i, $S_i(k,l) = M(k,l)$ if subject k belongs to the group i and if feature l is dependent on the characteristic of group i. For example, the first support matrix is for men so $S_1(k,l) = M(k,l)$ if subject k is a man and if feature l is sex-dependent. We use the same support matrices for the synthetic dataset. Table 5.1 gives the 23 characteristics used for the support matrices. M and W denote Men and Women. 54 - 59 means that the subjects that are between 54 and 59 are used in this support matrix. In order to know the feature dependencies, we use the statistical significance test introduced in Chapter 4 to determine the subsets of the features that are relevant to each of the graphs.

Feature dependence	**Characteristics used for the support matrices**
Sex-related	Men, Women
Age-related	54 - 59, 59 - 64, 64 - 69, 69 - 74, 74 - 79, 79 - 84, 84 - 92
Age & Sex related	54 - 59 & M, 59 - 64 & M, 64 - 69 & M, 69 - 74 & M, 74 - 79 & M, 79 - 84 & M, 84 - 92 & M, 54 - 59 & W, 59 - 64 & W, 64 - 69 & W, 69 - 74 & W, 74 - 79 & W, 79 - 84 & W, 84 - 92 & W

TABLE 5.1: List of the 23 support matrices

5.4 Results

We discuss here the different experiments realized for the task of prediction of conversion from MCI to AD. First, we optimized the hyperparameters in order to increase the performance and then we compared our results to the results of standard classification method algorithms. Finally, we inspect the embeddings Z and B generated by the GCN layer and assess whether they are meaningful.

5.4.1 Optimization of the hyperparameters

We develop a Python and Tensorflow implementation of the algorithm, building on the matrix completion code provided by Van Den Berg et al. [38].

We tested this architecture for the task of prediction of conversion from MCI to AD. We used the TADPOLE dataset as in Chapter 4 but we also created a synthetic dataset to help develop an understanding of the behaviour of the algorithm. Details concerning the creation of the synthetic dataset are given in Appendix A. The first step of the experiments was to choose the support matrices. After determining the normalization to use, we can optimize the hyperparameters in each case in order to assess the performance of the proposed architecture.

Optimization is realized with RBFOpt [116]. In order to ensure that classification performance generalizes, we optimize the hyperparameters over a validation set that is different from the test set. We want to maximize the validation AUC. We take the best validation AUC over 1000 iterations as the value of the function we want to optimize. The 5 hyperparameters are:

- parameters in front of the loss terms in the loss function (Eq. (5.1)) γ and β,
- number of hidden units for the GCN layer for one support a,
- number of hidden units for the dense layer b,
- learning rate.

We first run the optimization for the synthetic dataset for 500 different tries of values for the hyperparameters and the best value of the AUC on the validation set was 88.93. The optimized hyperparameters are presented in Table 5.2.

Hyperparameter	**Bounds**	**Identified value**
γ	0.1-100	0.119
β	0.001-1	0.091
a	1-100	61
b	1-100	47
Learning rate	5×10^{-5}-5×10^{-3}	7.80×10^{-3}

TABLE 5.2: Results of the optimization of the hyperparameters for the synthetic dataset.

Then we run the optimization on the TADPOLE dataset for 500 different tries of values for the hyperparameters and the best value of the AUC on the validation set was 79.26. The hyperparameters found for this value of AUC are reported in Table 5.3.

Hyperparameter	Bounds	Identified value
γ	0.1-100	0.576
β	0.001-1	0.914
a	1-100	99
b	1-100	25
Learning rate	5×10^{-5}-5×10^{-3}	9.97×10^{-3}

TABLE 5.3: Results of the optimization of the hyperparameters for the TADPOLE dataset.

5.4.2 Results of the experiments

In this section, we fix the parameters to the values found in the optimization step in Section 5.4.1. For each experiment, defined by a different initialization of the subjects in the training, validation and test sets, we now use the validation set to know at which iteration the validation AUC is the best and to compute the test AUC for that iteration. Then we average over the 100 different initializations to have generalized results. We test the proposed architecture on the synthetic dataset and on the TADPOLE dataset. The violin plots of the results on the TADPOLE dataset for linear SVM, sRGCNN [96], MG-GAE, the GCNN-based algorithm designed by Parisot et al. [103] and random forest are presented in Fig. 5.4.

Algorithm	Mean AUC $\pm$ std
MG-GAE	0.842 ± 0.026
Random Forest	0.760 ± 0.034
Linear SVM	0.934 ± 0.013

TABLE 5.4: Mean test AUC in the different cases presented for the synthetic dataset.

Algorithm	Mean AUC $\pm$ std
sRGCNN [96]	0.719 ± 0.056
MG-RGCNN GCNN similarity	0.739 ± 0.044
MG-GAE	0.748 ± 0.036
Random Forest	0.771 ± 0.032
Linear SVM	0.690 ± 0.027
Multi-Layer Perceptron - (6)	0.736 ± 0.037
Parisot et al [103] - (7)	0.767 ± 0.036

TABLE 5.5: Mean test AUC in the different cases presented for the TADPOLE dataset.

For the synthetic dataset, the linear SVM achieves the best performance. Our algorithm outperforms the random forest by 8%. For the TADPOLE dataset, the architecture proposed outperforms by 2.9% the sRGCNN by Vivar et al. [96]. On the

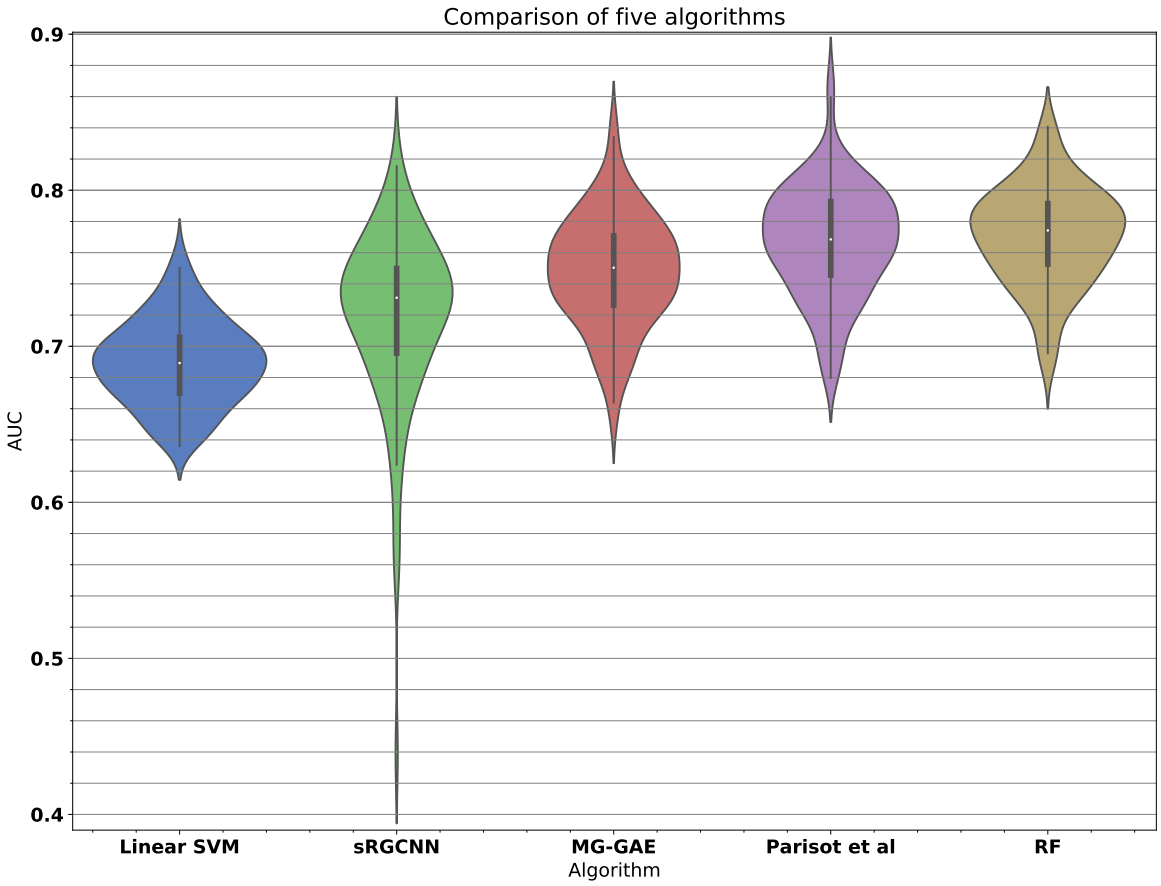


FIGURE 5.4: Violin plots of the distribution of the AUC over the 100 different train/validation/test initializations for linear SVM, sRGCNN, MG-GAE, the GCNN-based algorithm designed by Parisot et al. and random forest.

Parameter	**Value**
Split train/validation/test	0.6/0.2/0.2
Number of iterations	500

TABLE 5.6: Table of fixed hyperparameters to run each different algorithm.

matrix completion task for recommender systems, Van den Berg et al. [38] achieve better results with the GC-MC than Monti et al. [36] with the sRGCNN. However, the MG-GAE is being outperformed by the random forest by 2.3%. This may be due to the fact that there are only 21% of missing values and using the mean of all known features to impute this feature is not impacting the classification result for the random forest. Moreover, we might need to have more subjects to train the neural network.

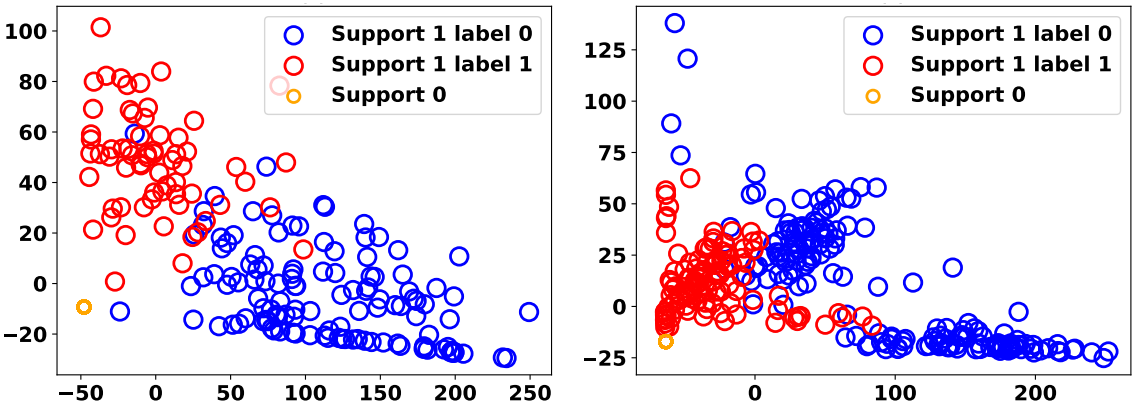
5.4.3 Vizualization of the embeddings

We have 23 support matrices that represent the data. They are defined in Section 5.3.2. For each one of these support matrices, we have o outputs of the GCN layer to encode something relevant from the support matrix. o is the subject-specified embedding dimension. We want to check if the output of the GCN layer actually takes into account the attribute of the subject that is supposed to be taken into account in the support matrix.

In order to do such a check, we apply principal component analysis (PCA) on the o outputs. We take into account the two principal components of PCA in order to see if there are differences in the subjects belonging to different groups. We plot the results in a scatter plot where the x axis represents the first component of PCA and the y axis represents the second component of PCA. We have 23 plots, one per support matrix. The group "Support 1" corresponds to the group of subjects that have the attribute that is under study. For example, if we want to inspect the embedding for men, "Support 1" is composed of all the men. We split this group into two subgroups to see if the embedding takes into account the two different classification results (MCIc or MCInc). The group "Support 0" is composed of all the other subjects that do not have the attribute. In all the plots, we can clearly see that the two classes are well separated and that the embeddings accurately take into account the known subject's values to create a boundary between MCIc subjects and MCInc subjects.

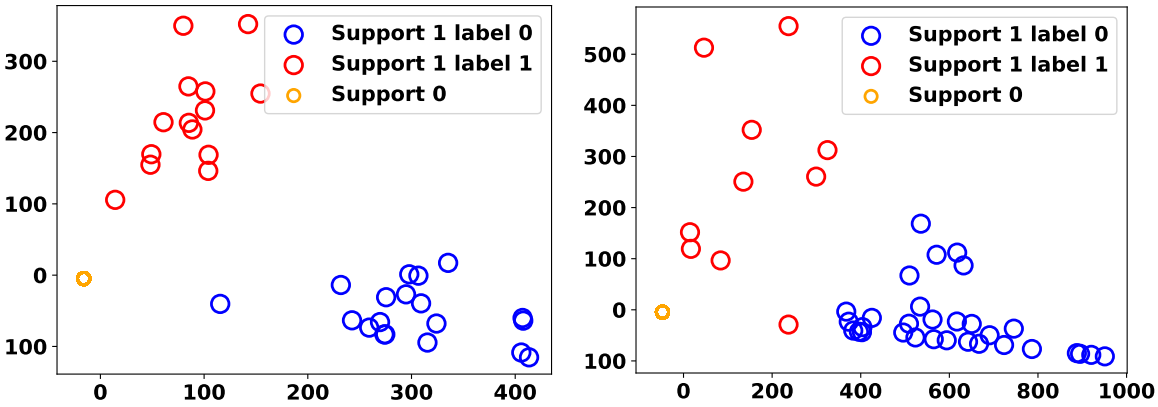
The embeddings are computed on the training data. For the TADPOLE dataset, $o = 99$. The MG-GAE was run with the hyperparameters from Table 5.3 and an AUC of 0.998 was obtained for the training set. We took the results at the best validation AUC, which was 0.785. The test AUC was 0.776. We only include 6 plots, two for sex-related embeddings (Fig. 5.5), two for age-related embeddings (Fig. 5.6) and two for age & sex-related embeddings (Fig. 5.7). The other plots are in Appendix B.

We can see that the different groups do not overlap too much and that the boundaries between the different groups are visible in every case. The GCN layer is doing what we want it to do, i.e., it is differentiating the subjects with the attribute(s) taken to built the support matrix and it is also giving a different embedding based on the label. If a subject does not belong to the group that the embedding built is based on then no embedding is computed for that subject. Considering the difference in the embedding and the splitting visible on the plots, it makes sense that a good AUC value is reached. The plots obtained for the synthetic dataset are available in Appendix C where the same conclusion can be drawn.



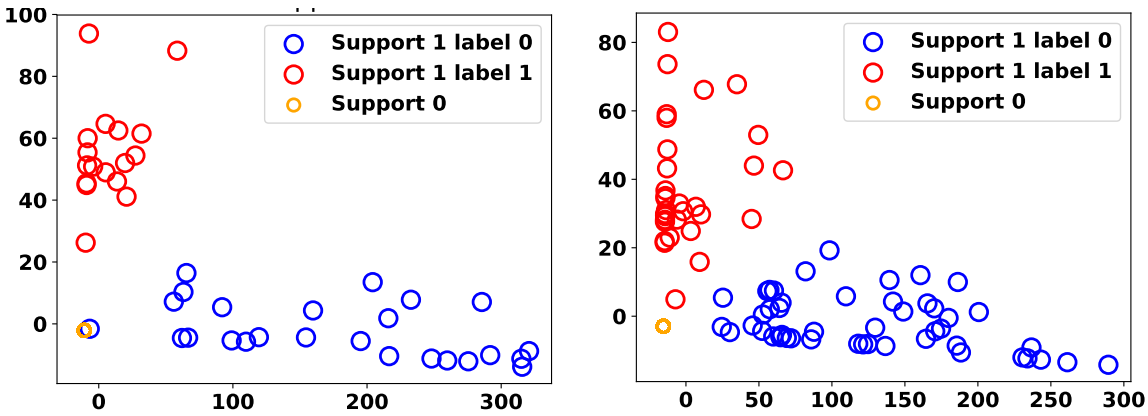
(A) Women. Support 1: Women, Support 0: Men. (B) Men. Support 1: Men, Support 0: Women

FIGURE 5.5: Scatter plots of the two first components of PCA for the sex-related embeddings.



(A) Subjects from 84 to 92 in support 1. (B) Subjects from 59 to 64 in support 1.

FIGURE 5.6: Scatter plots of the two first components of PCA for the age-related embeddings.



(A) Women from 69 to 74 in support 1. (B) Men from 69 to 74 in support 1.

FIGURE 5.7: Scatter plots of the two first components of PCA for the age & sex-related embeddings.

5.5 Conclusion

We introduce the multiple-graph graph auto-encoder, an architecture based on a graph auto-encoder to predict disease outcomes for datasets with missing values. It uses multiple bipartite graphs built on subjects' characteristics to create embeddings for each subject and each feature based on the known subject's value for each feature and on the relationships between features and subjects. This leads to an improvement of 3% on the mean AUC compared to the sRGCNN from Vivar et al. [96]. This novel architecture is also outperforming the MG-RGCNN architecture from Chapter 4 by 1%. However, it is being outperformed by the random forest and the GCNN-based algorithm designed by Parisot et al. [103], architectures where the missing values are imputed by a mean of the known values for this feature. This could be due to the fact that the dataset is not large and there are only 21% of missing values which is not enough to interfere with the classification results. Moreover, the AUC on the training set is close to 1 which is probably indicative of overfitting. The architecture is perhaps too powerful for the provided data. One way to alleviate this might be to introduce perturbances in the data.

6 Conclusion

We proposed in this thesis two novel graph-based approach to predict disease outcomes.

The first approach, the multiple-graph recurrent graph convolutional neural network, is novel because it uses multiple graphs based on the feature dependencies with attributes of the subjects (such as age or sex). The feature dependencies are computed by fitting a generalized linear model (GLM) to the known values of the studied feature. Features are then grouped according to their dependencies on the attributes. Then, multiple graphs are built, each one based on some attribute(s), where each subject is at one node of the graph. Each node is characterized by the subject's values of the features that are dependent on the attribute(s) that the graph is built on. The edge between two subjects represents the closeness in the attribute(s). An iterative process composed of multiple graph convolutional neural networks (GCNNs) and an LSTM is used to update the initial matrix to find the missing values in the features and in the labels.

The second approach, the multiple-graph graph auto-encoder, is the first architecture based on a graph auto-encoder to predict disease outcomes. It uses bipartite graphs between subjects and features constructed according to the characteristics of the subjects where the edges equal the subject's value of the feature. Then, these bipartite graphs help in building embeddings and a decoder is used to reconstruct all entries of the initial matrix, missing and non-missing.

We tested these two algorithms on the task of prediction of conversion from MCI to AD with the TADPOLE dataset, a matrix composed of 779 subjects and 564 features. It has 21% of missing values. Both methods outperform by respectively 2% and 3% the AUC of the sRGCNN by Vivar et al. [96], which is the only graph-based method for prediction of conversion from MCI to AD with missing data.

However, both methods are being outperformed by the random forest and the GCNN-based algorithm designed by Parisot et al. [103]. These architectures cannot process missing values so the missing values are imputed by a mean of the known values. The achieved performance could be due to the fact that the dataset is not large and there are only 21% of missing values which is not enough to interfere with

the classification results.

In future work, we will test the algorithms when more data is missing to determine if jointly performing matrix completion and classification can improve the classification performance in scenarios where there is a large amount of missing data.

We can also use more attributes of subjects to build the graphs such as the education level of subjects or mental test results.

As the results from the GCNN-based algorithm designed by Parisot et al. [103] are better than the results we obtained, it would be worth trying to move the architecture from Parisot et al. to a multiple graph architecture to see the performance.

Moreover, instead of focusing on classification and providing an output that only indicates if the subject will progress to the disease or not, we will incorporate a calibration mechanism so that the provided value represents the probability of conversion.

We will also apply the proposed multiple graph algorithm to other disease outcome datasets and explore methods for automatically choosing the attributes used to construct graphs instead of using ad hoc rules.

A Datasets for the prediction of Alzheimer's disease

Data used in this thesis were obtained from the ADNI database [66]. 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 at [117]. In particular, this work uses the TADPOLE dataset [118] constructed by the EuroPOND consortium [119] funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 666992.

A.1 TADPOLE dataset

We used in this thesis the TADPOLE dataset. More details on the dataset and on the different image modalities used are provided in Section 2.4. We will describe here the preprocessing steps.

A.1.1 Preprocessing of the TADPOLE dataset

The TADPOLE dataset is represented by a csv file with 12,741 rows, all representing a different examination for a patient at a specific time. As the data is longitudinal, several rows can be for the same subject but the scans and different tests were done at different times. Our goal is to predict the onset of Alzheimer's disease using only baseline data, so the first step was to remove the data where the visit code was different than baseline. The number of rows decreased to 1,737. Then, only the patients that were diagnosed with MCI at their baseline visit were kept. The number of rows decreased to 869. And finally, when labeling the data, we removed of the patients that converted to AD more than 48 months after the baseline and the patients that were wrongly labeled. The number of rows decreased to 779. The initial number of columns was 1,919. The first step was to remove the columns that did not correspond to medical examination data (age, sex, date, diagnosis, patient number...). Then, the columns where too much information was missing were also removed: if more than half of the values were missing, the column is removed. At

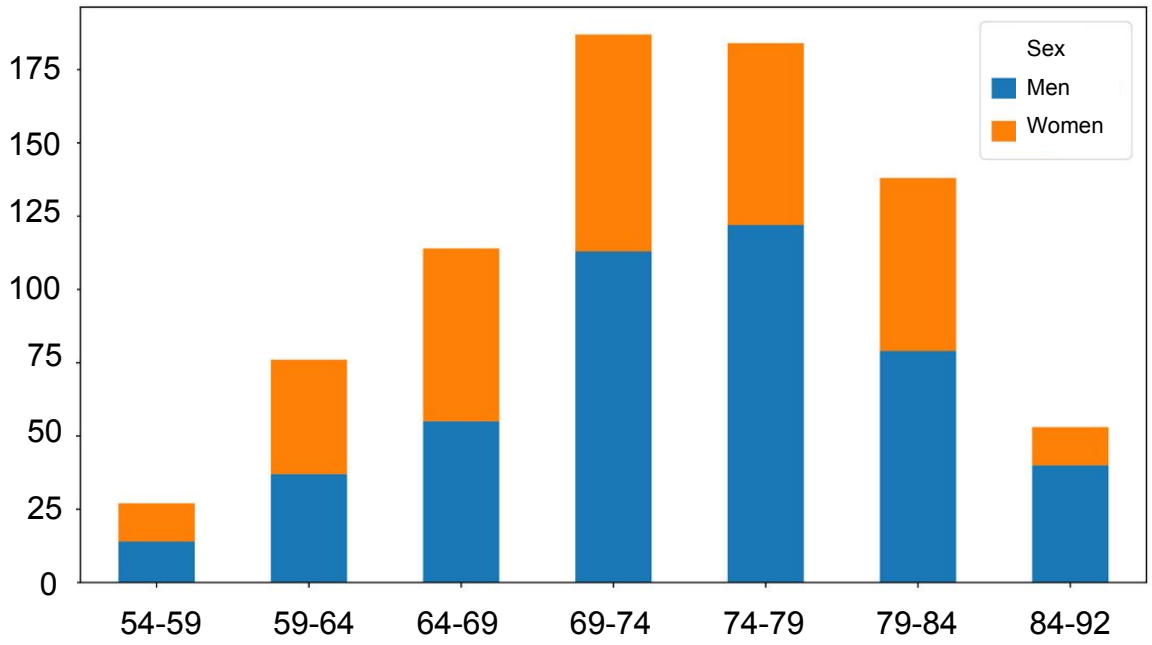


FIGURE A.1: Histograms of the number of men and women in each age group.

the end, the number of columns is 563. Our feature matrix has a shape 779×563. For the included subjects, 20.7% of feature values are missing. The characteristics of the TADPOLE dataset are given in Table A.1. A histogram with the repartition of women and men per age group is given in Fig. A.1.

The labels (MCIc and MCInc) are not given in the TADPOLE dataset. We compute the labels from the different diagnosis of a subject. In the TADPOLE dataset, we had to take into account two columns: DX_bl and DX_CHANGE. The first one represents the baseline diagnosis and the second one the change in the diagnosis over the visits. From seeing how the values of DX_CHANGE evolve over the visits, we can classify the subject as MCI converter (MCIc) or MCI non converter (MCInc). The number of MCIc is 296 and the number of MCInc is 483. The data is unbalanced.

Age	54-59	59-64	64-69	69-74	74-79	79-84	84-92	Total
Men	14	37	55	113	122	79	40	460
Women	13	39	59	74	62	59	13	319
Total	27	76	114	187	184	138	53	779

TABLE A.1: Characteristic of the subjects for the TADPOLE dataset.

A.2 Synthetic dataset

In order to conduct a more controlled analysis of the performance of the algorithm, we created a synthetic dataset. The synthetic dataset is constructed with specified relationships between the features and the characteristics of the subjects (age, sex, age&sex). The conversion probability is constructed to depend on a subset of the features.

A.2.1 Creation of the synthetic dataset

Let $M \in \mathbb{R}^{m \times n}$ be the synthetic dataset and $M(i,j) \in \mathbb{R}$ be the feature value for subject i and feature j. Let x_i denote the age, s_i denote the sex, and y_i denote the disease conversion variable for subject i. We draw random values for these subject attributes using the following distributions:

$$x_i \sim \mathcal{U}[x_{min}, x_{max}], \tag{A.1}$$

$$s_i \sim \mathcal{B}e(0.5), \tag{A.2}$$

$$y_i \sim \mathcal{B}e(0.5). \tag{A.3}$$

Let p be the number of features created, $p \gg m$. Features can be age related, sex related or age & sex related. At each creation of a feature j, we randomly choose a dependence for the feature. Let $d_j \in$ [age, sex, age & sex] denote the dependence variable, f_{ij}=f(d_j, x_i,s_i) $\in \mathcal{R}$ denote a variable that relates to the value of the feature dependence, m_j denote the slope and i_j denote the intercept. Let $\epsilon_{ij} \in \mathcal{R}$ denote the noise for each subject/feature pair. Then $M(i,j) = m_j f_{ij} + i_j + \epsilon_{ij}$. Then we randomly delete $p - m$ features. For each feature j, let v_j denote a random variable related to the label. For each subject i, $M(i,j) = M(i,j) + v_j * y_i$.

$$d_j \in [age, sex, age\&sex] \sim Categorical(1/3, 1/3, 1/3). \tag{A.4}$$
$$f_{ij} = x_i \text{ if } d_j = \text{age}, \tag{A.5}$$
$$= s_i \text{ if } d_j = \text{sex}, \tag{A.6}$$
$$= s_i x_i \text{ if } d_j = \text{age \& sex}. \tag{A.7}$$
$$m_j \sim \mathcal{U}[-z, z]. \tag{A.8}$$
$$i_j \sim \mathcal{U}[a, b]. \tag{A.9}$$
$$\epsilon_{ij} \sim \mathcal{N}(0, \sigma). \tag{A.10}$$
$$v_j \sim \mathcal{U}[c, d]. \tag{A.11}$$

A.2.2 Implementation

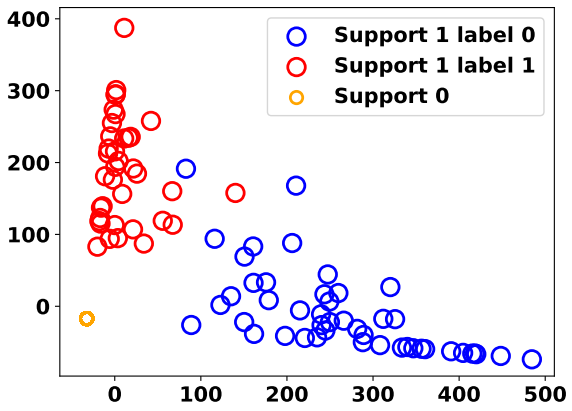
We choose the following values for the parameters of the synthetic dataset.

Parameter	x_{min}	x_{max}	p	z	a	b	σ	c	d
Value	64	92	1000	0.1	0	1	5	0	0.25

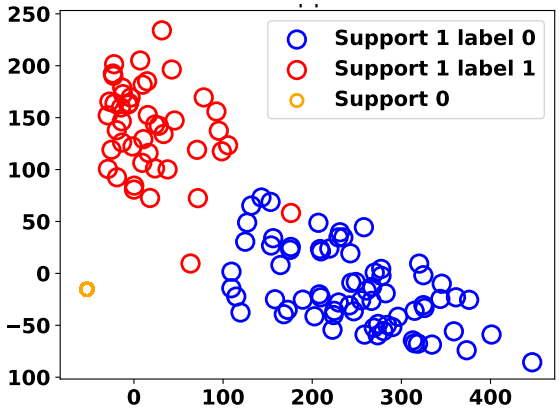
TABLE A.2: Table of parameters for the synthetic dataset.

B Vizualization of the embeddings on the TADPOLE dataset.

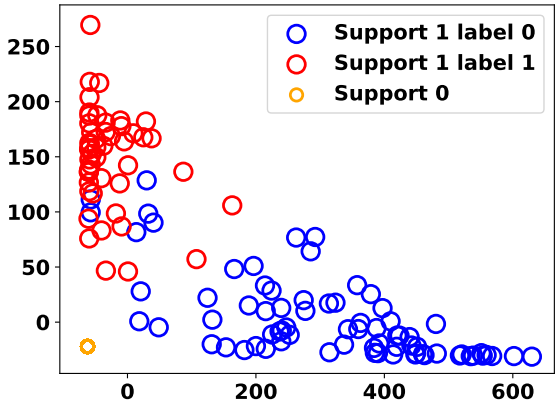
Here are the other scatter plots of the two first components of PCA on the TADPOLE dataset for the age-related embeddings (Fig. B.1) and for the age & sex-related embeddings (Fig. B.2).



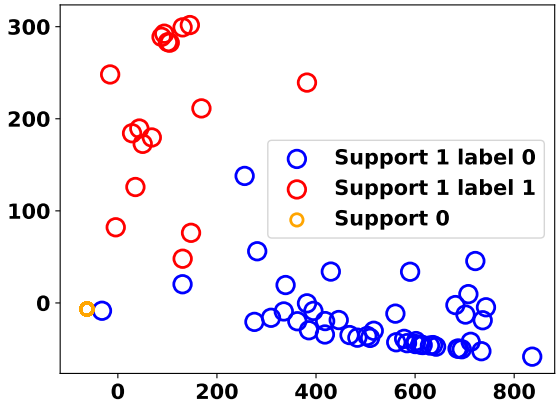
(A) Subjects from 79 to 84 in support 1.



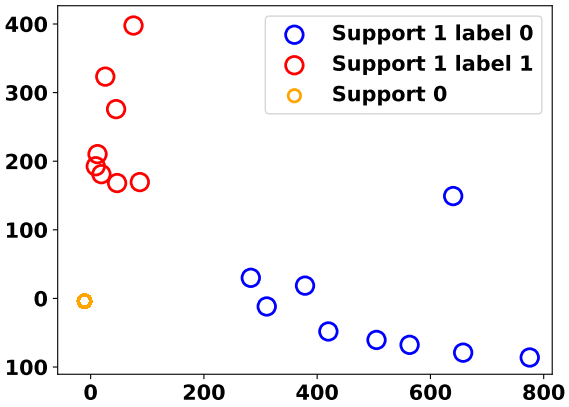
(B) Subjects from 74 to 79 in support 1.



(C) Subjects from 69 to 74 in support 1.

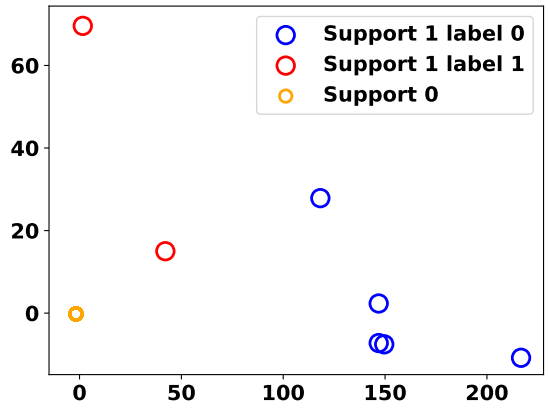


(D) Subjects from 64 to 69 in support 1.

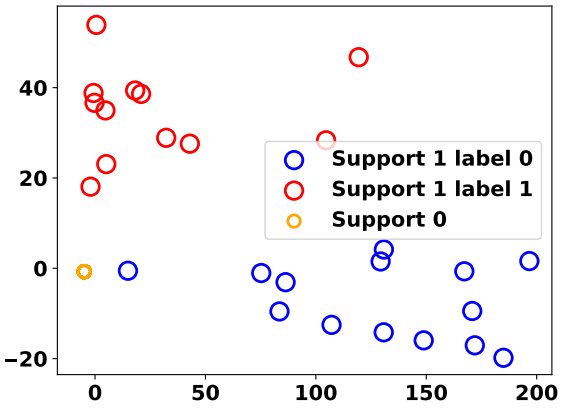


(E) Subjects from 54 to 59 in support 1.

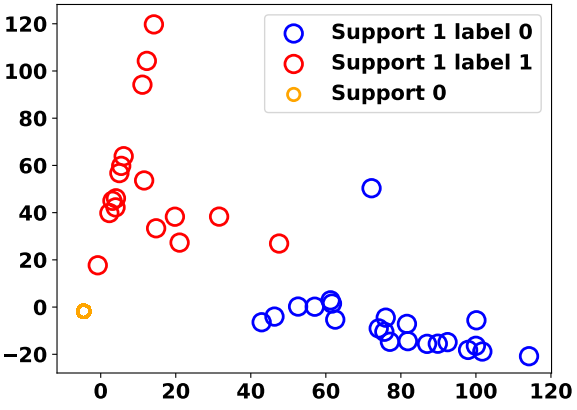
FIGURE B.1: Scatter plots of the two first components of PCA for the age-related embeddings.



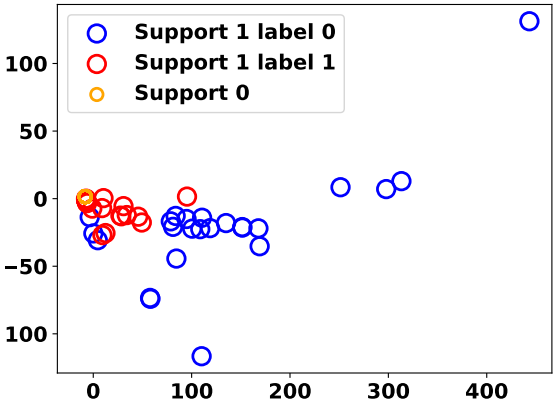
(A) Women from 84 to 92 in support 1.



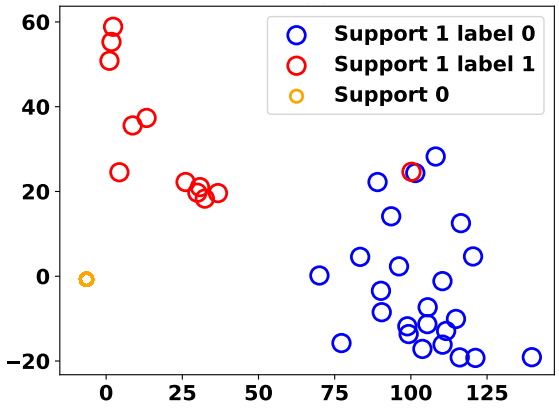
(B) Men from 84 to 92 in support 1.



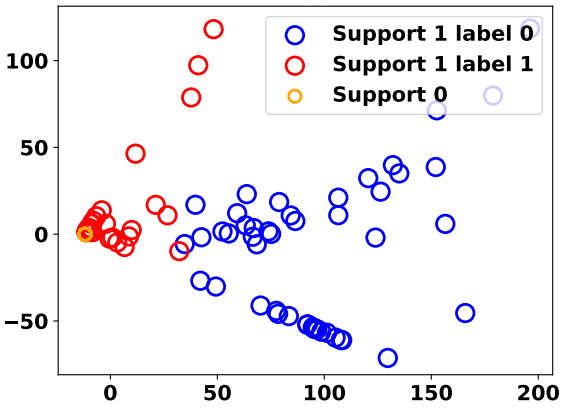
(C) Women from 79 to 84 in support 1.



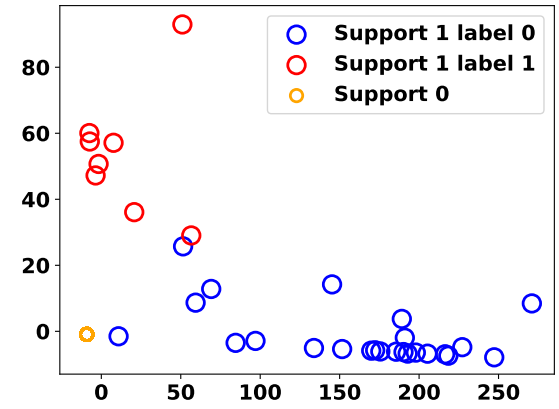
(D) Men from 79 to 84 in support 1.



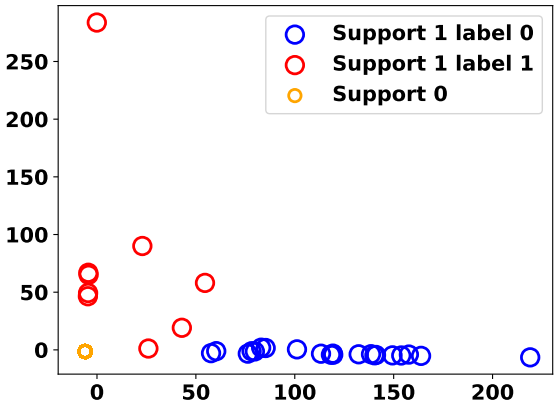
(E) Women from 74 to 79 in support 1.



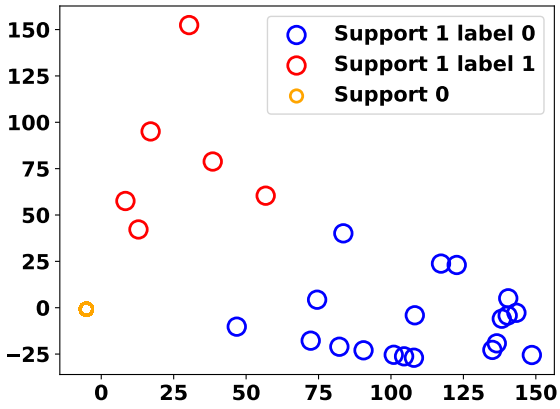
(F) Men from 74 to 79 in support 1.



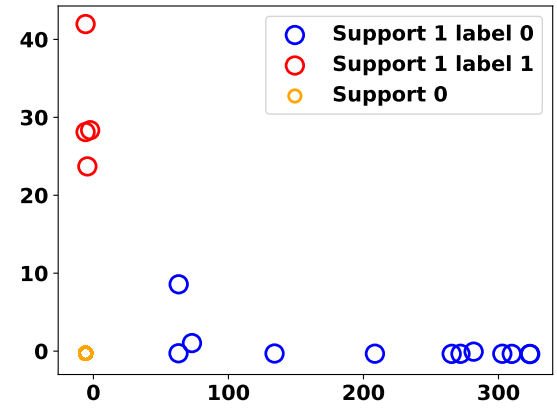
(G) Women from 64 to 69 in support 1.



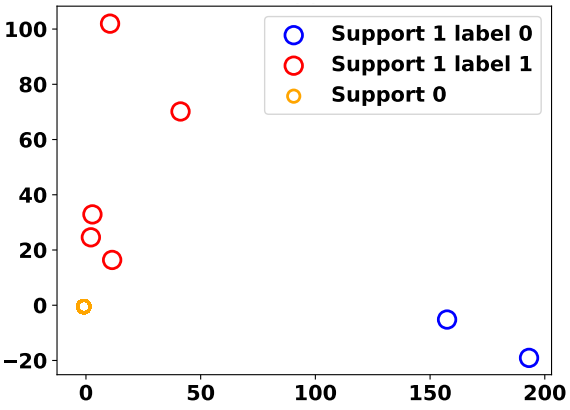
(H) Men from 64 to 69 in support 1.



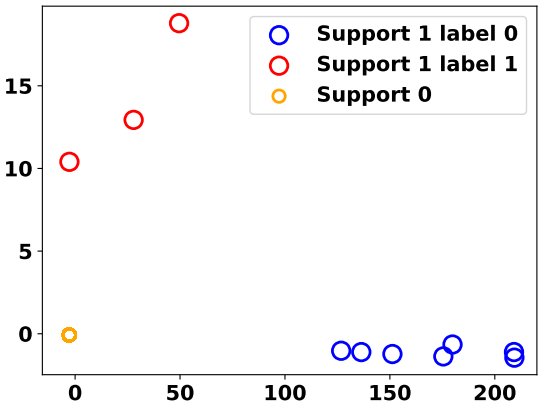
(I) Women from 59 to 64 in support 1.



(J) Men from 59 to 64 in support 1.



(K) Women from 54 to 59 in support 1.

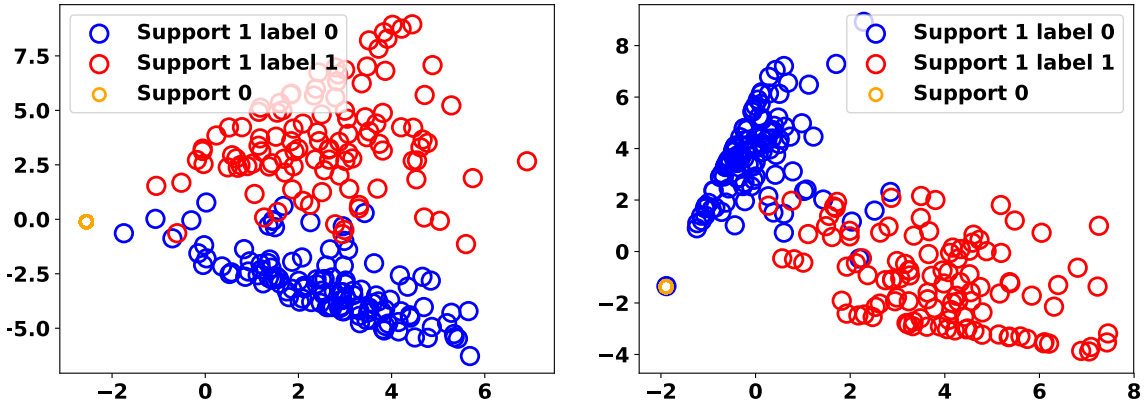


(L) Men from 54 to 59 in support 1.

FIGURE B.2: Scatter plots of the two first components of PCA for the age & sex-related embeddings.

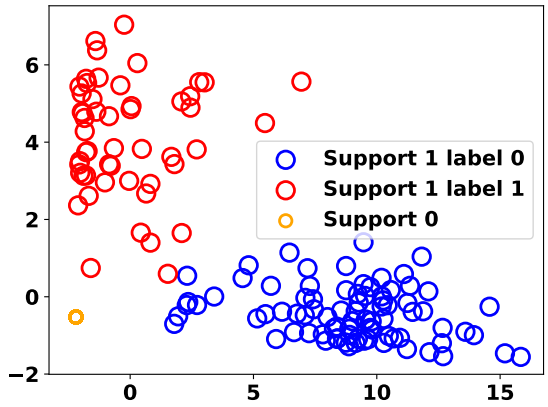
C Vizualization of the embeddings on the synthetic dataset.

For the synthetic dataset, $o = 61$. Fig. C.1, C.2 and C.3 highlight the results for the synthetic dataset where the architecture was run with the hyperparameters from Table 5.2 and an AUC of 1 was obtained for the training set. We took the results at the best validation AUC, which was 0.86. The test AUC was 0.86.

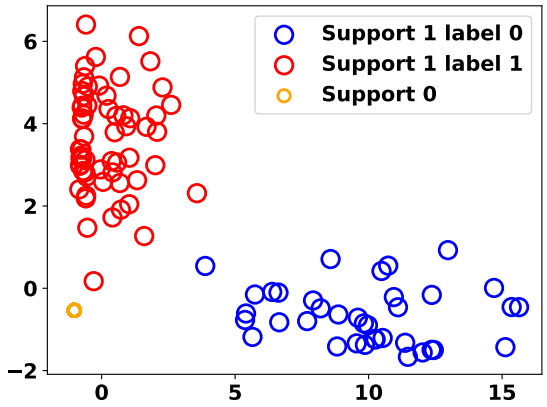


(A) Women. Support 1: Women, Support 0: Men. (B) Men. Support 1: Men, Support 0: Women

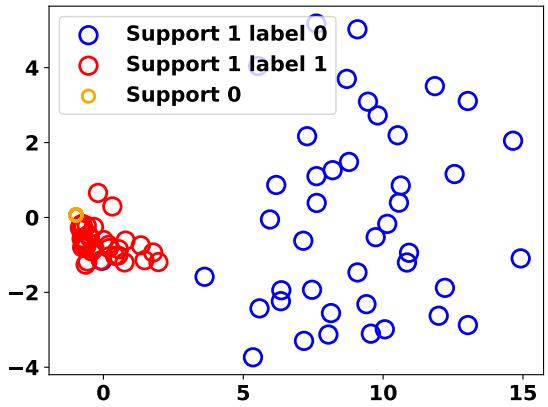
FIGURE C.1: Scatter plots of the two first components of PCA for the sex-related embeddings.



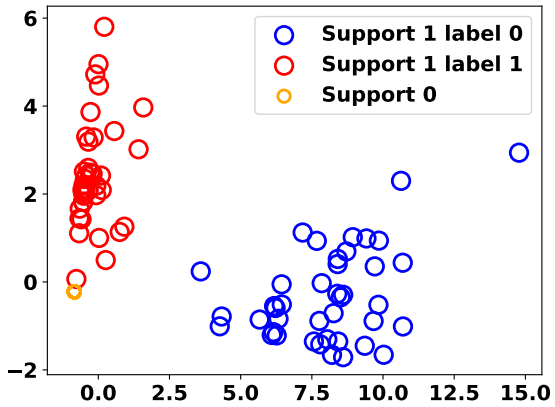
(A) Subjects from 84 to 92 in support 1.



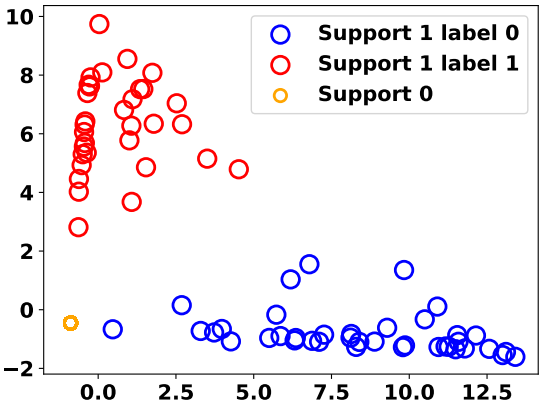
(B) Subjects from 79 to 84 in support 1.



(C) Subjects from 74 to 79 in support 1.

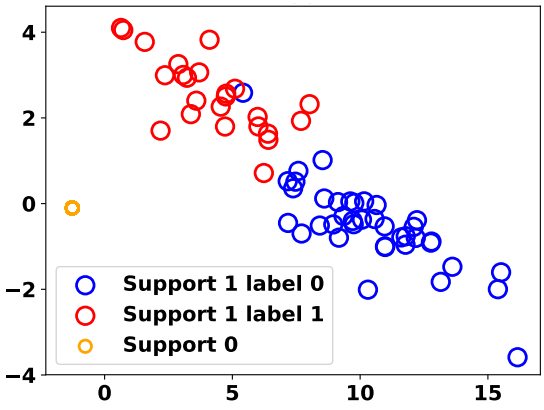


(D) Subjects from 69 to 74 in support 1.

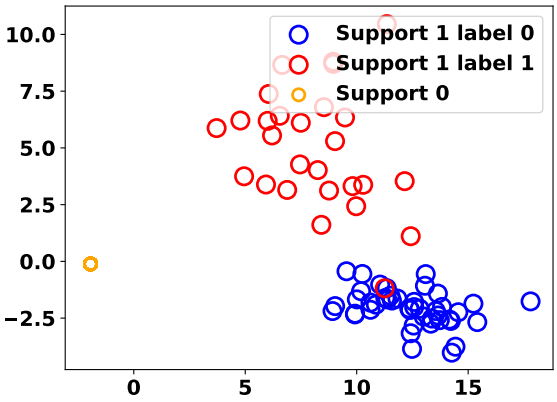


(E) Subjects from 64 to 69 in support 1.

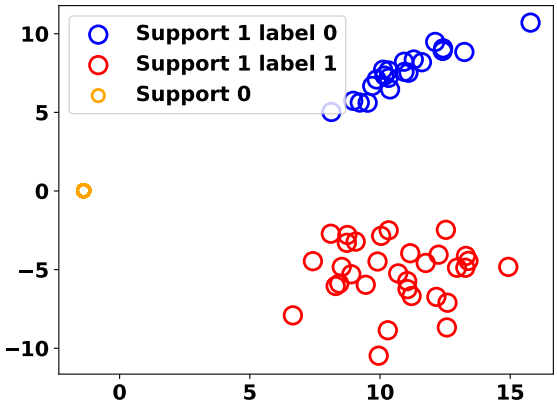
FIGURE C.2: Scatter plots of the two first components of PCA for the age-related embeddings.



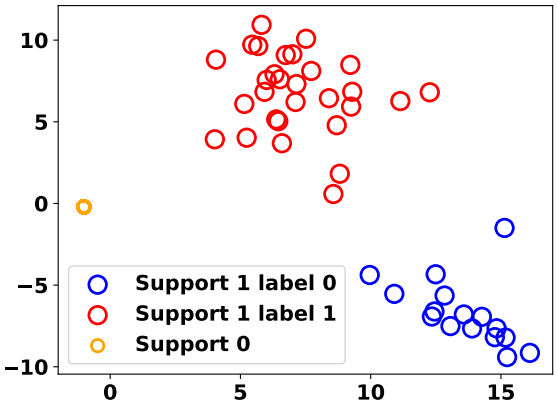
(A) Women from 84 to 92 embedding.



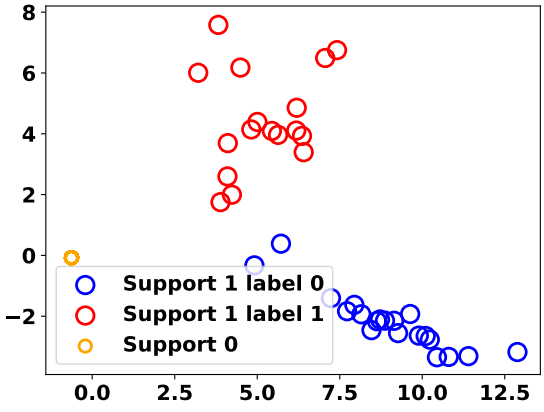
(B) Men from 84 to 92 embedding.



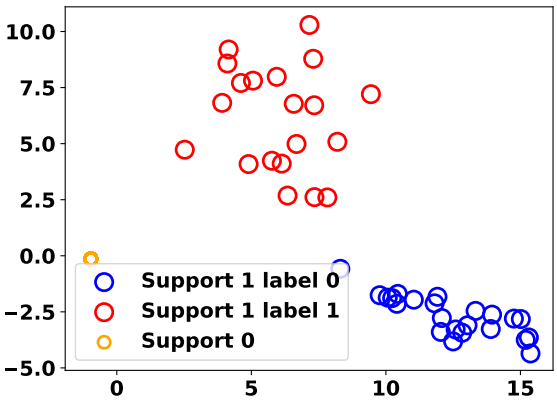
(C) Women from 79 to 84 embedding.



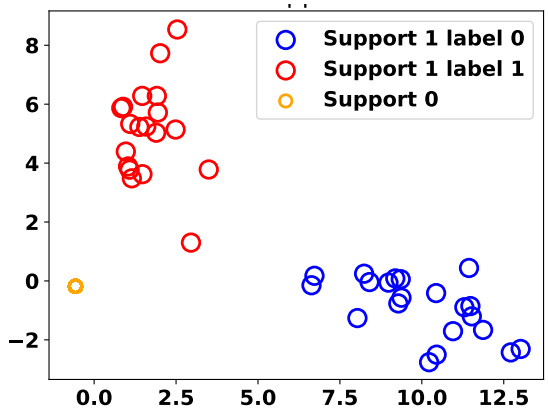
(D) Men from 79 to 84 embedding.



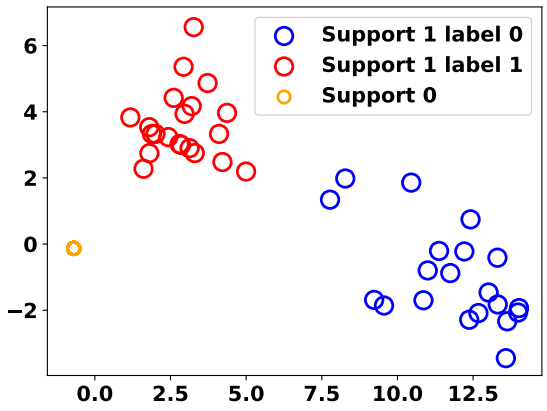
(E) Women from 74 to 79 embedding.



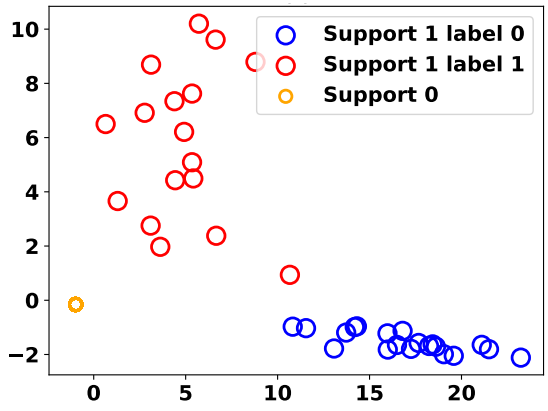
(F) Men from 74 to 79 embedding.



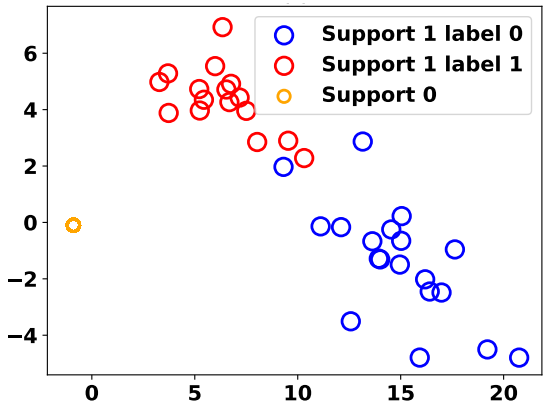
(G) Women from 69 to 74 embedding.



(H) Men from 69 to 74 embedding.



(I) Women from 64 to 69 embedding.



(J) Men from 64 to 69 embedding.

FIGURE C.3: Scatter plots of the two first components of PCA for the age & sex-related embeddings.

Bibliography

[1] F. H. Study, *Cardiovascular disease (10-year risk)*. [Online]. Available: https://www.framinghamheartstudy.org/fhs-risk-functions/cardiovascular-disease-10-year-risk/.

[2] Y. Paschalidis, *How machine learning is helping us predict heart disease and diabetes*, May 30, 2017. [Online]. Available: https://hbr.org/2017/05/how-machine-learning-is-helping-us-predict-heart-disease-and-diabetes.

[3] W. Dai, T. S. Brisimi, W. G. Adams, T. Mela, V. Saligrama, and I. C. Paschalidis, "Prediction of hospitalization due to heart diseases by supervised learning methods", *Int. J. Medical Informatics*, vol. 84, no. 3, pp. 189–197, Mar. 2015.

[4] A. J. Steele, S. C. Denaxas, A. D. Shah, H. Hemingway, and N. M. Luscombe, "Machine learning models in electronic health records can outperform conventional survival models for predicting patient mortality in coronary artery disease", *Public Library of Sci. ONE*, vol. 13, no. 8, pp. 1–20, Aug. 2018.

[5] J. T. Senders, P. C. Staples, A. V. Karhade, M. M. Zaki, W. B. Gormley, M. L. Broekman, T. R. Smith, and O. Arnaout, "Machine learning and neurosurgical outcome prediction: A systematic review", *World Neurosurgery*, vol. 109, pp. 476 –486, Jan. 2018.

[6] P. Azimi, E. C. Benzel, S. Shahzadi, S. Azhari, and H. R. Mohammadi, "Use of artificial neural networks to predict surgical satisfaction in patients with lumbar spinal canal stenosis", *J. Neurosurgery: Spine*, vol. 20, no. 3, pp. 300–305, Mar. 2014.

[7] T. M. Dumont, A. I. Rughani, and B. I. Tranmer, "Prediction of symptomatic cerebral vasospasm after aneurysmal subarachnoid hemorrhage with an artificial neural network: Feasibility and comparison with logistic regression models", *World neurosurgery*, vol. 75, no. 1, pp. 57–63, Jan. 2011.

[8] H.-Y. Shi, S.-L. Hwang, K.-T. Lee, and C.-L. Lin, "In-hospital mortality after traumatic brain injury surgery: A nationwide population-based comparison of mortality predictors used in artificial neural network and logistic regression models", *J. Neurosurgery*, vol. 118, no. 4, pp. 746–752, Apr. 2013.

[9] M. Abouzari, A. Rashidi, M. Zandi-Toghani, M. Behzadi, and M. Asadollahi, "Chronic subdural hematoma outcome prediction using logistic regression and an artificial neural network", *Neurosurgical rev.*, vol. 32, no. 4, pp. 479–484, Aug. 2009.

[10] E. K. Oermann, A. Rubinsteyn, D. Ding, J. Mascitelli, R. M. Starke, J. B. Bederson, H. Kano, L. D. Lunsford, J. P. Sheehan, J. Hammerbacher, *et al.*, "Using a machine learning approach to predict outcomes after radiosurgery for cerebral arteriovenous malformations", *Scientific reports*, vol. 6, p. 21 161, Feb. 2016.

[11] P. Azimi and H. R. Mohammadi, "Predicting endoscopic third ventriculostomy success in childhood hydrocephalus: An artificial neural network analysis", *J. Neurosurgery: Pediatrics*, vol. 13, no. 4, pp. 426–432, Apr. 2014.

[12] K. E. Emblem, B. Nedregaard, J. K. Hald, T. Nome, P. Due-Tonnessen, and A. Bjornerud, "Automatic glioma characterization from dynamic susceptibility contrast imaging: Brain tumor segmentation using knowledge-based fuzzy clustering", *J. Magnetic Resonance Imaging*, vol. 30, no. 1, pp. 1–10, Jul. 2009.

[13] K. E. Emblem, M. C. Pinho, F. G. Zöllner, P. Due-Tonnessen, J. K. Hald, L. R. Schad, T. R. Meling, O. Rapalino, and A. Bjornerud, "A generic support vector machine model for preoperative glioma survival associations", *Radiology*, vol. 275, no. 1, pp. 228–234, Apr. 2015.

[14] *Alzheimer's Disease International - dementia statistics*. [Online]. Available: https://www.alz.co.uk/research/statistics.

[15] N. I. H. National Institute on Aging, *How is Alzheimer's disease diagnosed?* [Online]. Available: https://www.nia.nih.gov/health/how-alzheimers-disease-diagnosed.

[16] A. Association, *Mild Cognitive Impairment (MCI)*, May 22, 2017. [Online]. Available: https://www.alz.org/alzheimers-dementia/what-is-dementia/related_conditions/mild-cognitive-impairment.

[17] T. P. Garcia and K. Marder, "Statistical approaches to longitudinal data analysis in neurodegenerative diseases: Huntington's disease as a model", *Current Neurology and Neuroscience Reports*, vol. 17, no. 2, p. 14, Feb. 2017.

[18] J. A. Nelder and R. W. M. Wedderburn, "Generalized linear models", *J. Royal Statistical Society. Series A (General)*, vol. 135, no. 3, pp. 370–384, 1972.

[19] S. Zeger and K. Liang, "An overview of methods for the analysis of longitudinal data", *Statist. in Medicine*, vol. 11, no. 14-15, pp. 1825–1839, Oct. 1992.

[20] R. D. Gibbons, D. Hedeker, and S. DuToit, "Advances in analysis of longitudinal data", *Annu. Rev. Clinical Psychology*, vol. 6, no. 1, pp. 79–107, 2010.

[21] N. M. Laird and J. H. Ware, "Random-effects models for longitudinal data", *Biometrics*, vol. 38, no. 4, pp. 963–974, Dec. 1982.

[22] M. Davidian, *Nonlinear models for repeated measurement data*. Routledge, Nov. 2017.

[23] C. Czado, *Lecture 10: Linear mixed models (linear models with random effects)*. [Online]. Available: https://www2.stat.duke.edu/~sayan/Sta613/2017/lec/LMM.pdf.

[24] M. Davidian and D. M. Giltinan, "Nonlinear models for repeated measurement data: An overview and update", *J. agricultural, biological & environmental statist.*, vol. 8, no. 4, p. 387, Dec. 2003.

[25] F. A. Quintana, W. O. Johnson, L. E. Waetjen, and E. B. Gold, "Bayesian nonparametric longitudinal data analysis", *J. American Statistical Association*, vol. 111, no. 515, pp. 1168–1181, 2016.

[26] R. I. Jennrich and S. M. Robinson, "A Newton-Raphson algorithm for maximum likelihood factor analysis", *Psychometrika*, vol. 34, no. 1, pp. 111–123, Mar. 1969.

[27] M. J. Lindstrom and D. M. Bates, "Newton-Raphson and EM algorithms for linear mixed-effects models for repeated-measures data", *J. American Statistical Association*, vol. 83, no. 404, pp. 1014–1022, Dec. 1988.

[28] F. Gumedze and T. Dunne, "Parameter estimation and inference in the linear mixed model", *Linear Algebra & its Appl.*, vol. 435, no. 8, pp. 1920–1944, Oct. 2011.

[29] L. Wu, X. J. Hu, and H. Wu, "Joint inference for nonlinear mixed-effects models and time to event at the presence of missing data", *Biostatistics*, vol. 9, no. 2, pp. 308–320, Apr. 2008.

[30] V. H. Lachos, L. M. Castro, and D. K. Dey, "Bayesian inference in nonlinear mixed-effects models using normal independent distributions", *Computational Statist. & Data Anal.*, vol. 64, pp. 237–252, Aug. 2013.

[31] K. Gao and A. B. Owen, "Estimation and inference for very large linear mixed effects models", *arXiv preprint arXiv:1610.08088*, 2016.

[32] Z. Tan, K. Roche, X. Zhou, and S. Mukherjee, "Scalable algorithms for learning high-dimensional linear mixed models", in *Proc. 34th Conf. Uncertainty in Artificial Intell.*, Monterey, CA, USA, Aug. 2018, pp. 259–268.

[33] P. O. Perry, "Fast moment-based estimation for hierarchical models", *J. Royal Statistical Society: Series B (Statistical Methodology)*, vol. 79, no. 1, pp. 267–291, Jan. 2017.

[34] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization", *Foundations Computational Math.*, vol. 9, no. 6, p. 717, Apr. 2009.

[35] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Matrix completion on graphs", in *Proc. Advances Neural Inform. Process. Syst. 27, Out of the Box: Robustness in High Dimension Workshop*, Montreal, QC, Canada, Dec. 2014.

[36] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks", in *Proc. Advances Neural Inform. Process. Syst. 30*, Long Beach, CA, USA, Dec. 2017, pp. 3697–3707.

[37] A. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang, and Y. Li, "A survey of matrix completion methods for recommendation systems", *Big Data Mining & Analytics*, vol. 1, no. 4, pp. 308–323, Dec. 2018.

[38] R. van den Berg, T. Kipf, and M. Welling, "Graph convolutional matrix completion", in *Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining, Deep Learning Day*, London, UK, Aug. 2018.

[39] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering", in *Proc. Advances Neural Inform. Process. Syst. 29*, Barcelona, Spain, Dec. 2016, pp. 3844–3852.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Proc. Advances Neural Inform. Process. Syst. 25*, Lake Tahoe, NV, USA, Dec. 2012, pp. 1097–1105.

[41] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups", *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[42] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features", in *Proc. 11th European Conf. Comput. Vision*, Heraklion, Crete, Greece, Sep. 2010, pp. 140–153.

[43] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains", *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[44] F. R. Chung and F. C. Graham, *Spectral graph theory*, 92. American Mathematical Soc., 1997.

[45] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs", in *Proc. 1st Int. Conf. Learning Representations*, Scottsdale, AZ, USA, May 2013.

[46] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data", *arXiv:1506.05163*, 2015.

[47] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks", in *Proc. 5th Int. Conf. Learning Representations*, Toulon, France, Apr. 2017.

[48] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: Graph convolutional neural networks with complex rational spectral filters", *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 97–109, Jan. 2019.

[49] F. Monti, K. Otness, and M. M. Bronstein, "Motifnet: A motif-based graph convolutional network for directed graphs", in *IEEE Data Sci. Workshop*, Lausanne, Switzerland, Jun. 2018, pp. 225–228.

[50] S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation", in *Proc. Advances Neural Inform. Process. Syst. 29*, Dec. 2016, pp. 2244–2252.

[51] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling", in *Proc. Conf. Empirical Methods in Natural Language Process.*, Copenhagen, Denmark, Sep. 2017, pp. 1506–1515.

[52] X. Bresson and T. Laurent, "Residual gated graph convnets", in *Proc. 6th Int. Conf. Learning Representations*, Vancouver, BC, Canada, May 2018.

[53] R. Anirudh and J. J. Thiagarajan, "Bootstrapping graph convolutional neural networks for autism spectrum disorder classification", in *Proc. IEEE 44th Int. Conf. Acoust., Speech & Signal Process.*, Brighton, UK, May 2019.

[54] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory", *Appl. & Computational Harmonic Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.

[55] J Atwood and D Towsley, "Diffusion-convolutional neural networks", in *Proc. Advances Neural Inform. Process. Syst. 29*, Dec. 2016, pp. 1993–2001.

[56] Y. Hechtlinger, P. Chakravarti, and J. Qin, "A generalization of convolutional neural networks to graph-structured data", in *Proc. 5th Int. Conf. Learning Representations*, Toulon, France, Apr. 2017.

[57] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs", in *Proc. IEEE Conf. Comput. Vision & Pattern Recognition*, Honolulu, US, Jul. 2017, pp. 5425–5434.

[58] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs", in *Proc. IEEE Conf. Comput. Vision & Pattern Recognition*, Honolulu, USA, Jul. 2017, pp. 29–38.

[59] F. P. Such, S. Sah, M. A. Dominguez, S. Pillai, C. Zhang, A. Michael, N. D. Cahill, and R. Ptucha, "Robust spatial filtering with graph convolutional neural networks", *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 884–896, Sep. 2017.

[60] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs", in *Proc. 33rd Int. Conf. Mach. Learning*, vol. 48, New York, NY, USA, Jun. 2016, pp. 2014–2023.

[61] D. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints", in *Proc. Advances Neural Inform. Process. Syst. 28*, Montreal, QC, Canada, Dec. 2015, pp. 2224–2232.

[62] W. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs", in *Proc. Advances Neural Inform. Process. Syst. 30*, Long Beach, CA, USA, Dec. 2017, pp. 1024–1034.

[63] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling", in *Proc. 6th Int. Conf. Learning Representations*, Vancouver, BC, Canada, May 2018.

[64] N. Verma, E. Boyer, and J. Verbeek, "Feastnet: Feature-steered graph convolutions for 3D shape analysis", in *Proc. IEEE Conf. Comput. Vision & Pattern Recognition*, Salt Lake City, UT, USA, Jun. 2018, pp. 2598–2606.

[65] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems", in *Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, London, UK, Aug. 2018, pp. 974–983.

[66] *Adni | alzheimer's disease neuroimaging initiative*. [Online]. Available: adni.loni.usc.edu.

[67] *Tadpole - data*. [Online]. Available: https://tadpole.grand-challenge.org/Data/.

[68] C. R. Jack Jr and D. M. Holtzman, "Biomarker modeling of Alzheimer's disease", *Neuron*, vol. 80, no. 6, pp. 1347–1358, Dec. 2013.

[69] C. developers, *Volume pre-processing - clinica documentation*. [Online]. Available: http://www.clinica.run/doc/Pipelines/T1_Volume/.

[70] F. Mahmoudi, K. Elisevich, H. Bagher-Ebadian, M.-R. Nazem-Zadeh, E. Davoodi-Bojd, J. M. Schwalb, M. Kaur, and H. Soltanian-Zadeh, "Data mining mr image features of select structures for lateralization of mesial temporal lobe epilepsy", *Public Library of Sci. ONE*, vol. 13, no. 8, pp. 1–19, Aug. 2018.

[71] *Freesurfer*. [Online]. Available: https://surfer.nmr.mgh.harvard.edu/.

[72] *Brain-life/app-freesurfer*. [Online]. Available: https://github.com/brain-life/app-freesurfer.

[73] G. M Ashraf, N. H Greig, T. A Khan, I. Hassan, S. Tabrez, S. Shakil, I. A Sheikh, S. K Zaidi, M. Akram, N. R Jabir, *et al.*, "Protein misfolding and aggregation in Alzheimer's disease and type 2 diabetes mellitus", *CNS & Neurological Disorders-Drug Targets*, vol. 13, no. 7, pp. 1280–1293, 2014.

[74] D. Flöck, S. Colacino, G. Colombo, and A. Di Nola, "Misfolding of the amyloid β-protein: A molecular dynamics study", *Proteins: Structure, Function, and Bioinformatics*, vol. 62, no. 1, pp. 183–192, Jan. 2006.

[75] M. Murphy and H LeVine III, "Alzheimer's disease and the β-amyloid peptide", *J. Alzheimer's Disease*, vol. 19, no. 1, pp. 311–323, Jan. 2010.

[76] R. Lal, H. Lin, and A. P. Quist, "Amyloid β ion channel: 3D structure and relevance to amyloid channel paradigm", *Biochimica et Biophysica Acta - Biomembranes*, vol. 1768, no. 8, pp. 1966–1975, Aug. 2007.

[77] A. d. C. Alonso, I. Grundke-Iqbal, H. S. Barra, and K. Iqbal, "Abnormal phosphorylation of τ and the mechanism of Alzheimer neurofibrillary degeneration: Sequestration of microtubule-associated proteins 1 and 2 and the disassembly of microtubules by the abnormal τ", *Proc. National Academy of Sci. USA*, vol. 94, no. 1, pp. 298–303, Jan. 1997.

[78] S. Khatoon, I. Grundke-Iqbal, and K. Iqbal, "Levels of normal and abnormally phosphorylated τ in different cellular and regional compartments of Alzheimer disease and control brains", *Federation European Biochemical Societies Letters*, vol. 351, no. 1, pp. 80–84, Aug. 1994.

[79] K. Herholz, "PET studies in dementia", *Ann. nuclear medicine*, vol. 17, no. 2, pp. 79–89, Apr. 2003.

[80] N. Okamura, H. Arai, M. Higuchi, M. Tashiro, T. Matsui, X.-S. Hu, A. Takeda, M. Itoh, and H. Sasaki, "[18f] FDG-PET study in dementia with lewy bodies and Alzheimer's disease", *Progress in Neuro-Psychopharmacology & Biological Psychiatry*, vol. 25, no. 2, pp. 447–456, Feb. 2001.

[81] U. of California Berkeley, *PET scans reveal key details of Alzheimer's protein growth in aging brains*. [Online]. Available: https://medicalxpress.com/news/2016-03-pet-scans-reveal-key-alzheimer.html.

[82] BruceBlaus, *File:blausen 0216 cerebrospinalsystem.png - wikimedia commons*. [Online]. Available: https://commons.wikimedia.org/w/index.php?curid=31118595.

[83] M. R. Brier, B. Gordon, K. Friedrichsen, J. McCarthy, A. Stern, J. Christensen, C. Owen, P. Aldea, Y. Su, J. Hassenstab, *et al.*, "τ and aβ imaging, CSF measures, and cognition in Alzheimer's disease", *Sci. translational medicine*, vol. 8, no. 338, p. 66, May 2016.

[84] A. H. Simonsen, S.-K. Herukka, N. Andreasen, I. Baldeiras, M. Bjerke, K. Blennow, S. Engelborghs, G. B. Frisoni, T. Gabryelewicz, S. Galluzzi, *et al.*, "Recommendations for CSF AD biomarkers in the diagnostic evaluation of dementia", *Alzheimer's & Dementia*, vol. 13, no. 3, pp. 274–284, Mar. 2017.

[85] J.-H. Kang, M. Korecka, J. B. Toledo, J. Q. Trojanowski, and L. M. Shaw, "Clinical utility and analytical challenges in measurement of cerebrospinal fluid amyloid-β1–42 and τ proteins as Alzheimer disease biomarkers", *Clinical chemistry*, vol. 59, no. 6, pp. 903–916, Jun. 2013.

[86] J. L. Bernal-Rusiel, D. N. Greve, M. Reuter, B. Fischl, and M. R. Sabuncu, "Statistical analysis of longitudinal neuroimage data with linear mixed effects models", *NeuroImage*, vol. 66, pp. 249 –260, Feb. 2013.

[87] G. Ziegler, W. Penny, G. Ridgway, S. Ourselin, and K. Friston, "Estimating anatomical trajectories with bayesian mixed-effects modeling", *NeuroImage*, vol. 121, pp. 51 –68, Nov. 2015.

[88] D. Li, S. Iddi, W. Thompson, and M. Donohue, "Bayesian latent time joint mixed effect models for multicohort longitudinal data", *Statistical Methods in Medical Res.*, Nov. 2017.

[89] J.-B. Schiratti, S. Allassonnière, O. Colliot, and S. Durrleman, "Mixed-effects model for the spatiotemporal analysis of longitudinal manifold-valued data", in *Proc. 5th MICCAI Workshop Math. Foundations of Computational Anatomy*, Munich, Germany, Oct. 2015.

[90] J. L. Bernal-Rusiel, M. Reuter, D. N. Greve, B. Fischl, and M. R. Sabuncu, "Spatiotemporal linear mixed effects modeling for the mass-univariate analysis of longitudinal neuroimage data", *NeuroImage*, vol. 81, pp. 358 –370, Nov. 2013.

[91] Y. Li, J. H. Gilmore, D. Shen, M. Styner, W. Lin, and H. Zhu, "Multiscale adaptive generalized estimating equations for longitudinal neuroimaging data", *NeuroImage*, vol. 72, pp. 91–105, May 2013.

[92] X. Zhang, L. Li, H. Zhou, D. Shen, *et al.*, "Tensor generalized estimating equations for longitudinal imaging analysis", *Statistica Sinica*, 2017.

[93] S. Korolev, A. Safiullin, M. Belyaev, and Y. Dodonova, "Residual and plain convolutional neural networks for 3D brain MRI classification", in *Proc. IEEE 14th Int. Symp. Biomedical Imaging*, Melbourne, Australia, Apr. 2017, pp. 835–838.

[94] J. Arco, J. Ramírez, C. Puntonet, J. Górriz, and M. Ruz, "Improving short-term prediction from MCI to AD by applying searchlight analysis", in *Proc. IEEE. 13th Int. Symp. Biomedical Imaging*, Prague, Czech Republic, Apr. 2016, pp. 10–13.

[95] H. Choi and K. H. Jin, "Predicting cognitive decline with deep learning of brain metabolism and amyloid imaging", *Behavioural Brain Res.*, vol. 344, pp. 103–109, May 2018.

[96] G. Vivar, A. Zwergal, N. Navab, and S.-A. Ahmadi, "Multi-modal disease classification in incomplete datasets using geometric matrix completion", in *Proc. 2nd Int. Workshop Graphs Biomedical Image Analysis*, Granada, Spain, Sep. 2018, pp. 24–31.

[97] H.-I. Suk, S.-W. Lee, D. Shen, Alzheimer's Disease Neuroimaging Initiative, *et al.*, "Latent feature representation with stacked auto-encoder for AD/MCI diagnosis", *Brain Structure & Function*, vol. 220, no. 2, pp. 841–859, Mar. 2015.

[98] D. Lu, K. Popuri, G. W. Ding, R. Balachandar, and M. F. Beg, "Multimodal and multiscale deep neural networks for the early diagnosis of Alzheimer's disease using structural mr and FDG-PET images", *Scientific reports*, vol. 8, no. 1, p. 5697, Apr. 2018.

[99] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[100] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proc. 29th IEEE Conf. Comput. Vision & Pattern Recognition*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[101] J. A. Etzel, J. M. Zacks, and T. S. Braver, "Searchlight analysis: Promise, pitfalls, and potential", *Neuroimage*, vol. 78, pp. 261–269, Sep. 2013.

[102] E. Moradi, A. Pepe, C. Gaser, H. Huttunen, and J. Tohka, "Machine learning framework for early MRI-based Alzheimer's conversion prediction in MCI subjects", *NeuroImage*, vol. 104, pp. 398–412, Jan. 2015.

[103] S. Parisot, S. I. Ktena, E. Ferrante, M. Lee, R. G. Moreno, B. Glocker, and D. Rueckert, "Spectral graph convolutions for population-based disease prediction", in *Proc. 20th Int. Conf. Medical Image Computing & Comput-Assisted Intervention*, Quebec City, QC, Canada, Sep. 2017, pp. 177–185.

[104] K. Thung, E. Adeli, P. Yap, and D. Shen, "Stability-weighted matrix completion of incomplete multi-modal data for disease diagnosis", in *Proc. 19th Int. Conf. Medical Image Computing & Comput-Assisted Intervention*, Athens, Greece, Oct. 2016, pp. 88–96.

[105] A. Goldberg, B. Recht, J. Xu, R. Nowak, and X. Zhu, "Transduction with matrix completion: Three birds with one stone", in *Proc. Advances Neural Inform. Process. Syst. 23*, Vancouver, BC, Canada, Dec. 2010, pp. 757–765.

[106] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", in *Proc. 13th Int. Conf. Artificial Intell. & Statist.*, Sardinia, Italy, May 2010, pp. 249–256.

[107] C. Mazure and J. Swendsen, "Sex differences in Alzheimer's disease and other dementias", *The Lancet Neurology*, vol. 15, no. 5, pp. 451–452, Apr. 2016.

[108] J. Viña and A. Lloret, "Why women have more Alzheimer's disease than men: Gender and mitochondrial toxicity of amyloid-β peptide", *J. Alzheimer's disease*, vol. 20, no. s2, S527–S533, 2010.

[109] S. J. Ritchie, S. R. Cox, X. Shen, M. V. Lombardo, L. M. Reus, C. Alloza, M. A. Harris, H. L. Alderson, S. Hunter, E. Neilson, *et al.*, "Sex differences in the adult human brain: Evidence from 5216 UK Biobank participants", *Cerebral Cortex*, vol. 28, no. 8, pp. 2959–2975, Aug. 2018.

[110] A. N. Ruigrok, G. Salimi-Khorshidi, M.-C. Lai, S. Baron-Cohen, M. V. Lombardo, R. J. Tait, and J. Suckling, "A meta-analysis of sex differences in human brain structure", *Neuroscience & Biobehavioral Rev.*, vol. 39, pp. 34–50, Feb. 2014.

[111] R. V. Marinescu, N. P. Oxtoby, A. L. Young, E. E. Bron, A. W. Toga, M. W. Weiner, F. Barkhof, N. C. Fox, S. Klein, D. C. Alexander, *et al.*, "Tadpole challenge: Prediction of longitudinal evolution in Alzheimer's disease", *arXiv preprint arXiv:1805.03909*, Aug. 2018.

[112] K. J. Bjuland, G. C. C. Lohaugen, M. P. Martinussen, and J. Skranes, "Cortical thickness and cognition in very-low-birth-weight late teenagers.", *Early human develop.*, vol. 89, no. 6, pp. 371–380, Jun. 2013.

[113] W. Li, M.-J. van Tol, M. Li, W. Miao, Y. Jiao, H.-J. Heinze, B. Bogerts, H. He, and M. Walter, "Regional specificity of sex effects on subcortical volumes across the lifespan in healthy aging", *Human brain mapping*, vol. 35, no. 1, pp. 238–247, Jan. 2014.

[114] M. Filippi, M. A. Rocca, N. De Stefano, C. Enzinger, E. Fisher, M. A. Horsfield, M. Inglese, D. Pelletier, and G. Comi, "Magnetic resonance techniques in multiple sclerosis: The present and the future", *Archives of Neurology*, vol. 68, no. 12, pp. 1514–1520, Dec. 2011.

[115] Y. Taki, B. Thyreau, S. Kinomura, K. Sato, R. Goto, R. Kawashima, and H. Fukuda, "Correlations among brain gray matter volumes, age, gender, and hemisphere in healthy individuals", *Public Library of Sci. One*, vol. 6, no. 7, e22734, 2011.

[116] A. Costa and G. Nannicini, "Rbfopt: An open-source library for black-box optimization with costly function evaluations", *Math. Programming Computation*, vol. 10, no. 4, pp. 597–629, Dec. 2018.

[117] *Acknowledgement list for adni publications*. [Online]. Available: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

[118] *Tadpole - home*. [Online]. Available: https://tadpole.grand-challenge.org.

[119] *Europond | european progression of neurological disease initiative*. [Online]. Available: http://europond.eu/.