Prediction of Mild Cognitive Impairment Progression using Longitudinal MRI Scans

THOMAI STATHOPOULOU

Master in Machine Learning
Date: December 3, 2016
Supervisor: Kevin Smith, Farshad Falahati
Examiner: Johan Håstad
Swedish title: Detta är den svenska översättningen av titeln
School of Computer Science and Communication

Abstract

English abstract goes here.

Sammanfattning

Träutensilierna i ett tryckeri äro ingalunda en faktor där trevnadens ordningens och ekonomiens upprätthållande, och dock är det icke sällan som sorgliga erfarenheter göras ordningens och ekon och miens därmed upprätthållande. Träutensilierna i ett tryckeri äro ingalunda en oviktig faktor, för trevnadens ordningens och och dock är det icke sällan.

Contents

Abbreviations

AD Alzheimer’s Disease

ANN Artificial Neural Network

CSF Cerebrospinal Fluid

EM Expectation Maximization

HMM Hidden Markov Model

MCI Mild Cognitive Impairment

MRI Magnetic Resonance Imaging

NMR Nuclear Magnetic Resonance

OPLS Orthogonal Projection to Latent Structures

RF Radio frequency

RNN Recurrent Neural Network

SVM Support Vector Machine

Chapter 1

Introduction

This chapter is a general introduction to the subject of study of the thesis, as well as the objective of the research done throughout the thesis work.

1.1 The Ageing Brain

The brain is an extraordinary organ. It is the organ that manages our entire body and automatically coordinates all the necessary body functions, such as breathing, blood circulation and digestion. Additionally, it enables us to do all the functions we consciously do, such as walking, talking, seeing etc. It is also the organ that enables us to think and, in general, be conscious.

But even anatomically the brain is very interesting. It consists of nerve cells (**neurons**) and other types of cells. It is estimated that it contains about 100 billion neurons and 100 trillion **synapses** (gaps between the neurons, used by the neurons to communicate). Interestingly, even though the brain is only about 2% of the body weight, it receives 20% of the blood's supply, thus the demand of about 400 billion blood vessels (called **capillaries**).

Basic Anatomy

The brain is divided into different parts, all responsible for different functions.

Cerebral Hemispheres Initially the brain is divided into two **cerebral hemispheres**, most commonly referred to as left and right hemisphere, taking up almost 85% of its total weight. The two hemispheres appear to implement different functionalities. The left hemisphere seemingly focuses on details (e.g. recognizing a person in the crowd), whereas the right hemisphere focuses on broad background (e.g. understanding the relative position of objects in space). The outer layer of the hemispheres is called **cerebral cortex** and is responsible for the processing of incoming information as well as regulating cognitive functions.

Cerebellum Located at the base of the brain, the **cerebellum** takes up about 10% of the total weight. It is also divided into two hemispheres and it is responsible for the body's balance and coordination. It receives information from the eyes, ears, muscles and joints regarding the body's movements and position.

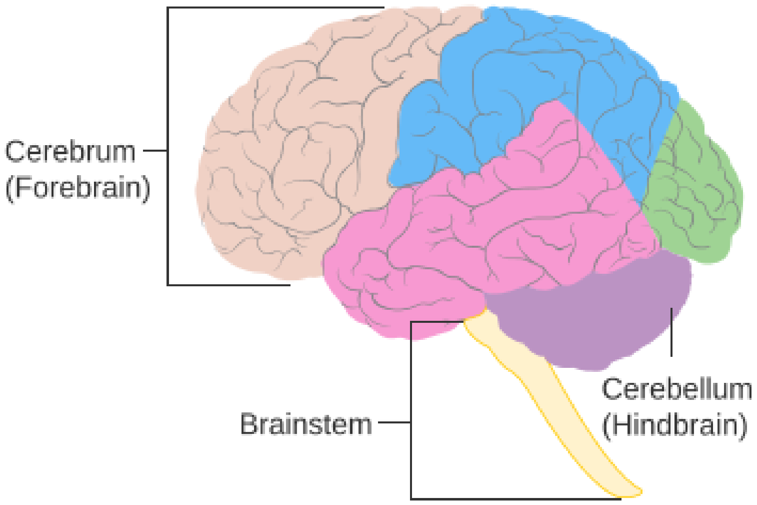


Figure 1.1: Basic Brain Anatomy from *Wikipedia*

Brain stem The **brain stem** is placed close to the cerebellum and it is the connection between the brain and the spinal cord. It is the part that controls all the unconscious functions and behaves as the messenger for signals sent between the brain and other organs.

Other Parts Within the cerebral hemispheres, we have located several other important parts, such as the hippocampus, the hypothalamus etc., all of which are responsible for different bodily functionalities and as a whole are crucial for a normal living.

Healthy Ageing

As mentioned earlier the brain is filled with a large number of neurons, which are responsible for it working properly. What is very interesting about the neurons is that, as opposed to ordinary cells, they have a very long life-span. They are created while the human is still a fetus and can live for as long as 100 years. If they die they are extremely rarely regenerated. Therefore it is vital that they are properly maintained and repaired in case of damage.

While the human body ages, it changes. The same thing happens to the brain as well. Changes in the neurons and neurotransmitters, which can be caused by degradation of white matter, can have a negative effect on the communication between neurons. Shrinking of parts of the brain, thinning of the blood vessels or even inflammation of certain areas if an injury or disease has occurred are some of the physiological changes that appear with ageing.

Depending on the extend of these changes and on the time when they start happening they can affect each individual on different levels. Some of the most common indications of ageing are a decreased ability of learning new things, or retrieving information from memory, or even higher difficulty in performing certain tasks, which would otherwise feel easier. However, given enough time, a healthy elder will be able to perform these tasks successfully.

Interestingly, it has been observed that, in many cases the brain activates certain parts in order to perform a task that would otherwise require different parts of the brain. Even though this is not fully explained by scientists, it can be perceived as a kind of compensation mechanism for difficulties that other regions may be experiencing.

In general there are not definite factors that lead to normal or not normal ageing, but a number of factors are believed to play a role. Factors such as overall health, environment, education (and by extension intellectual stimulation of the brain) and genetics are believed to be some of the factors that could affect the brain's course throughout its life.

1.2 Alzheimer's Disease

As discussed in Section 1.1, the brain goes through morphological and structural changes while it ages. These changes cause a decline in cognitive and memory functions. For a healthy individual, this decline is not as severe so as to interfere with their daily lives. However in an ever increasing number of cases, this decline is quite severe, causing problems in the daily life of an elder person and in many cases leading to early death. This is the case of **dementia**, which is categorised as a neuro-degenerative disorder.

Alzheimer's disease (**AD**) is the most common form of dementia, accounting for $50 - 80\%$ of dementia cases. The average life expectancy of people suffering from AD is $3 - 10$ years and depends greatly on the age when the disease was diagnosed or first appeared. It is believed to be related to structural atrophy of certain regions of the brain, pathological amyloid depositions and metabolic alterations in the brain. However, scientists are still uncertain if these are the causes of the disease, or are by-products of its development.

Mild cognitive impairment (**MCI**), which in many cases is also referred to as amnestic mild cognitive impairment (due to its effect on memory), is a condition which is considered to be an early stage of AD. MCI is not characterized as a disease, it is however a condition that resembles AD, but with less intense symptoms. People that exhibit symptoms of MCI, experience problems with memory, as mentioned earlier, language and even judgement. These problems are not so grave, so as to interfere with their lives and that is why the condition cannot be characterized as AD or dementia. They are however severe enough to be noticeable by others and therefore be separated from healthy ageing symptoms.

The lines between the three stages (healthy ageing, MCI, AD) are quite blurry and there are no definite criteria for estimating in which stage each individual is, but over years of research certain ways have been developed to assess the brain's health.

1.3 Brain Studies & Machine Learning

Studying the brain and by extension its abnormalities has been a very important field in science and medicine. However, for a long time, a drawback was the inability to study it non-invasively. Even today, a definite diagnosis for AD can only be made post-mortem, during the autopsy, when a doctor is able to examine the amyloid plaques that are created in the brain and other indications of brain degeneration.

The need for an early detection and diagnosis of AD and MCI is imperative, since it gives doctors a chance to prevent further degeneration or try to decrease the symptoms through therapy. Especially early detection and diagnosis of MCI can help patients to avoid developing AD. It is estimated that patients with MCI progress to AD with a rate of $10 - 15\%$, while healthy individuals progress to any form of dementia with a rate of $1 - 2\%$.

The MRI (Appendix A) has given scientists the opportunity to look into the human body in vivo. This means that it is possible to see and examine the brain, at different stages of the disease's progression, or even examine healthy brains, in order to establish a baseline of how it should look like. With the addition of computer science and machine learning it has been possible to study the brain and its changes either due to simply ageing, or due to illness.

A lot of research has been focused on identifying the optimal features of an MRI scan, that is the best diagnostic predictors, and extracting them, in order to be able to proceed with further diagnosis ([1], [2], [3], [4], [5]). The most commonly used characteristics are the volume of gray and white matter, either throughout the entire brain or within specific regions, such as the frontal, temporal, parietal and hippocampal cortex. Cortical thickness is also quite commonly used, as well as density maps of cerebrospinal fluid (CSF).

However, it can be a very difficult and tedious task to manually extract and calculate the necessary features and more importantly prone to mistakes. Hence a number of tools have been developed, such as FreeSurfer ([6]), FSL ([7]) and SPM ([8]), that enable scientists and medical doctors to manipulate MRI scans in different ways and extract accurate features.

There has also been great progress in the attempt to manipulate those features. Using different classification methods it is possible to distinguish between a healthy and a diseased brain, both from AD and MCI. Some of the most common methods and algorithms are support vector machines (SVM) ([4], [5]), orthogonal projection to latent structures (OPLS) ([9], [10]), artificial neural networks (ANN) ([9], [11]), decision trees and other methods for classification or regression. These methods use cross-sectional MRI scans and mostly focus on classifying the scans as [healthy, MCI or AD], [healthy or demented] and even predicting the age of the individual, while producing very promising results, with support vector machines (SVM) achieving accuracy up to 93.2%.

Studying just cross-sectional MRI scans, even though successful, does not give that much information. A quite recent attempt has been made, in order to study longitudinal MRI scans, i.e. MRI scans of the same patient acquired over periods of predefined time windows (usually 6 months or 1 year). This way it is possible to not only examine a healthy/non-healthy brain (and by extension learn how to classify it), but also study how the brain changes over time either because of a disease, or normal ageing.

Some research that includes the temporal domain has been done on fMRI scans ([12], [13]) focusing primarily on the responses that certain regions exhibit. These studies use mostly regression models (linear, generalized least squares etc.), but cannot be considered to be longitudinal. In [14] the use of Hidden Markov Models (HMMs) is introduced in an attempt to detect early dementia (mild Alzheimer's disease) in elderly people. In this study, features extracted from a sequence of slices of an MRI scan are formed into a time-series which then, using HMMs, are analysed and classified. The proposed method is quite successful in detecting early dementia and in certain experiments performs with accuracies as high as 97.8%.

Even though these studies perform well, the use of longitudinal MRI scan is still absent. When it comes to actual longitudinal MRI scans, HMMs seem to be a very fitting method ([15], [16]). HMMs (Section 2.2) can successfully incorporate the time variable when examining observations (MRI scans). In [15] longitudinal MRIs and HMMs are used in order to detect the age of non-demented subjects, which is acquired with an average error of as low as 2.57 years.

1.4 Research Question

As mentioned earlier, MCI is a cognitive condition, where the individual exhibits signs of cognitive decline that are intense enough so that they are noticeable, but not as intense so that they can be characterized as AD or any other form of dementia. It is considered to be a prodromal stage of AD. Even though it is not a definite diagnosis (meaning that progression to AD is inevitable), it can serve as a red flag for the doctor and the individual, so that they can both be warned, be alert and start "treating" the individual so that a milder disease can occur later or even so that the disease can be avoided.

Like any other cognitive condition, many factors are important concerning its progression, such as the time when it first appears, the mental status of the individual etc. This plethora of factors affects the progression of the condition as well, which could either progress to AD or remain stable or even convert to a state where the individual is considered healthy (similar cognitive abilities are "classified" differently at different ages). It would therefore be very helpful for a doctor to be able to predict or know to a certain extend whether a patient diagnosed with MCI is prone to progress to a more serious condition or not.

The current thesis is an attempt to analyse and study the structural information extracted from longitudinal brain MRI scans. In particular, using the longitudinal series of MRI scans of different individuals, that are diagnosed with different conditions (Healthy, MCI, AD) and that progress to different conditions (Healthy, MCI, AD) it is attempted to study the progression of the individuals diagnosed with MCI and predict the development of their condition and whether or not it will develop in AD, using only the structural (volumetric) information of the MRI scans and no other biomarkers or clinical and cognitive measurements.

Chapter 2

Methods

The second chapter presents the background theory and the techniques and materials used for the design and completion of the thesis. The proposed system is described, along with all the information needed from the reader in order to understand how the system works and experiments are carried out.

2.1 Material

The MRI data used in this thesis were obtained by the Alzheimer's disease Neuroimaging Initiative (ADNI). ADNI was launched in 2003 and is funded by the National Institute on Ageing (NIA), the National Institute of Biomedical Imaging and Bioengineering (NIBIB), the Food and Drug Administration (FDA), several private pharmaceutical companies and non-profit organizations (Alzheimer's Association, Institute for Study of Ageing) and in collaboration with the NIH Foundation [17].

Its primary goal is to develop and implement methods for the acquisition of longitudinal data on patients with AD and MCI, as well as healthy individuals. It tries to test whether serial MRI, PET, other biological markers and clinical and neuropsychological assessments can be combined to measure the progression of MCI and early AD and also provide a generally accessible imaging and clinical data repository, which describe longitudinal changes in brain structure and metabolism, cognitive function, and biomarkers in healthy elderly, MCI, and AD patients. ADNI subjects were recruited from over 50 sites across the U.S.A. and Canada [18].

For the current thesis longitudinal data of 631 individuals were provided. Each individual has different number of follow-up scans, ranging from 1 up-to 3 follow-ups, all with a one-year time window between scans. Figure 2.1 and Table 2.1 provide a summary of the basic characteristics of the dataset. In total 1913 MRI scans were provided (1.5T sagittal 3D T1-weighted MPRAGE MRI scans). It can be seen in the Figure that the cases that are diagnosed with AD from the first scan remain only in that stage, there are no conversions to neither MCI nor healthy states. Also, these cases have up-to two follow-up scans (three MRI scans in total per sequence). Within the other two groups, the individuals have from one up-to three follow-up scans and their diagnoses progress to all possible conditions.

These MRI scans are preprocessed using the Freesurfer [19] pipeline, which is an open source suite that provides tools for the extensive and automated analysis of key features of the human brain. This includes volumetric segmentation of most macroscopically visible brain structures, segmentation of hippocampal sub-fields, inter-subject alignment based on cortical folding patterns, estimation of architectonic boundaries from in vivo data, mapping of the thickness of cortical gray matter and other functions, which would be impossible to do manually and for large number of data. As a result, a total of 55 MRI-derived regional measures, including 34 cortical thickness and 21 subcortical volumes, are extracted from each MRI scan.

Table 2.1: Summary of the Data-set

		Total	Females	Males
No. of Participants		631	264	367
Condition at First Scan	Healthy	192	92	100
	MCI	309	109	200
	AD	130	63	67
Condition at Last Scan	Healthy	189	87	102
	MCI	202	68	134
	AD	240	109	131
No. of Follow-Up Scans	1	254	107	147
	2	103	48	55
	3	274	109	165
Age (min - max)		55.2 - 89.7	55.2 - 89.7	56.5 - 88.9

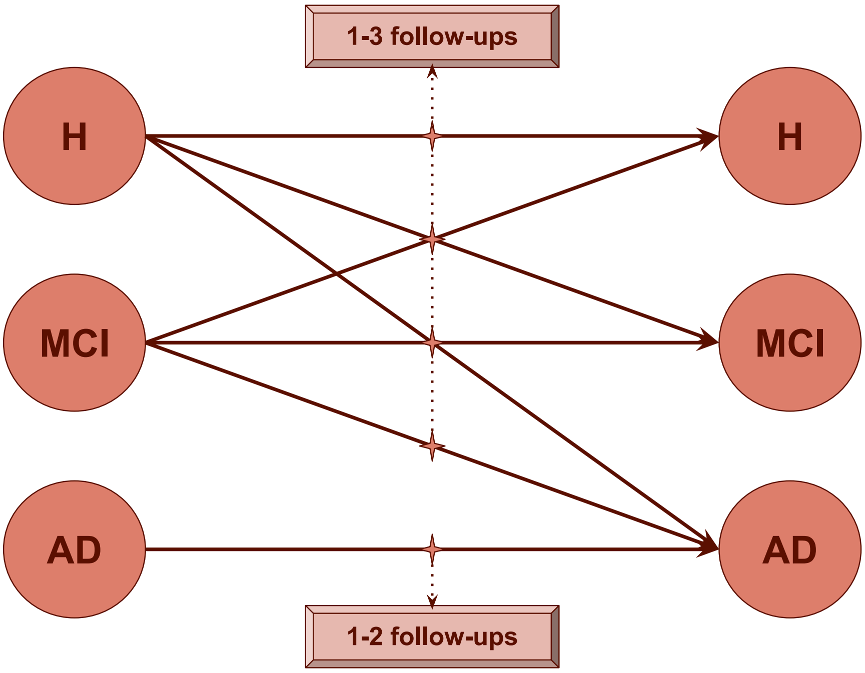


Figure 2.1: Subjects' Initial and Final Diagnoses

2.2 Hidden Markov Models

A **Hidden Markov Model** (**HMM**) is a tool that can model a series of observations that are produced by a real-world process assumed to be a Markov chain, that produces a number of hidden states (latent variables).

Markov Chain Assuming that there is a number of N possible *states*, $S = s_1, s_2, \dots, s_N$. The process starts from one of these states and can move "forward" from one state to the other, producing a sequence of different states. At every step, the process can move to a new state based on a probability distribution, that involves the current state i and all other states (including itself). This is called *transition probability* and is usually represented by an $N \times N$ matrix A, called *transition matrix*:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{i2} & \dots & a_{NN} \end{bmatrix}$$

Each element a_{ij} of this matrix gives the probability of moving towards state s_j when the current state is s_i. Matrix A is row-stochastic, meaning that each row is a unique probability distribution which needs to add up to 1. It is possible for elements to be zero, which would indicate that a certain transition is not possible (e.g. if $a_{35} = 0$, it is not possible to move to state s_5 from state s_3).

In addition to matrix A, we have vector $\pi = [\pi_1, \pi_2, \dots, \pi_N]$. It is a probability distribution for the starting state of the sequence. Using A and π it is possible to produce a sequence of states (potentially infinite), which is called a **Markov chain**. The basic concept behind the Markov chain is that at every step of the chain, the next step is *dependent only on the current state*.

Hidden Markov Models Building upon the Markov chain described in the previous paragraph it is now assumed that the state sequence that is produced by the process is not observable (it is hidden), but that it is possible to acquire a sequence of observations produced by the states at every step of the process. We therefore assume that we have a number of N possible states, $S = s_1, s_2, \dots, s_N$, a transition matrix A, an initial probabilities vector π and a number of M possible observations $O = o_1, o_2, \dots, o_M$. When the process, at step t, is at state s_i, it emits an observation based on the $N \times M$ emission matrix B:

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1M} \\ b_{21} & b_{22} & \dots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{i2} & \dots & b_{NM} \end{bmatrix}$$

Similarly to the transition matrix, the emission matrix is also row-stochastic and each row is the probability distribution of the corresponding state emitting a possible observation (e.g. element b_{ij} is probability of state s_i emitting observation o_j).

Having defined an HMM λ, there are three problems of interest that we need to solve in order for it to be applicable to real-life applications:

1. **Evaluation Problem:** Given an HMM λ and a sequence of observations $O = o_1, o_2, \dots, o_T$, what is the probability $P(O|\lambda)$ that the sequence is generated by the model λ?

2. **Decoding Problem:** Given an HMM λ and a sequence of observations $O = o_1, o_2, \dots, o_T$, what is the most likely state sequence that could have produced the observations sequence?

3. **Learning Problem:** Given an HMM λ and a sequence of observations $O = o_1, o_2, \ldots, o_T$, how can the model parameters be adjusted so that the probability $P(O|\lambda)$ is maximised?

Fortunately these problems have been solved and there are algorithms that provide the solutions:

Evaluation Problem | Forward Algorithm This algorithm introduces an auxiliary variable called forward variable, which is defined as $\alpha_t(i) = P(o_1, o_2, \ldots, o_t, s_t = i|\lambda)$. It represents the following probability:

Having observed the observation sequence until the t-th observation, what is the probability that the t-th state is $s_t = i$?

We can easily calculate $\alpha_1(i)$ for $1 \leq i \leq N$ as follows:

$$\alpha_1(i) = \pi_i b_{io_1},$$

which, for all possible i's will give us the probabilities of having observed the first observation originating from all possible states. The algorithm then uses the following recursive formula:

$$\alpha_{t+1}(j) = b_{jo_{t+1}} \sum_{i=1}^{N} \alpha_t(i) a_{ij}, 1 \leq j \leq N, 1 \leq t \leq T-1$$

Using this formula it is possible to calculate all possible $\alpha_T(i), 1 \leq i \leq N$. Finally, the probability that we are looking for can be calculated:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

This method has complexity $O(N^2T)$. Similarly we can define another auxiliary variable $\beta_t(i)$, called the backward variable:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T|s_t = i, \lambda)$$

This new variable gives the probability that, given an HMM λ and assuming that the tth state is $s_t = i$, the observation sequence from step $t+1$ to the end is $O = o_{t+1}, o_{t+2}, \ldots, o_T$.

For β_t the last time-step $\beta_T(i)$ is set first, for $1 \leq i \leq N$:

$$\beta_T(i) = 1$$

This is an arbitrary definition for β_T being 1 for all i. Then, $\beta_t(i)$ is recursively calculated:

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_{jo_{t+1}}, 1 \leq i \leq N, 1 \leq t \leq T$$

After calculating both the forward and backward variables we can alternatively calculate $P(O|\lambda)$:

$$P(O|\lambda) = \sum_{i=1}^{N} P(O, s_t = i|\lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i)$$

Decoding Problem | Viterbi Algorithm For the decoding problem there are two different alternative approaches to what the solution could be. One possibility is to try to find the states s_t that are individually most likely. This solution produces a sequence with the *highest number of correct states*. It requires a new variable,

$$\gamma_t(i) = P(s_t = i|O, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)},$$

which gives the probability that the t-th state is $s_t = i$, given an HMM λ and an observation sequence $O = o_1, o_2, \ldots, o_T$. Using $\gamma_t(i)$ we can get all optimal states s_t by simply finding:

$$s_t = \arg \max_{1 \le i \le N} (\gamma_t(i)), 1 \le t \le T$$

This method, even though it produces a sequence with the highest number of the most likely states, has an important drawback. That is the fact that it completely disregards the **possible** transitions. This could result in a state sequence that includes transitions that are not even valid (it could be that $a_{ij} = 0$, but it is possible for the sequence to contain the transition $s_t = i, s_{t+1} = j$).

Therefore the most common approach to this problem is the Viterbi algorithm, which produces the most likely state sequence as a whole. We introduce yet another variable:

$$\delta_t(i) = \max_{s_1 s_2 \ldots s_{t-1}} P(s_1, s_2, \ldots, s_{t-1}, s_t = i, o_1, o_2, \ldots, o_{t-1}|\lambda)$$

This variable gives the maximum probability that partial observation and state sequences up to step t can have, given that we currently are at state $s_t = i$. Again, to initialize, for $1 \le i \le N$:

$$\delta_1(i) = \pi_i b_{io_1}$$

And the recursive formula is:

$$\delta_{t+1}(j) = [\max_{1 \le i \le N} \delta_t(i) a_{ij}] b_{jo_{t+1}}, 1 \le i \le N, 1 \le t \le T-1$$

So, basically, after calculating $\delta_T(j), 1 \le j \le N$, we can find state j_T^* which maximizes $\delta_T(j_T^*)$:

$$j_T^* = \arg \max_{1 \le j \le N} \delta_T(j)$$

Starting from this state, it is possible to back-track the most likely state sequence using each j_t^* as a pointer for the optimal state of the previous step.

Learning Problem | Baum-Welch Algorithm The learning problem is in a way a problem of “training” an HMM, given a set of observation sequences, which we need to model, having little or no prior knowledge about how the model works (matrices A and B and vector π).

The Baum-Welch algorithm uses the Expectation-Maximization (EM) method, in order to estimate a set of parameters for an HMM $\lambda = (A, B, \pi)$, which maximize $P(O|\lambda)$ at least locally. The basic concept of every EM algorithm is that it calculates a set of parameters, which are then re-estimated in order to maximize a certain quantity (in our case $P(O|\lambda)$), that is expressed via the auxiliary function:

$$Q(\lambda, \bar{\lambda}) = \sum_{s} P(s|O, \lambda) \log(P(O, s, \bar{\lambda}))$$

This algorithm uses the variables $\alpha_t(i)$, $\beta_t(i)$ and $\gamma_t(i)$ introduced in the two previous problems. It also introduces another variable, that similarly to $\gamma_t(i)$ can be expressed using $\alpha_t(i)$ and $\beta_t(i)$:

$$\xi_t(i,j) = P(s_t = i, s_{t+1} = j|O,\lambda) = \frac{P(s_t = i, s_{t+1} = j, O|\lambda)}{P(O|\lambda)} = \frac{\alpha_t(i)a_{ij}\beta_{t+1}(j)b_{jo_{t+1}}}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}\beta_{t+1}(j)b_{jo_{t+1}}}$$

This last variable gives the probability of being at state $s_t = i$ at the tth step and moving to state $s_{t+1} = j$ at the $(t+1)$th step. Having defined $\xi_t(i,j)$, we can now express variable $\gamma_t(i)$ defined in the decoding problem as:

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j), 1 \leq i \leq N, 1 \leq t \leq T$$

At this point it is simple to describe the learning process. We initialize the model's parameters, either randomly, or based on any prior knowledge we may have about the model $\lambda = (A, B, \pi)$. We then calculate the α's, β's, γ's and xi's based on the formulas given earlier. We re-calculate the model's parameters in order to maximize the quantity $P(O|\lambda)$, using the following formulas:

$$\pi_i' = \gamma_1(i), 1 \leq i \leq N$$

which is the expected number of times the model has started from state $s_1 = i$ at the first step,

$$a_{ij}' = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}, 1 \leq i \leq N, 1 \leq j \leq N$$

which is the expected number of transitions from state $s = i$ to state $s = j$ relative to the transitions from state $s = i$ and

$$b_{jo_k}' = \frac{\sum_{t=1}^{T}1_{o_t=o_k}\gamma_i(t)}{\sum_{t=1}^{T}\gamma_i(t)}, \text{ where } 1_{o_t=o_k} = \begin{cases}1, & \text{if } o_t = o_k\\ 0, & \text{otherwise}\end{cases}$$

which is the expected number of observations being o_k while being in state $s = i$ relative to the expected number of times being in state $s = i$.

While the Baum-Welch algorithm cannot guarantee finding a global maximum, it guarantees that it will definitely converge to at least a local maximum.

Different Types of HMMs

Observations As described earlier, matrix B gives the probability that a certain observation is emitted by a certain state. Basically every row of the matrix represents the probability distribution of the observations for the corresponding state. Naturally since an HMM models real-life processes, it is possible that the observations can be either discrete or continuous. In either case the probability distributions have the fitting type.

Infinite Duration HMM In theory all HMMs are infinite duration, unless specifically designed otherwise, meaning that the process can move from state to state forever, and practically the length of the sequences (either state or observation) is only restricted by the real-life constraints of the process (it usually stops at some point).

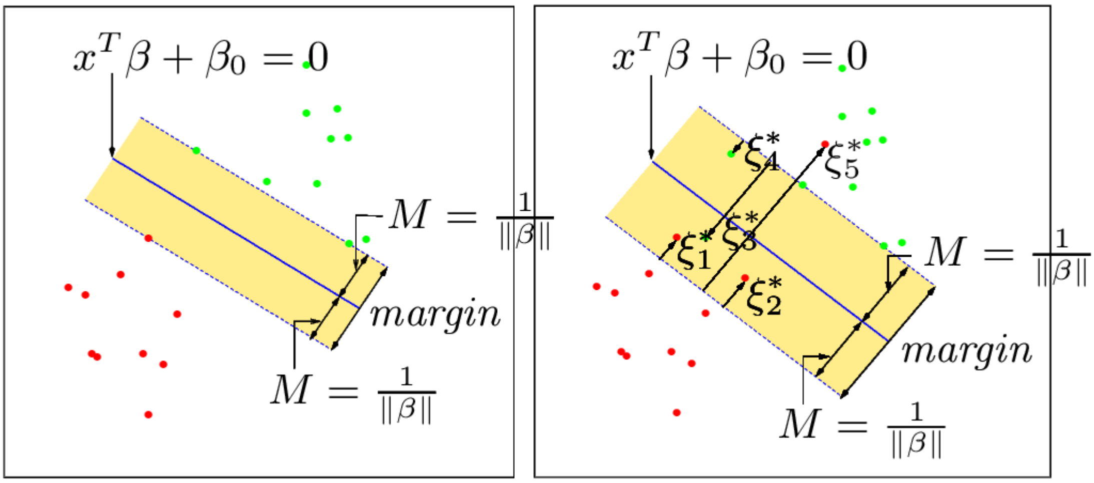


Figure 2.2: Support Vector Classifiers [20]

Finite Duration HMM For this type of HMM, a special state is introduced, which is called *exit* state. Once the Markov chain reaches that state there is no observation emission and the process stops. In this case the transition matrix A becomes $N \times (N+1)$. This practically means that there is a transition probability from any state to the $(N+1)$-th *exit* state, but since there is no $(N+1)$-th row, it is not possible to transition from the exit state to any other state.

Left-Right HMM A Left-Right HMM operates under the restriction that as we are “stepping” forward, the states’ indices can only increase or stay the same, but can never decrease. This type is able to model applications or signals that change irreversibly over time (i.e. speech signals). The basic property of the Left-Right HMMs can be expressed by:

$$a_{ij} = 0, j \leq i$$

Null Transitions & Tied States Certain applications, if the length of the observation sequence is not long or the amount of training data is small, may require a more complex model than usual. In these cases the model allows *null transitions*, which are basically state transitions that emit no observations, enabling the model to produce state sequences that are longer than the corresponding observation sequences. Additionally it is also possible to tie parameters together. This sets up an equivalence relation between HMM parameters for different states. This facilitates the training process and initial parameter estimation.

2.3 Support Vector Machines

A **Support Vector Machine** (**SVM**) is a *supervised learning* algorithm, which is used for binary data classification and also, less frequently and with certain extensions, for regression tasks and also multi-labelled data classification. The SVM has a very robust performance when it comes to data noise and sparsity, which makes it very effective and very popular in a great variety of applications. When it comes to binary classification the main goal of an SVM is, given a labelled data-set, to find the optimal separating hyperplane, such, that it correctly classifies the training data and generalizes in the best way possible with unseen data.

Let D be a training data-set, that contains N pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$, where $x_i \in \mathrm{I\!R}^p$ and $y_i \in \{-1, 1\}$.[1] The data-set is assumed to be perfectly separable. Given this training set, the SVM will define a hyperplane $x : f(x) = x^T\beta + \beta_0$, where β is a unit vector. $f(x)$ gives the signed distance of a point x to the hyperplane. If vector β is adjusted correctly, then the hyperplane can be separating the data-points that belong to the two classes, with $y_i f(x_i) > 0 \forall i$, while also optimizing the hyperplane so that the distance of the data-points that lie closest to it (*margin*) is maximum. The left image of Figure 2.2 shows such a separating hyperplane, for the case where the training set consists of 2D points. It can be shown that the width of the margin is equal to $2M = \frac{2}{\|\beta\|}$. This leads to the following inequality:

$$y_i(x_i^T\beta + \beta_0) \geq M, i = 1, \ldots, N, \tag{2.1}$$

meaning that all points are lying at a minimum distance M from the hyperplane, depending on their class. The data-points, that lie on the boundaries of the classes (distance $\frac{1}{\|\beta\|}$ on either side of the hyperplane) and, therefore, are the ones that define the margin and determine the position of the hyperplane, are called **support vectors**.

In the case, where the two classes are not perfectly separable by a hyperplane, an approach to overcome this issue, and still be able to calculate the optimal hyperplane with the maximum margin is to introduce the so-called *slack variables* $\xi = (\xi_1, \xi_2, \ldots, \xi_N)$ and modify the inequality previously introduced (Eq. 2.1) as follows:

$$y_i(x_i^T\beta + \beta_0) \geq M(1 - \xi_i), i = 1, \ldots, N, \xi_i \geq 0, \sum_{i=1}^{N} \xi_i \leq C.$$

These slack variables, indicate the extend to which each data-point is “allowed” to violate the class’ borders. This could mean that certain data-points may lie on the right side of the hyperplane, but within the margin (correctly classified), or they may lie on the wrong side of the hyperplane (misclassified). Basically each variable ξ_i is the proportion to which, the distance $f(x)$ is on the wrong side of the margin. ξ_i’s values can be divided into the following areas:

$$\xi_i \begin{cases} = 0, & \text{if } x_i \text{ lies on the right side of the margin} \\ \in (0, 1), & \text{if } x_i \text{ lies on the wrong side of the margin (correct classification)} \\ = 1, & \text{if } x_i \text{ lies on the hyperplane} \\ > 1, & \text{if } x_i \text{ is missclassified} \end{cases}$$

With the use of these variables, the classifier is still able to define a hyperplane with maximum margin, at the expense of having a number of points either misclassified, or lying within the margin. By applying the constraint $\sum \xi_i \leq C$ we define the “strictness” of the training procedure, basically restrict the number of misclassifications.

This whole alteration and use of the slack variables is displayed in the right image of Figure 2.2.

In the case that the given data-set is not linearly separable, even with the use of the slack variables, it is sometimes possible to define a linear separating hyperplane, by mapping the data to a higher dimension, or by expressing the data using a different co-ordinate system. In general, a linear solution can be found by defining a different feature space, in order to map the data and render it linearly separable.

Figure 2.5 shows the case where the data-set is not linearly separable. When the data is expressed using the polar co-ordinate system, it becomes perfectly separable, a very simple and “easy” case for an SVM classifier. Figure 2.4, on the other hand, shows the same data-set, where the approach for the classification is different. In this case the data has been mapped to a higher dimension ($2D \rightarrow 3D$), where again it is perfectly separable by a hyperplane of a higher dimension.

[1] Notation in this section follows [20]

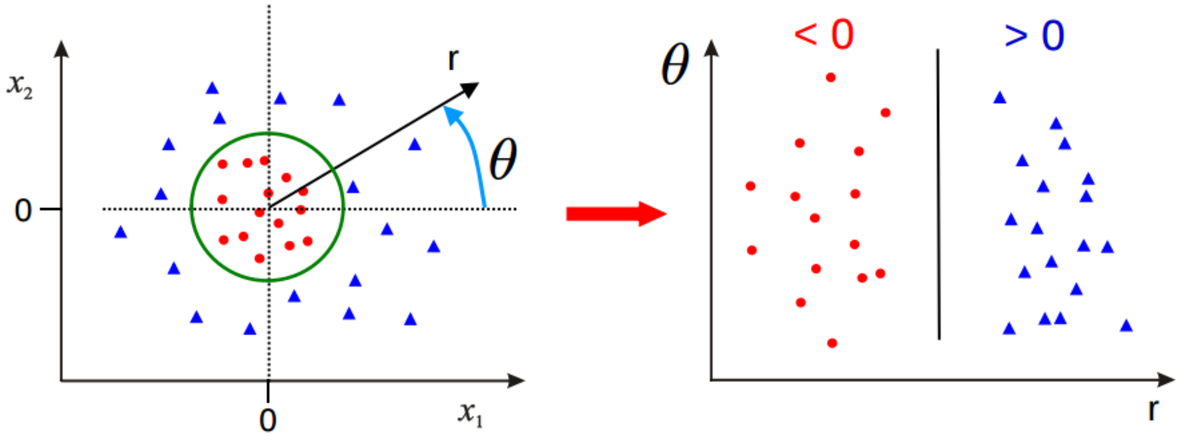


Figure 2.3: Using Polar Coordinates to achieve linear separation [21]

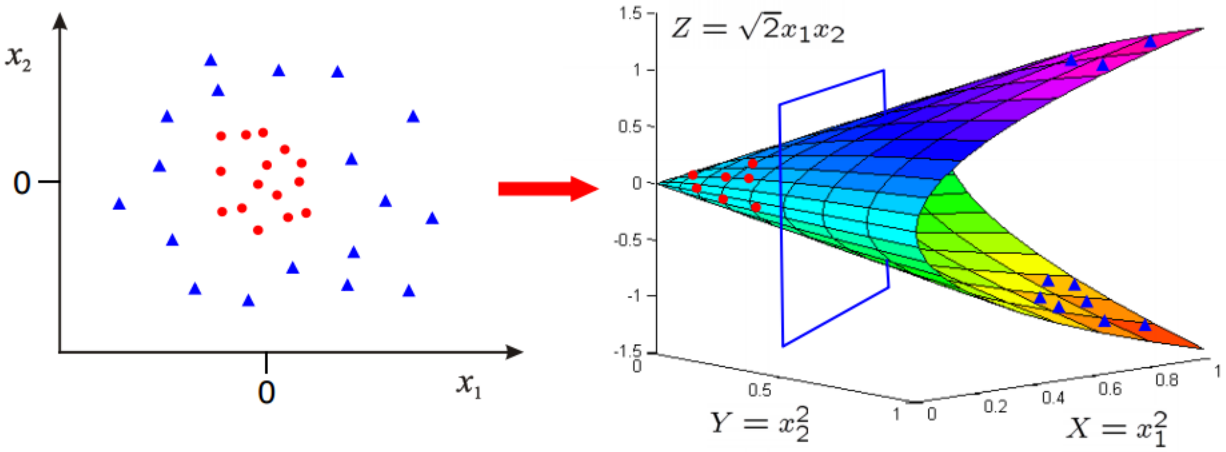


Figure 2.4: Mapping data from 2D to 3D to achieve linear separation [21]

In both cases the data-points have been replaced by their representation in a new feature space, using the *feature map* $\Phi(x)$. This way the initial hyperplane function is expressed:

$$f(x) = \Phi(x)^T \beta + \beta_0.$$

Using this feature map, during training and during the final classification of the data, in calculations, only the inner product $\Phi(x_j)^T \Phi(x_i)$ occurs. The need for knowledge of the mapping $\Phi(x)$ is eliminated and the need for knowledge of the inner product rises. Hence, the algorithm uses a **kernel function**, which is defined as follows:

$$k(x_j, x_i) = \Phi(x_j)^T \Phi(x_i).$$

The use of the kernel method, simplifies the calculations occurring during training and lightens the computational load. More importantly it converts the computational complexity of the algorithm to a function of the size of the data-set (N) from a function of the dimensionality of the data-set. This means that the algorithm can handle large feature spaces that potentially carry more information.

Some of the most popular kernels are:

1. **Linear**: $k(x, x') = x^T x'$
2. **Polynomial**: $k(x, x') = (1 + x^T x')^d, d > 0$
3. **Radial Basis Function**: $k(x, x') = \exp(-\gamma \|x - x'\|^2))$

2.4 Proposed Methods

As discussed in Section 1.4 the aim of the thesis is to study the course of the MCI condition throughout time by examining brain structure (volumetric features extracted from brain MRI scans using Freesurfer (Sec. 2.1)). It has been noted that HMMs work primarily with this concept, i.e. it studies sequences of observations that are presumably produced by a Markov chain throughout time. Therefore the HMM is the basis of our approach.

There are three different methods that have been developed, each one built upon the previous. In the following sections, these methods are going to be discussed and explained in the following sections.

Notations and Data Separation At this point, it is fitting to present certain basic notations that are going to be used in the following sections, as well as the way that the data is going to be divided and used. As described in Section 2.1 the data-set consists of a number of MRI scans that belong to different subjects. Each subject has a series of scans that consists of the **cross-sectional** scan, which is the initial scan, and a number of **follow-ups** (one, two or three).

There are two ways to divide the data-set. One way is based on the diagnosis of the cross-sectional scan, which is referred to as **subject-group** and can be **healthy (H)**, **mild cognitive impairment (MCI)** or **Alzheimer's disease (AD)**. The other way is to group the subject based on their final follow-up. This grouping is referred to as **subject-end-group** and produces the same categories as the subject-group (H, MCI & AD).

Even though this type of separation of the data is derived naturally from the clinical conditions of the subject, within the scope of this thesis it is more practical to divide it differently. The subject-grouping and subject-end-grouping still remain, however the labels that they get assigned change. For the subject-group, the H and AD groups are merged, so the subject-group can now be either **non-MCI** or **MCI**. This is more fitting, because the aim is to study the progression of the MCI subject-group, since it constitutes a high-risk group. Following the same concept, we wish to detect whether the MCI will deteriorate to AD or not. Therefore the subject-end-group can be either **AD** or **non-AD**.

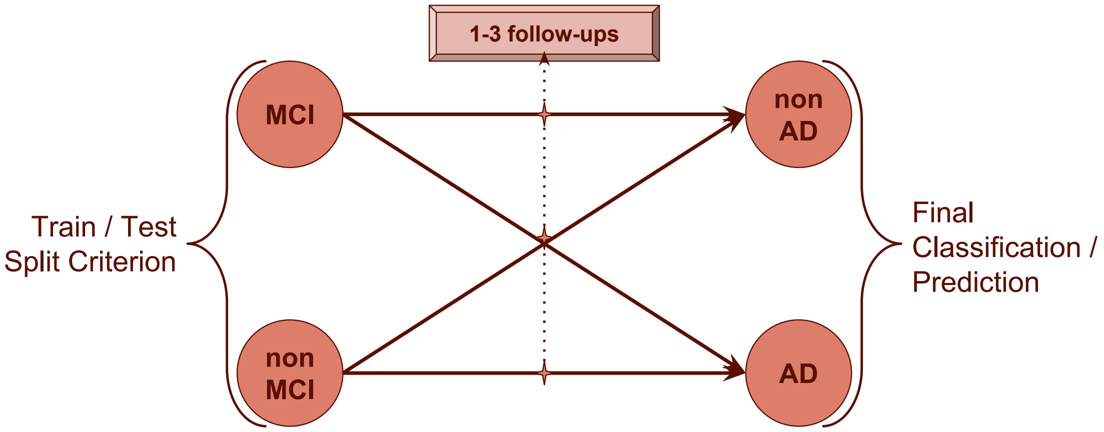

1-3 follow-ups
MCI
non MCI
non AD
AD
Train / Test Split Criterion
Final Classification / Prediction

Figure 2.5: Training & Testing Separation of the Data-set

2.4.1 HMM Classification

The first approach that we tried uses only HMMs. As described in Section 2.2 (**Learning Problem**) an HMM can be trained in such a way, so that the probability $P(O|\lambda)$ is maximized, where λ is the HMM model and $O = [o_1, o_2, \ldots, o_T]$ is a sequence of observations. In the case of the current data, one observation O sequence is one longitudinal MRI scan sequence of a subject, where each one of the observations o_t are vectors of size $[1 \times 55]$ representing the volumetric data extracted from each scan, with $T \in [2, 4]$.

We initially produce the observation sequences of all the subjects. Then the observations of the non-MCI subject-group are used in order to train two HMMs, λ_{AD} and λ_{non-AD}, using for each HMM only the observations with subject-end-groups AD and non-AD correspondingly. After the two HMMs have been trained, we then use the MCI subject-group to test it. With the use of the forward algorithm (Section 2.2, **Evaluation Problem**), two probabilities are produced for each observation sequence, $P_{\text{AD}}(O_i|\lambda_{\text{AD}})$ and $P_{\text{non-AD}}(O_i|\lambda_{\text{non-AD}})$, indicating the probability that each sequence could be produced by the corresponding HMM. Based on these probabilities, the prediction for the sequence is decided:

$$y_i = \begin{cases} \text{AD}, & \text{if } P_{\text{AD}}(O_i|\lambda_{\text{AD}}) \geq P_{\text{non-AD}}(O_i|\lambda_{\text{non-AD}}) \\ \text{non-AD}, & \text{if } P_{\text{AD}}(O_i|\lambda_{\text{AD}}) < P_{\text{non-AD}}(O_i|\lambda_{\text{non-AD}}) \end{cases}$$

This method aims to explore how well an HMM can extract and model information about the temporal structural changes of a brain when it is on the path towards AD or when it ages in a healthy manner. Then, using this information, we aim to examine how similar these changes are to the changes of a brain diagnosed with MCI.

A descriptive diagram of this method is shown in Figure 2.6.

2.4.2 HMM Modelling - SVM Classification

Building upon the logic of the previous section, we want to study whether an HMM is able to extract and model information that can be exploited by a different system, in order to be able to predict the final

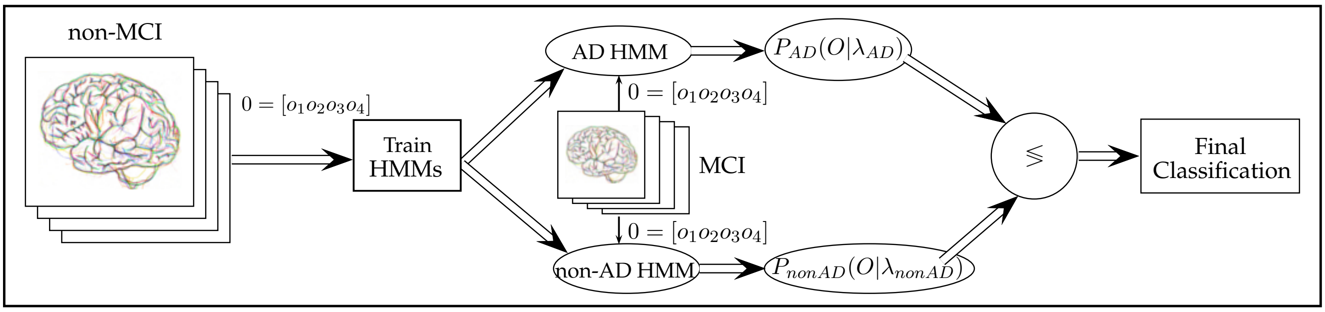


Figure 2.6: Work-flow Diagram for the HMM classification Method

diagnosis of each subject, in this case an SVM classifier.

The method prepares the data as before, producing observation sequences $O_i = [o_1, o_2, \ldots, o_T]$ for all subjects. An HMM is then trained using the non-MCI subject-group. The difference from the previous method is that now, only one HMM is trained, regardless of the subject-end-group of each observation. This is done that way, because we aim at a more generic feature extraction from the observations and want to explore the HMM's ability to model and define that information "on its own".

After the HMM λ is trained, it is used to produce state sequences for all the observation sequences (subject-groups non-MCI and MCI) (Section 2.2, **Decoding Problem, Viterbi Algorithm**). These state sequences are used as features for the next phase of the method. The feature sequences of the non-MCI subject-group are used in order to train an SVM classifier. The SVM is trained to binary classify the data into AD and non-AD, according to each sequence's subject-end-group. After the training, the SVM is tested on how well it is able to classify the MCI subject-group sequences into the two classes (AD, non-AD).

A descriptive diagram of this method is shown in Figure 2.7.

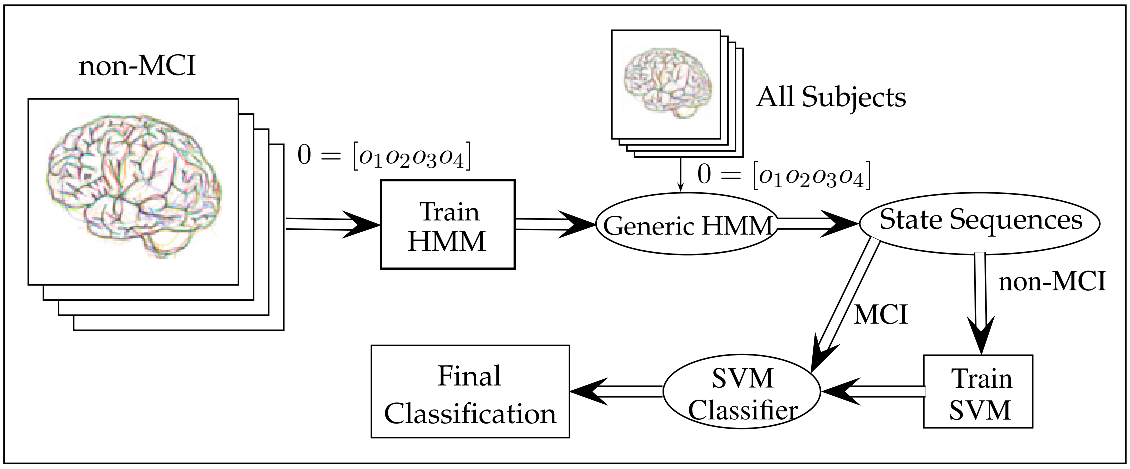
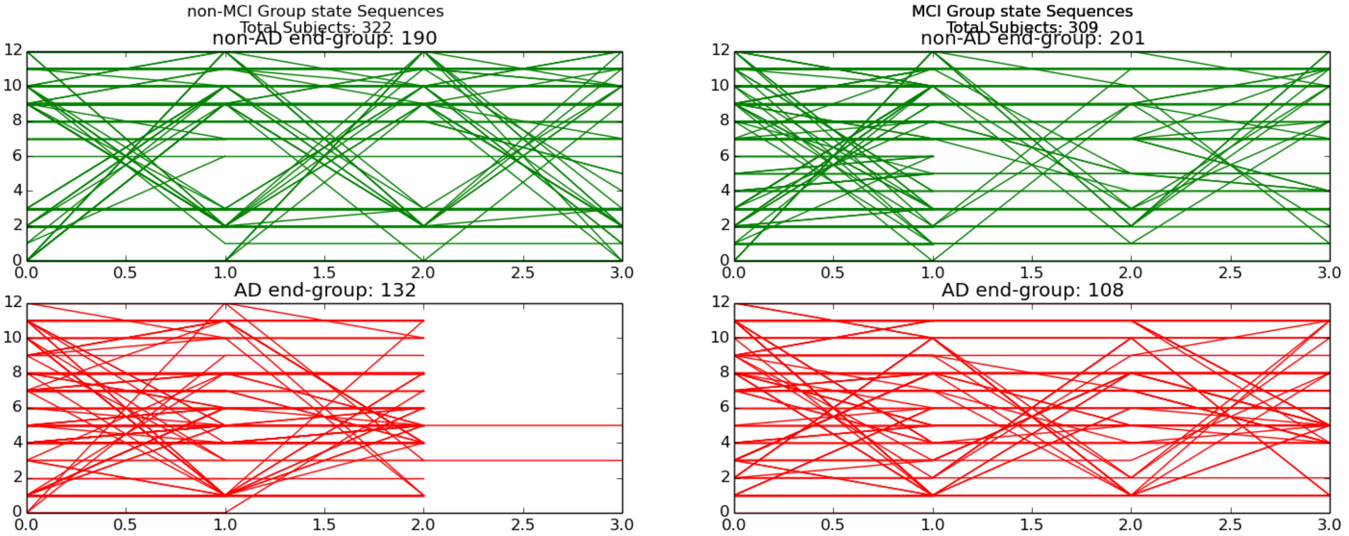


Figure 2.7: Work-flow Diagram for the HMM modelling with SVM classification Method

2.4.3 HMM Modelling - SVM Classification II

Due to the nature of the data (more will be discussed in Section 3.4), the state sequences produced by the HMM are quite unstable and display high variance, especially while the number of states increases. Additionally, the features that are produced (the unaltered state sequences), inevitably have varying lengths, according to the length of the observation sequence that produced it. This causes instability to



(a) State Sequences of non-MCI Subject-Group

(b) State Sequences of MCI Subject-Group

Figure 2.8: State Sequences produced by a 13-state HMM

the classification and interferes with the final prediction.

After producing state sequences for different numbers of states, it has been noticed that the state sequences of the non-AD subject-end-group follow a steadier path than those of the AD subject-end-group, both for the non-MCI and the MCI data (Figures 2.8a & 2.8b). There are fewer transitions occurring for the non-AD subjects. At this point, we attempt to exploit this characteristic and also attempt to remove the length variance from the produced features. We, therefore, produce the **transition frequency maps**, shown in Figure 2.9b. These maps are matrices of size $[noOfHMMStates \times noOfHMMStates]$, where each element a_{ij} is a counter of the transitions from state i to state j (Figure 2.9b, however, shows the transition maps produced by adding all the individual sequences, producing maps with seemingly more transitions than expected).

These matrices are vectorized and then used as the feature vectors of the subjects. It can be deduced by the nature of the initial matrices, that the vectors are very sparse, including mostly zeros except for a few non-zero elements that can take values $a_{ij} \in [1, 2, 3]$. Hence, the positions of the non-zero elements are of greater significance than their actual values.

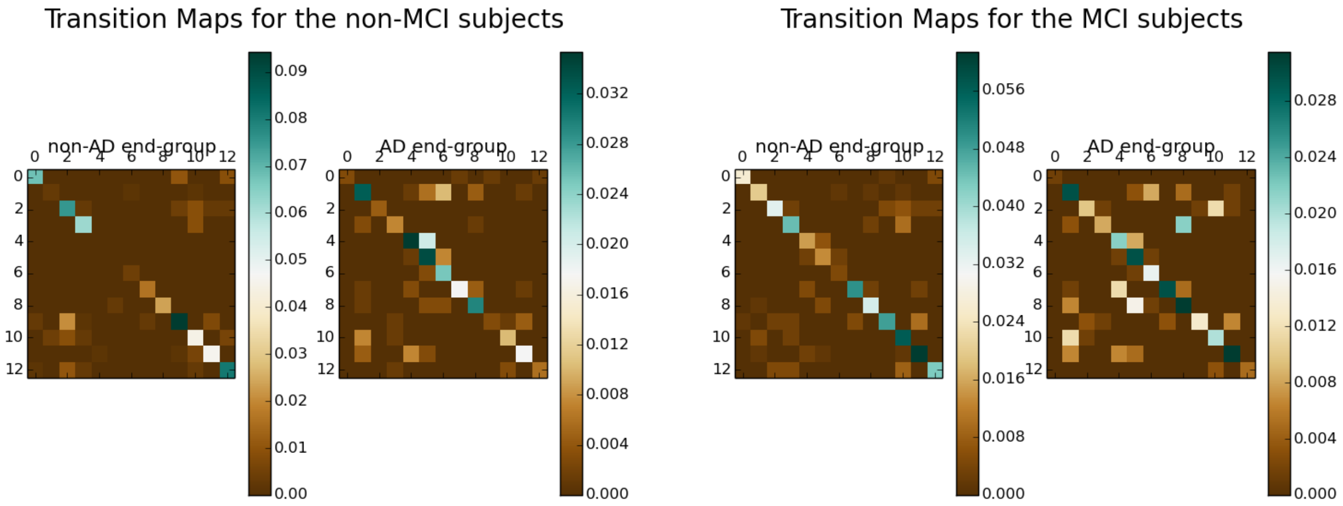
After the feature vectors have been produced, the method proceeds in a similar manner with the previous Section. An SVM classifier is trained with the use of the non-MCI subject-group's features and then tested on the MCI subject-group on its ability to classify the data into the classes AD and non-AD.

A descriptive diagram of this method is shown in Figure 2.10.

2.4.4 Practical Information

All three methods that have been described in the previous Sections have focused on excluding the MCI subject-group from the entire training phase and only use it to test the system that has been designed. This is designed that way, because we want to explore the informative "strength" of the longitudinal MRIs in providing patterns of the structure of the brain while it ages towards AD or normally and how well this information applies to the MCI data.

However, on a more practical approach, when building a prediction system it's best to use as much available data as possible so that the system can include that information when making a prediction



(a) Transition Frequency Maps of non-MCI Subject-Group

(b) Transition Frequency Maps of MCI Subject-Group

Figure 2.9: Transition Frequency Maps Produced by a 13-State HMM

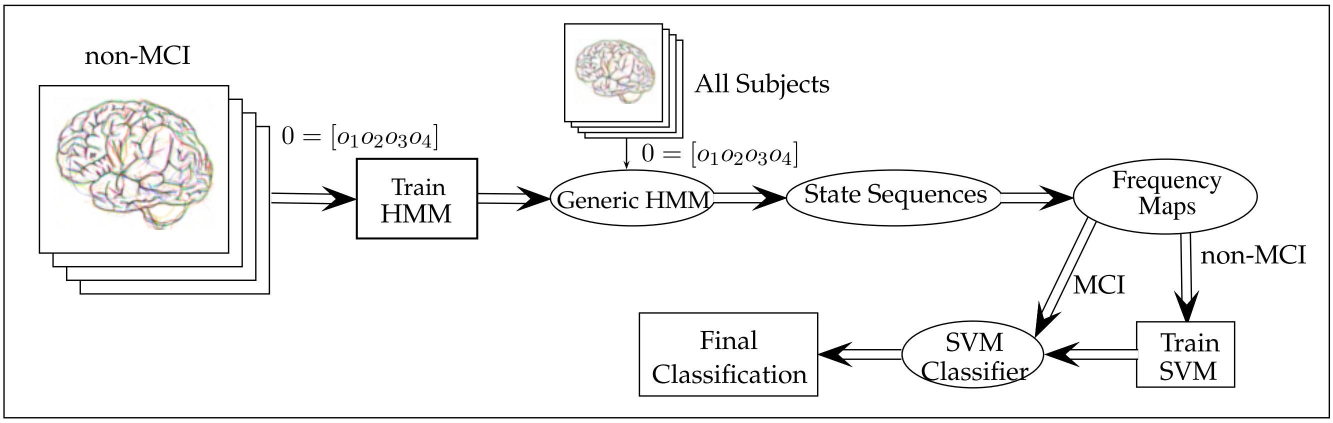


Figure 2.10: Work-flow Diagram for the HMM modelling with SVM classification Using Transition Maps Method

decision. Therefore, during the experiments we have also included altered versions of these methods, where we include part of the MCI data in the training process, by means of cross-validation. This will be described in more detail in Section 3.2.

Chapter 3

Results and Discussion

In the final chapter, the results and the evaluation of the experiments are presented. The different attempts are compared, contrasted and commented upon. At the end of the work the reader can find a discussion regarding the work of the thesis, its successful aspects, as well as the ones that can be improved.

3.1 Evaluation Metrics

For the evaluation of the experiments a certain group of metrics has been used. These are going to be mentioned in this Section.

True Positives (TP) The number of subjects that have correctly been classified as AD.

True Negatives (TN) The number of subjects that have correctly been classified as non-AD.

False Positives (FP) The number of subjects that have incorrectly been classified as AD, but are actually non-AD.

False Negatives (FN) The number of subjects that have incorrectly been classified as non-AD, but are actually AD.

Sensitivity (true positive rate - TPR) It is the proportion of the positive (AD) samples that have been correctly classified:

$$TPR = \frac{TP}{TP + FN}$$

Specificity (true negative rate - TNR) It is the proportion of the negative (non-AD) samples that have been correctly classified:

$$TNR = \frac{TN}{TN + FP}$$

Precision It is the proportion of the correctly classified samples as positive among the total number of positively classified samples.

Confusion Matrix It is a matrix that summarizes all previous metrics. The rows of the matrix represent the actual labels of the data, while the columns represent the predicted label by the classifier. Each element is a count of the data-points that fall into each category, elements of the diagonal represent the “true rates” (either positive or negative), while all other elements represent the “false rates”. Confusion matrices can be helpful even in the cases of multilabel classification. Naturally, it is expected that the matrix has the highest counts along the diagonal and lower counts or even zeros scattered around it.

F1 score It is the harmonic mean of precision and sensitivity:

$$F1 = \frac{2TP}{2TP + FP + FN}$$

Receiver Operating Characteristic (ROC curve) The ROC curve is a graphical way of showcasing the general performance of a model, while a certain parameter is varied. It is the plot of the true positive rate - TPR (Sensitivity) against the false positive rate - FPR ($1 - specificity$). The ROC curve characterises a better model as it gets closer to the upper-left corner (point $(0, 1)$) of the ROC space indicating the model exhibits high TPR and low FPR. The diagonal in the ROC space ($TPR = FPR$ for all values of the varying parameter) is characterising a random classifier and any curve below the diagonal is characterising a bad classifier (has more misclassifications than correct classifications). In general the closer the curve gets to the upper-left corner the better the model.

When building a model (or a number of models that need to be compared), other than the visual comparison that can be performed on the models’ curves (the same parameter should be varying for a more valid comparison), a common practice in order to quantify each model’s performance is to calculate the *area under the curve* (*AUC*) and then compare the numbers.

The ROC space can also be used even in the case where it is not possible or desired to test the performance while varying different parameters. In that case the performance is visualised with one point in the ROC space ((TPS, FPR)). The same “rules” apply here as before, in terms of better/worse performance. The closer the point is to the upper-left corner, the better the performance and of course it is not desired that the point lies on the diagonal line or lower than it. In case different models need to be compared, a common practice is to calculate the Euclidean distance of each model’s point from $(0, 1)$. The shorter the distance, the better the model. This practice is used also in the case of the ROC curves when one needs to determine the point of the curve that lies closer to $(0, 1)$, which would give the value of the varying parameter that produces the best results.

In Figure 3.1 we can see the ROC space and some of the possible model evaluations that we can get. If the model’s performance lies either within the red area or upon the diagonal line, it means that the model i not doing very well. On the other hand we wish that the model lies within the green area. The closer to $(0, 1)$ it lies, the better its performance, however we don’t really wish for it to be exactly on the corner. This may indicate a perfect classification but it could also be an indication of the model overfitting to the data.

3.2 Experimental Setup

The experiments are designed and executed using Python (*version* 2.7.6). In order to build the HMM models, the toolkit *hmmlearn* (*version* 0.2.0) was used. This toolkit is open-source and offers a number of algorithms and models regarding HMM-learning and usage. As it used to be part of the *scikit-learn* package (scikit-learn version 0.16 and earlier) it follows its API closely, but has been adapted to sequence data. Finally, scikit-learn has been used for the training and testing of the SVM classifier and other machine learning implementations (cross validation, evaluation metrics etc.).

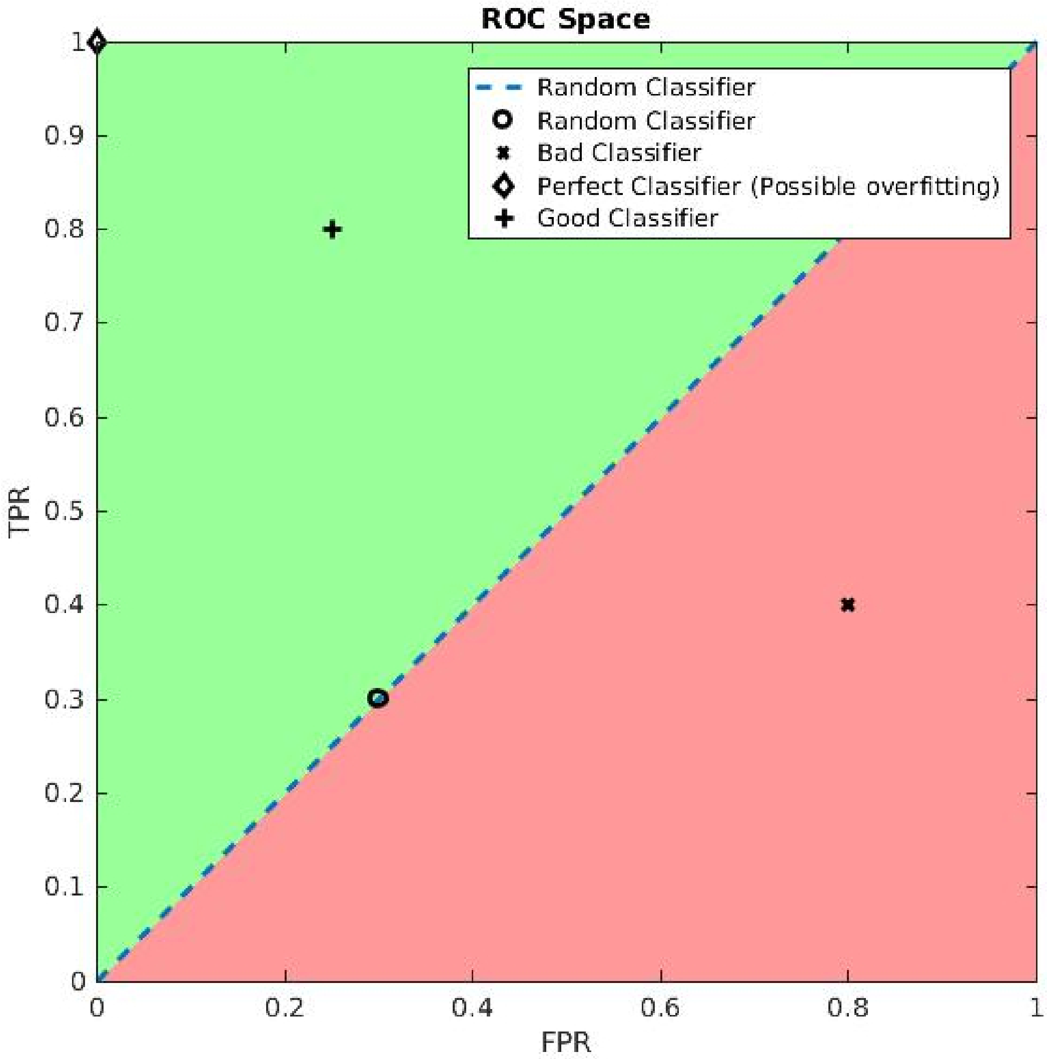


Figure 3.1: ROC Space Example

The HMMs are initiated as fully connected. This means that before any training occurs the transitions matrix contains no zero-elements and transitions from all states to all states are possible. The emissions probabilities are constructed with Gaussian emission distributions (other choices are Gaussian Mixture Model and multinomial/discrete emissions) with a "spherical" covariance (each state uses a single covariance value that applies to all the features). The number of the HMM's states varies depending on the implementation and test-run.

The SVM classifier uses a polynomial kernel of degree 3. The penalty parameter C is set to 63.26 (since the data is sparse and few, we want the classifier to be "strict" with misclassifications). Finally the kernel coefficient gamma is set to 0.001 and the independent term of the polynomial function is 3.

Cross-Validation As mentioned in paragraph 2.4.4 the methods that are developed are focused on excluding the MCI subject-group from the training process of both the SVM classifier and the HMM. In practice, during the experiments there were tests that also ran while including the MCI, in an attempt to build a better system and to test if that would indeed improve the performance of the system.

Cross validation is a method developed in order to evaluate the performance of different models. One issue that rises when training and testing a model is that during training the model learns how to perform well on the training data (or at least as well as possible), so when evaluating on its performance there, the results cannot be trusted. It therefore needs to show how well it performs on unseen data and that performance is the one that actually matters. k-fold cross validation is a common practice to overcome this issue. What happens is that, the data is divided into k subsets, where the $k-1$ are combined and used as training set and the k-th is used as unseen testing set. The process of training and testing is repeated k times and each one of the subsets is used as testing set once and as part of the training set $k-1$ times. At the end of the cross validation the evaluation metrics or errors that have been produced by each run are averaged to a final performance measure.

During the experiments we used a kind of inverted cross validation. In the beginning the data is already divided into training and testing sets (non-MCI and MCI subject-groups correspondingly). The MCI subject-group gets divided into 3 folds. At each one of the 3 iterations (k is 3 in this case) one of the folds gets merged with the non-MCI data and becomes part of the training set, while the other two are used as testing set.

Moreover, for the two methods that involve the training of an SVM classifier, cross validation has also been performed in the selection of the SVM model. In this case, the training data (either just the non-MCI or the combination of non-MCI and $\frac{1}{3}$ of MCI data) is divided into k folds ($5, 7$ or 10). k SVM classifiers are trained and tested and k F1-scores are calculated. The SVM classifier with the highest F1-score is the one that gets used by the model and classifies the testing set (MCI data) for the final evaluation of the method.

Experiments In practice, different approaches to the methods have been attempted. All three methods are tested exactly as described in Section 2.4 (no MCI data used in the training process), as well as by means of cross-validation, including a portion of the MCI data in the training process. Cross-validation on SVM was performed in all cases where SVM was actually used. Additionally, because of the nature of the data, the HMMs produce not very stable results (more will be discussed in Section 3.4). For this reason all the different experiments have been performed ten times and their results averaged, in order for the results to be smoother and any outliers to be eliminated. This a common practice when performing evaluation of probabilistic models, so that the results are smoother and more indicative of the actual performance of the models.

3.3 Results

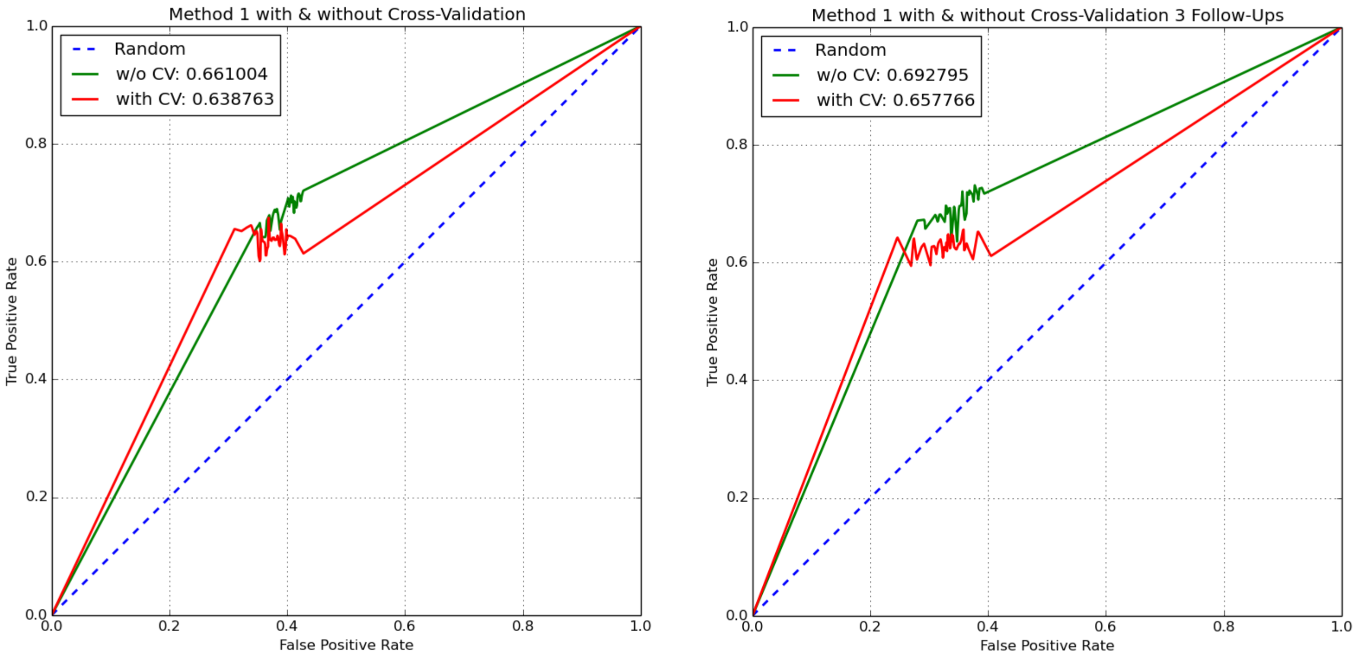
In this section the results of the experiments for all methods are summarized and presented. The following diagrams contain the metrics of the different methods being tested with and without the use of cross-validation on the training data and with $5, 7$ and 10-fold cross-validation on the SVM training (when applicable). Additional metrics for the subjects with the maximum number of follow-up scans (three follow-ups) are produced and shown.

3.3.1 Method I

In Figures 3.2a & 3.2b we can see the ROC curves produced when testing the first method for HMMs with 2 up-to 40 states. The method's results are in general satisfactory. We can see that the use of part of the MCI subjects for the training of the HMMs by means of cross-validation did not improve the overall performance of the method in terms of the area under the ROC curve (ROC-AUC) (0.661 for the simple method, over 0.638 for the cross-validation). In both cases, subjects with 3 follow-up scans produce better classification results.

Figures 3.3, 3.4, 3.5 &3.6 show the Sensitivity and Specificity measurement for increasing numbers of states. A general trend, though not very intense is that both metrics improve or remain constant up-to a certain number of states (~$20 - 25$) and then decrease. A common approach with these two metrics is to try to keep them at relative close numbers and as high as possible. It is of course desirable that they both are very high, but except for a perfect classifier, they tend to have an inverse behaviour (over-classifying within one class usually means under-classifying in the other). Thus, it is commonly accepted to compromise on one or the other, so that we can have a better overall performance. In our results, in the Sensitivity/Specificity diagrams, we indicate a 0.6 and 0.7 threshold.

As mentioned earlier, the graphs tend to decrease after a number of states has been used. It is worth noting that, when cross-validation is used, the performance is more stable and neither metric fall under the limit of 0.6, except for one occasion. On the other hand, when cross-validation is not used, Specificity decreases quite a lot after 20 states diverging a lot from sensitivity, whose quite high values improve the overall state of the ROC curve, hence producing higher ROC-AUC.



(a) Entire MCI Group

(b) MCI Subjects with 3 Follow-ups

Figure 3.2: ROC Curves of Method I

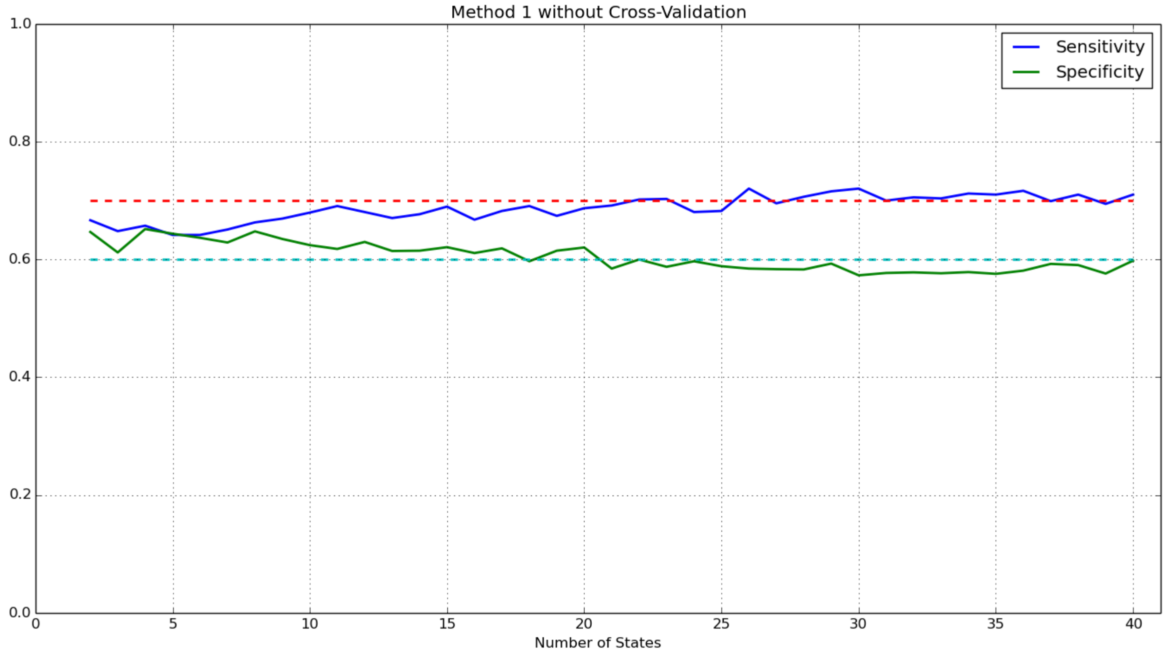


Figure 3.3: Sensitivity and Specificity of Method I without Cross-Validation for an increasing number of HMM states

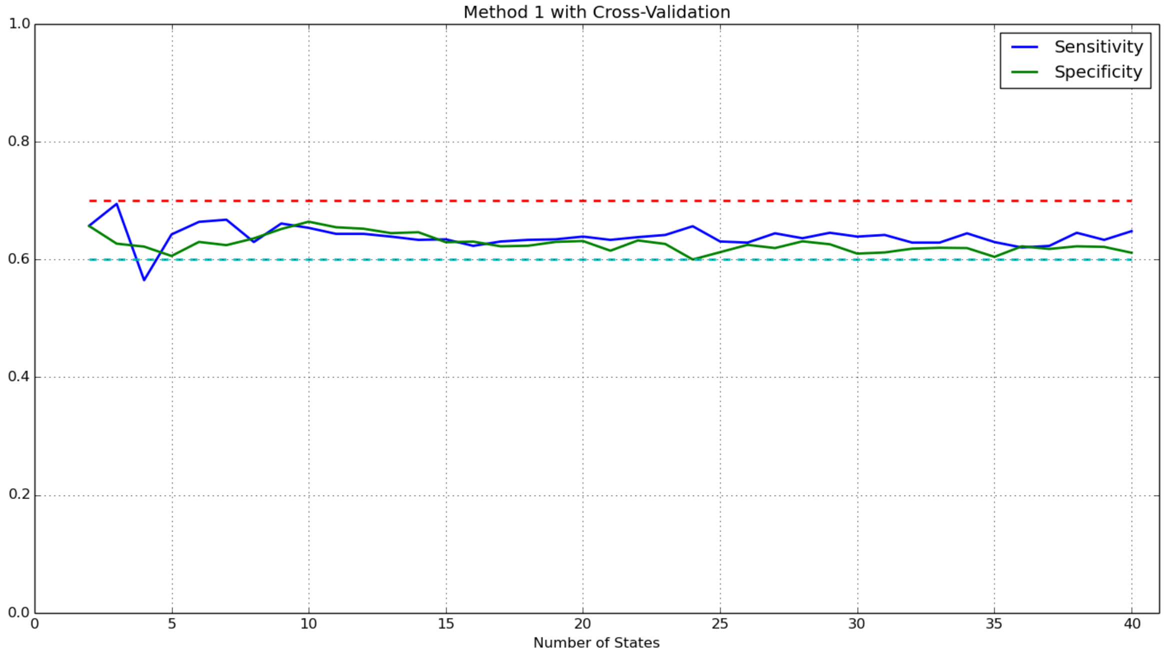


Figure 3.4: Sensitivity and Specificity of Method I with Cross-Validation for an increasing number of HMM states

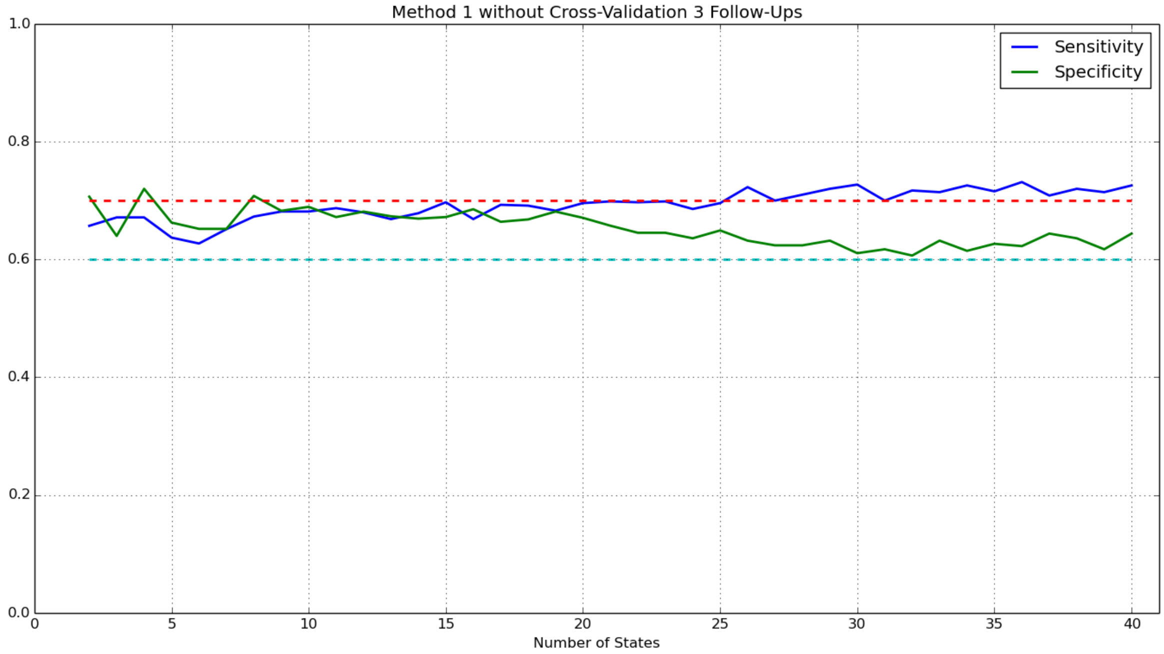


Figure 3.5: Sensitivity and Specificity of Method I for subjects with 3 Follow-ups without Cross-Validation for an increasing number of HMM states

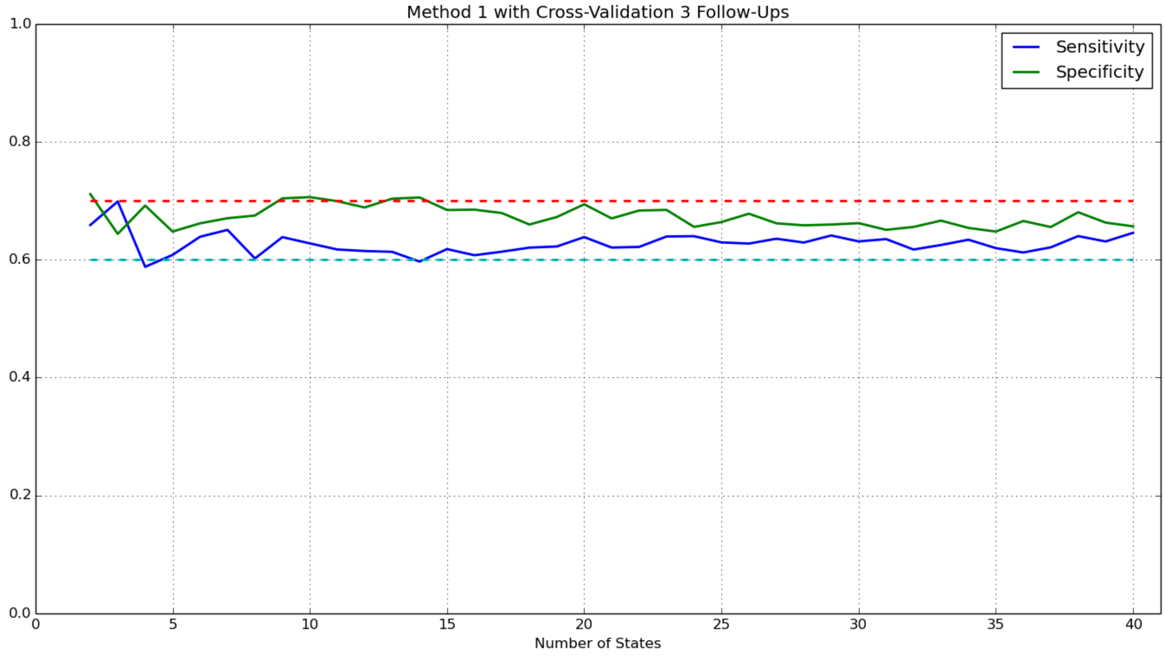


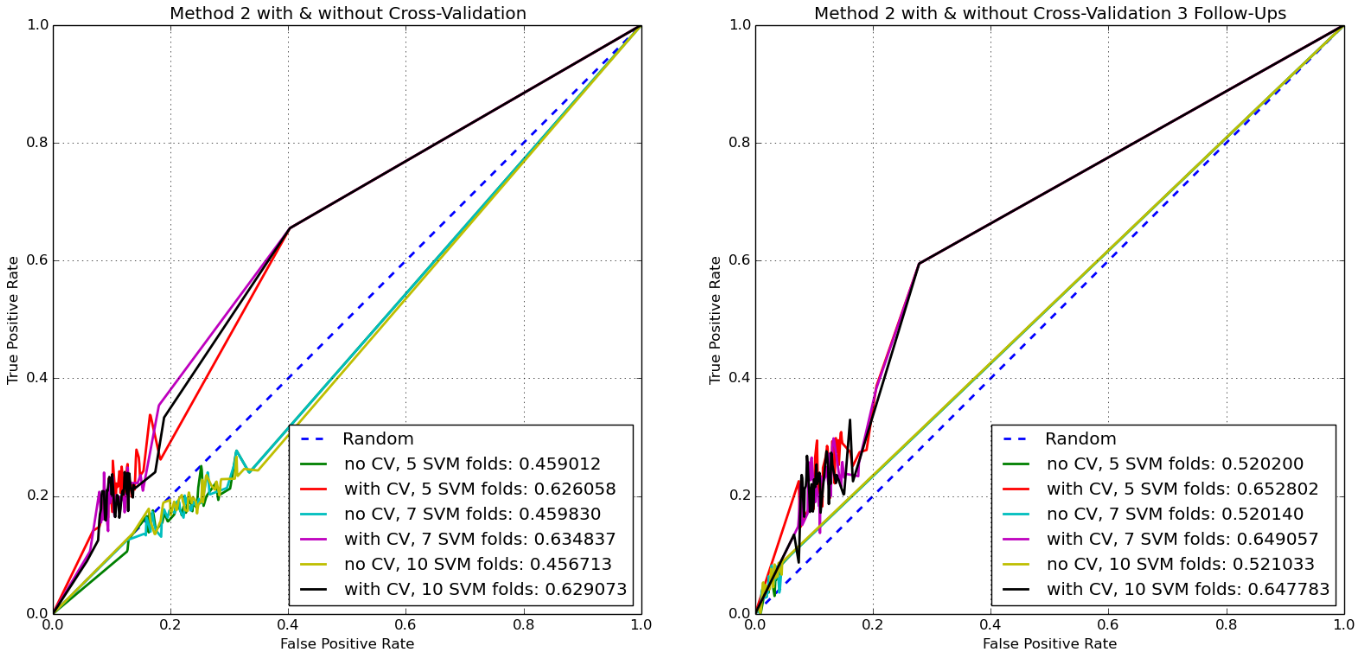
Figure 3.6: Sensitivity and Specificity of Method I for subjects with 3 Follow-ups with Cross-Validation for an increasing number of HMM states

3.3.2 Method II

In Figures 3.7a and 3.7b we see the ROC curves for the second method and for different numbers of folds for the SVM classification. The first thing to notice from these graphs is that the method performs very badly. When no cross-validation is used and the method is tested on all the data, it performs worse that a random classifier, basically having more misclassifications than correct classifications. In this case, the cross-validation significantly improves the ROC curve, but the AUC still remains lower than the first method's. As before the subjects with 3 follow-ups produce better results.

When examining Figures 3.8, 3.9, 3.10 & 3.11, the graphs seem much more interesting. We can see that the second method gives very high Specificity (almost 1 for subjects with 3 follow-ups), but very low Sensitivity. This phenomenon was discussed in the previous section. The SVM cross-validation does not seem to have a significant impact on the performance.

After further examining the results and studying the confusion matrices that have been produced (Figures 3.12a & 3.12b), we saw that the classifier basically classified most data as non-AD, which is the cause of the high Specificity/low Sensitivity. As mentioned in Section 2.4.3 the state sequences that are produced by the HMMs and used as feature vectors for the SVM are highly variant which turns them into non-separable data. This causes SVM, which is a very strong classifier to fail in finding a hyperplane that separates them. Hence the need of Method III.



(a) Entire MCI Group

(b) MCI Subjects with 3 Follow-ups

Figure 3.7: ROC Curves of Method II for Different SVM Folds

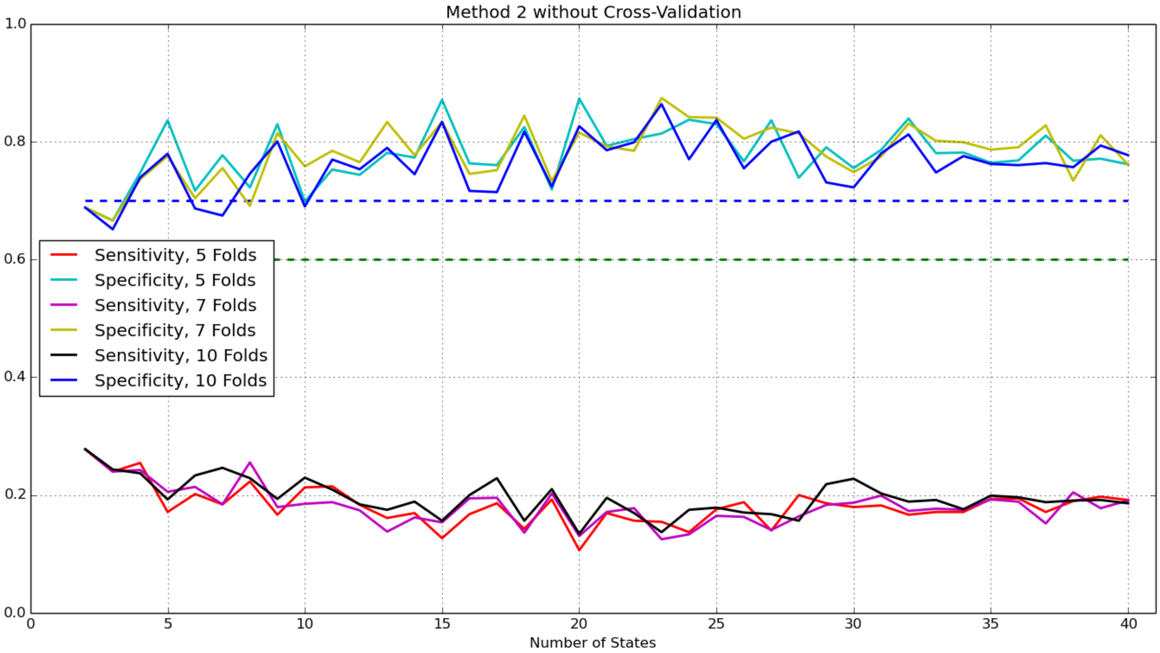


Figure 3.8: Sensitivity and Specificity of Method II without Cross-Validation for an increasing number of HMM states and Different SVM Folds

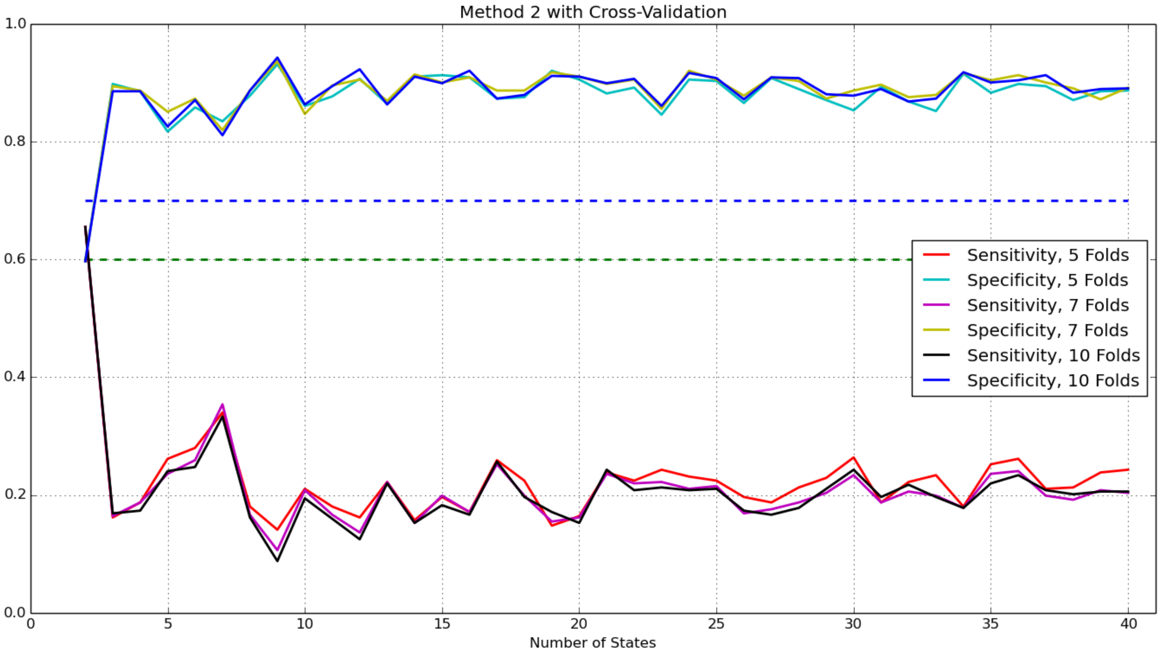


Figure 3.9: Sensitivity and Specificity of Method II with Cross-Validation for an increasing number of HMM states and Different SVM Folds

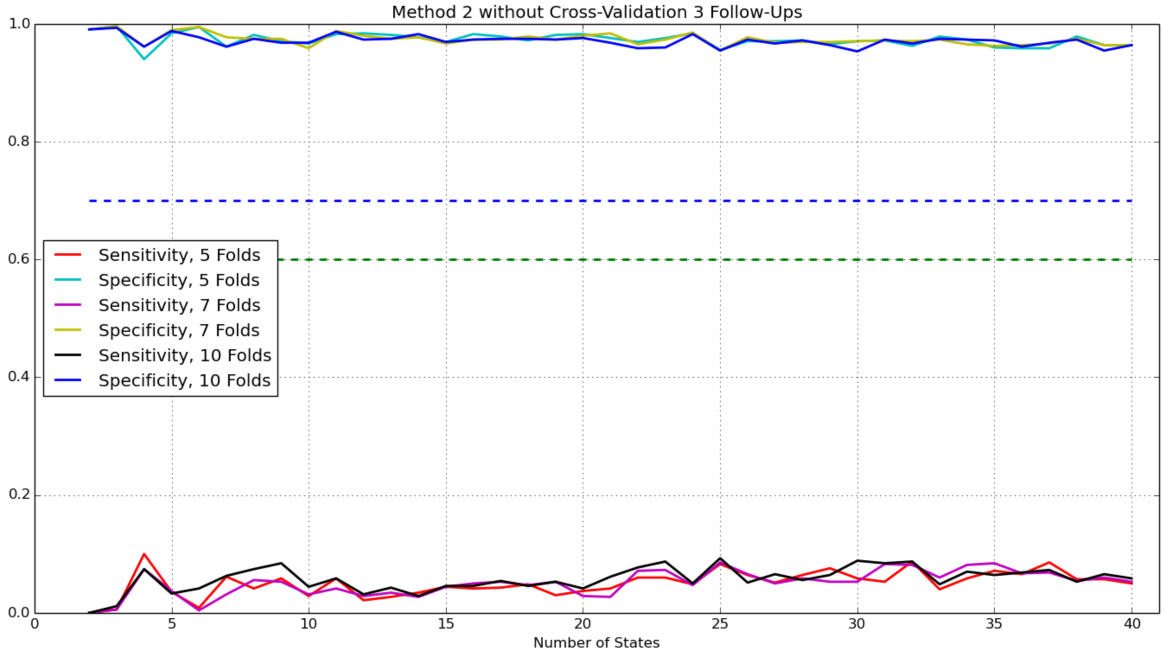


Figure 3.10: Sensitivity and Specificity of Method II for subjects with 3 Follow-ups without Cross-Validation for an increasing number of HMM states and Different SVM Folds

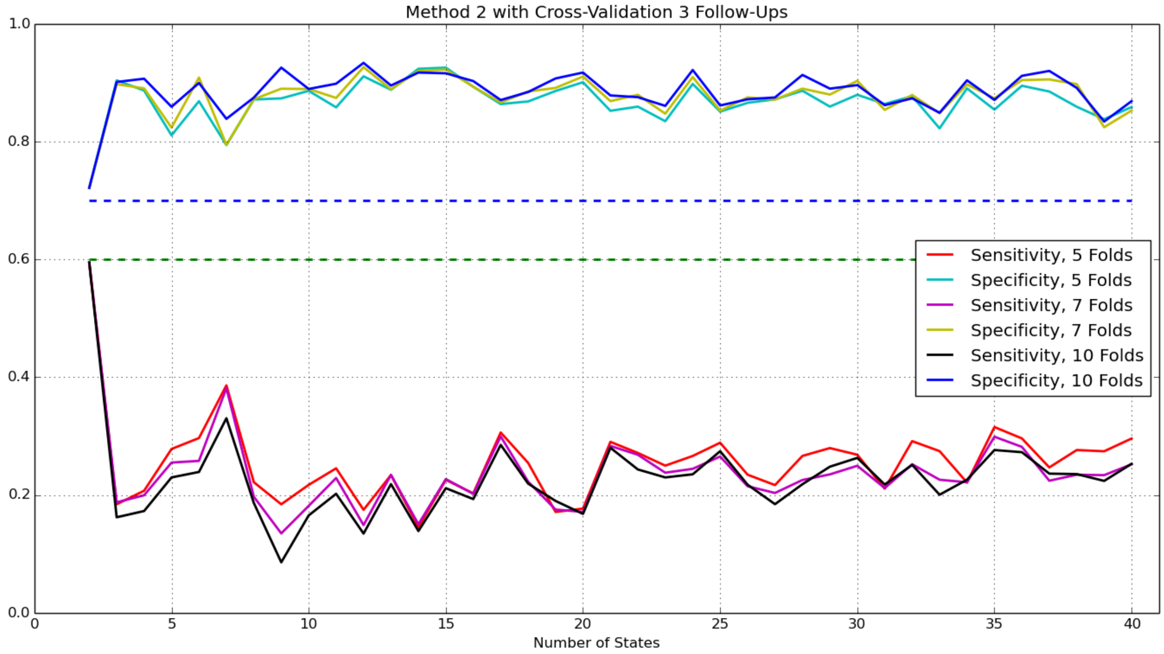
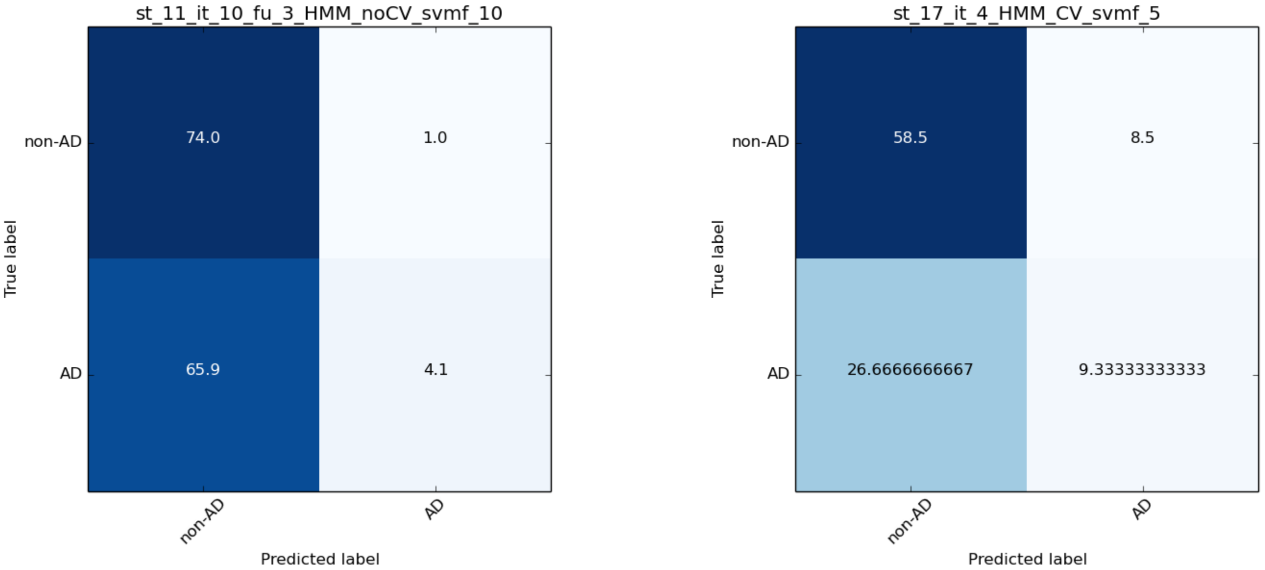


Figure 3.11: Sensitivity and Specificity of Method II for subjects with 3 Follow-ups with Cross-Validation for an increasing number of HMM states and Different SVM Folds



(a) 11 States without Cross-Validation, 10 SVM Folds & Subjects with 3 Follow-Ups

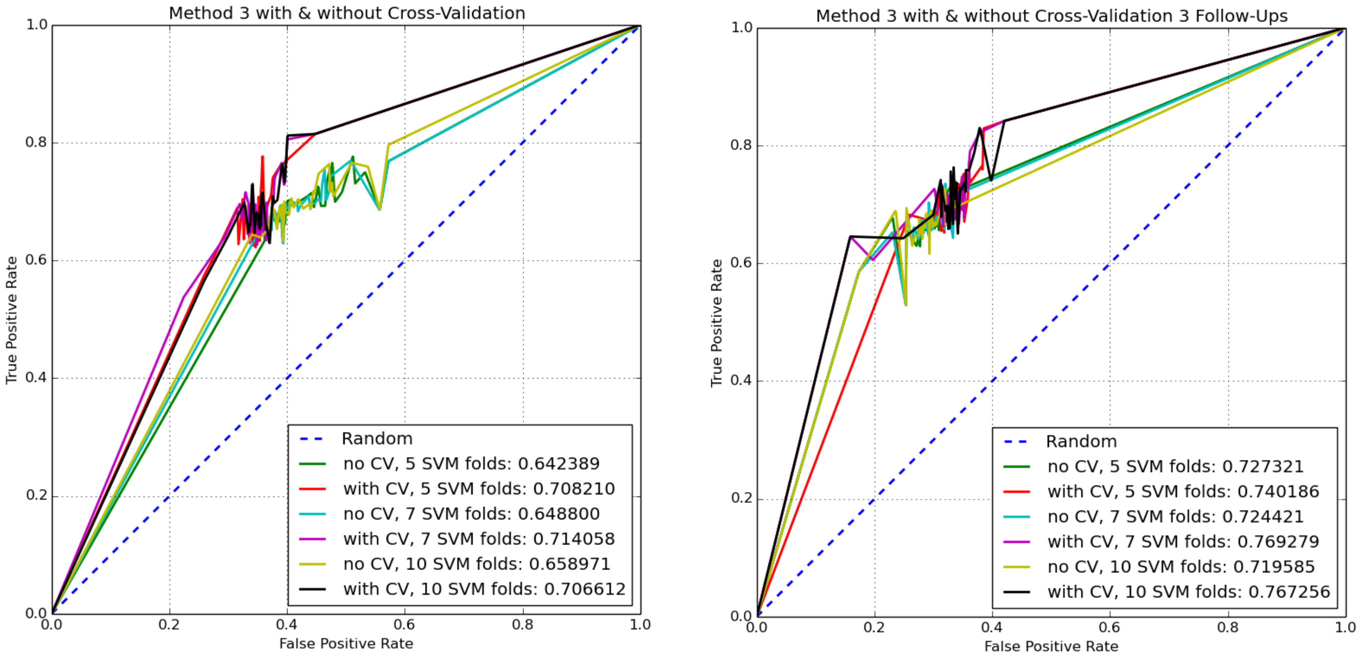
(b) 17 States with Cross-Validation, 5 SVM Folds & with all MCI Subjects

Figure 3.12: Confusion Matrices for Method II

3.3.3 Method III

As for the previous two methods, we first present the ROC curves of all the experiments ran for the third method. We can immediately observe that the curves show great improvement from the second method and significant improvement from the first. This indicates that the use of the frequency maps instead of the actual state sequences constitutes the data much more separable and still is able to carry significant information about the progression of the condition.

For this method we see an improvement of the ROC curves with the use of cross-validation and as in Method I it also helps with the divergence of Sensitivity and Specificity. Both metrics in this case are also decreasing as well as stabilising with the addition of more states to the HMM.



(a) Entire MCI Group

(b) MCI Subjects with 3 Follow-ups

Figure 3.13: ROC Curves of Method III for Different SVM Folds

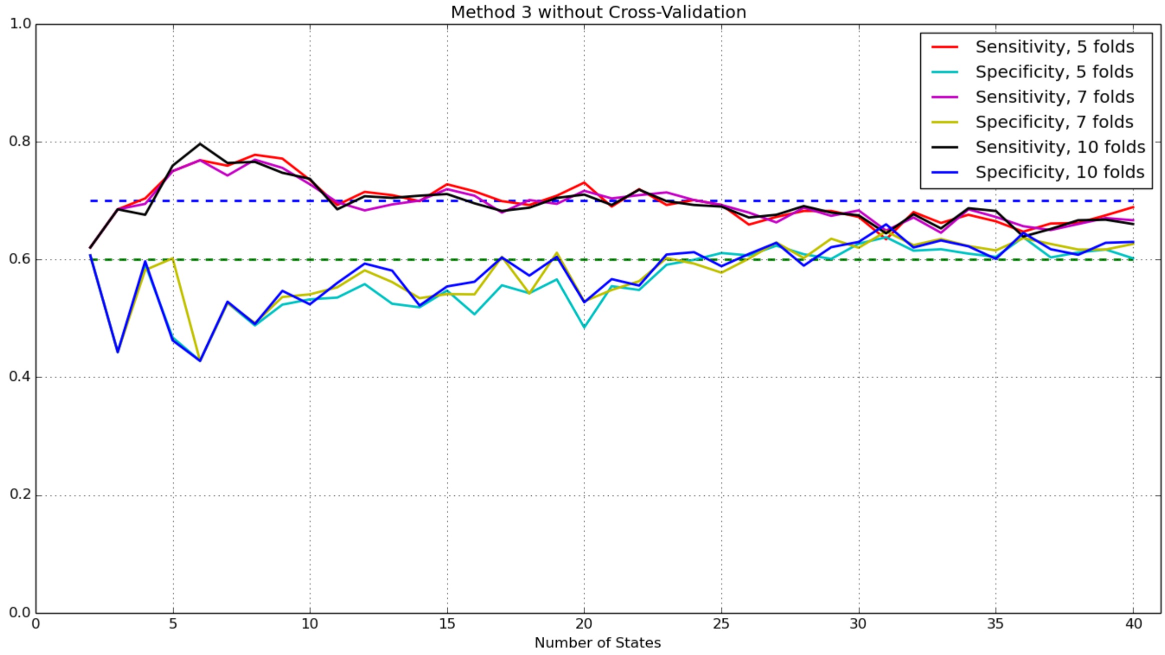


Figure 3.14: Sensitivity and Specificity of Method III without Cross-Validation for an increasing number of HMM states and Different SVM Folds

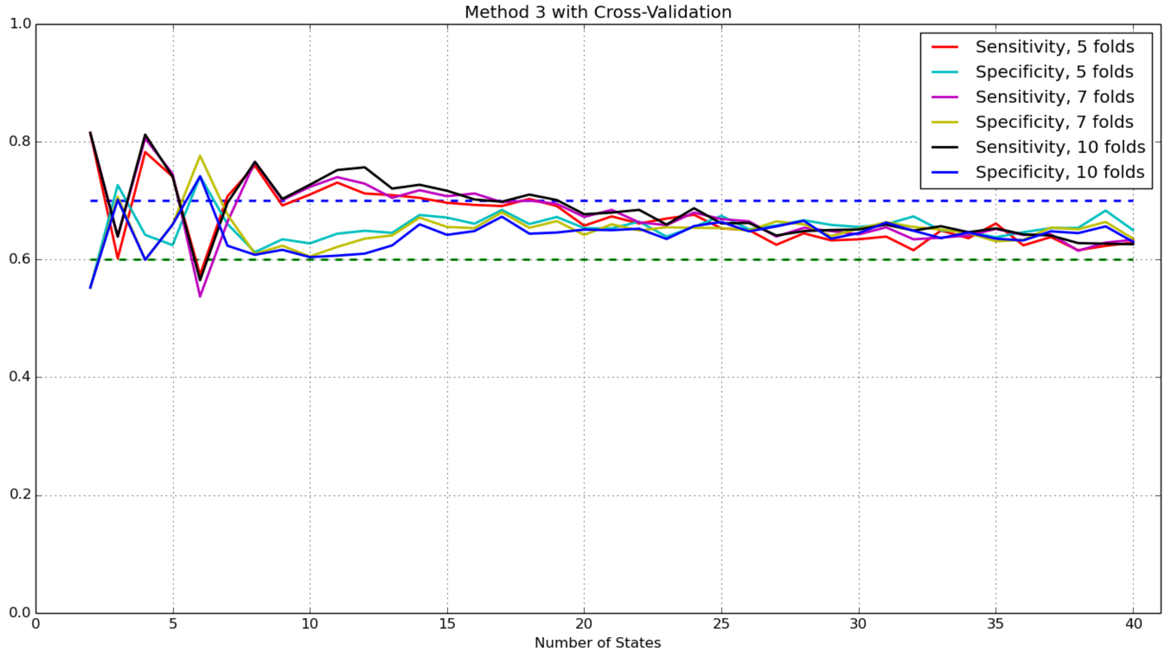


Figure 3.15: Sensitivity and Specificity of Method III with Cross-Validation for an increasing number of HMM states and Different SVM Folds

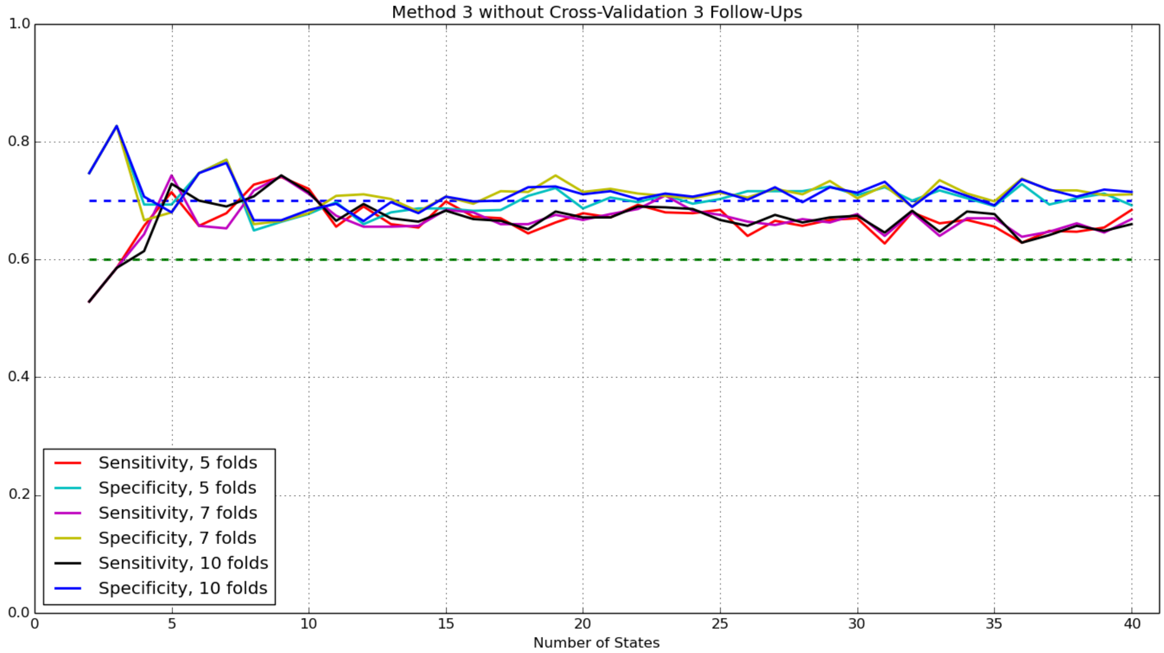


Figure 3.16: Sensitivity and Specificity of Method III for subjects with 3 Follow-ups without Cross-Validation for an increasing number of HMM states and Different SVM Folds

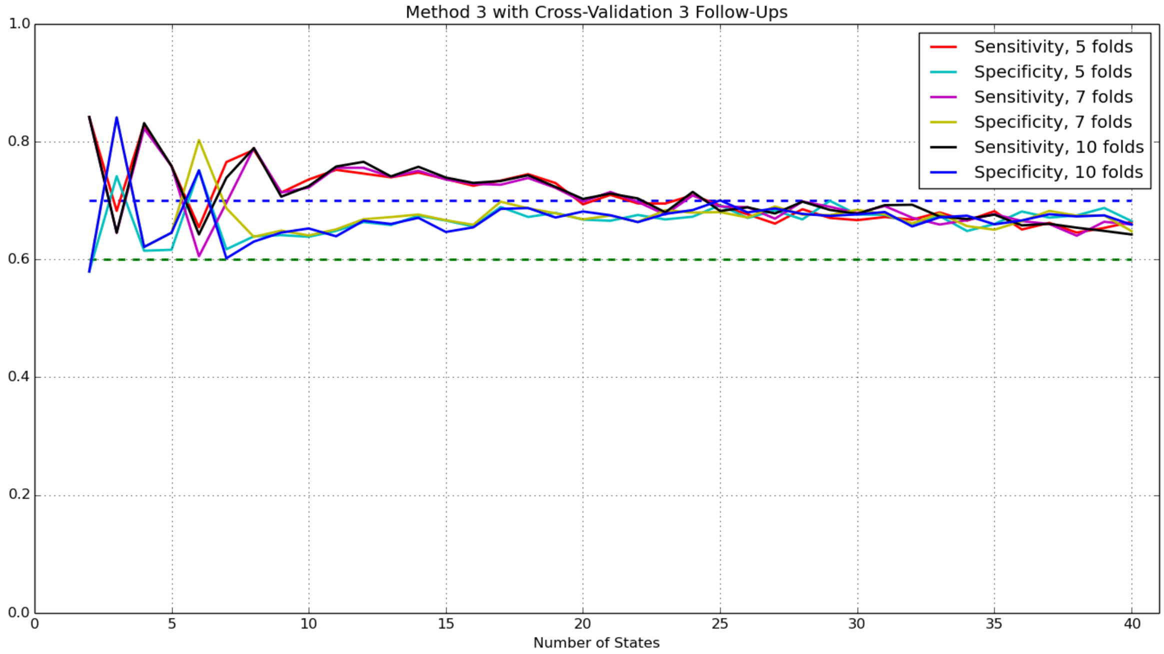


Figure 3.17: Sensitivity and Specificity of Method III for subjects with 3 Follow-ups with Cross-Validation for an increasing number of HMM states and Different SVM Folds

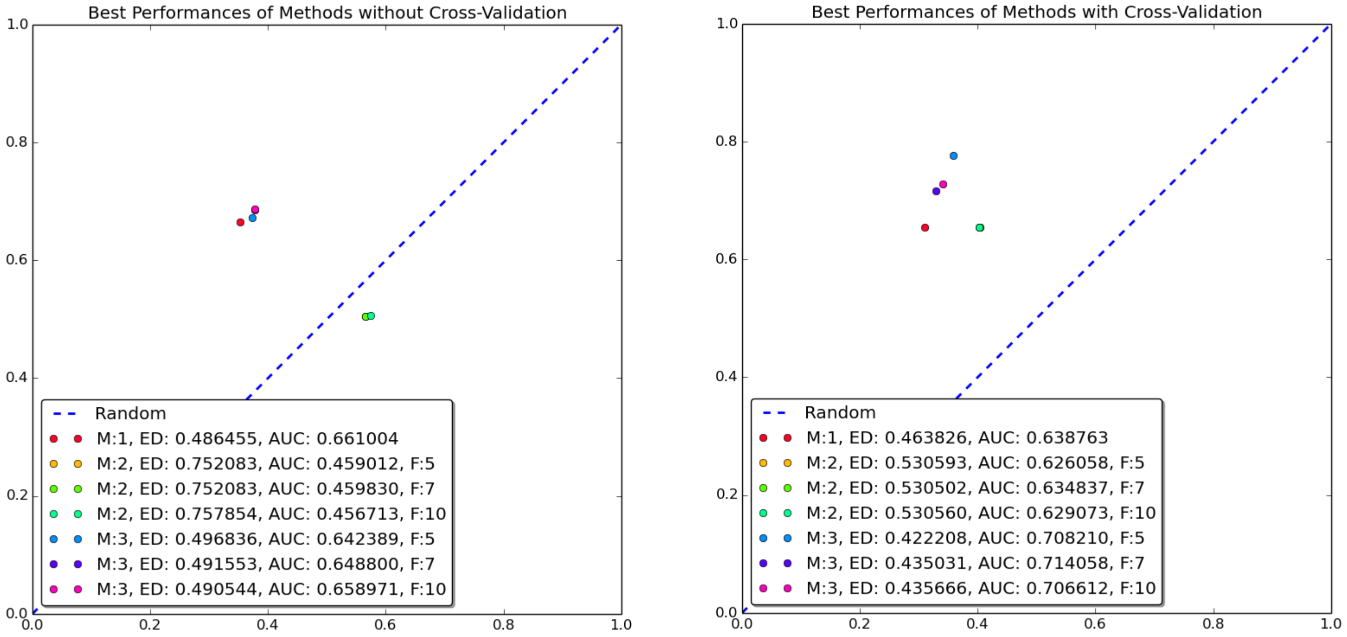
3.3.4 Results' Summary

Up until now, the results and the evaluation that has been done was on a more theoretical level and concerned the methods as an overall scientific search (ROC curves and Sensitivity/Specificity graphs produced for all numbers of states). However, on a more practical level, if these methods were to be used, then we would be more interested in specific instances of the classifiers that have been trained and tested. Meaning that for this application it would be more fitting to compare and contrast (and in the end, actually use) the instances of each method that produced that best results.

For this we have produced Figures 3.18 & 3.19, where we visualise an ROC space with the data points of these instances of each method. The different data-points in these Figures correspond to the true positive rate and 1−true negative rate of each method's best instance. In the legend the different methods are indicated by the letter M. We also provide the Euclidean distance of each point (ED) from the top left corner (the shorter the distance, the better the performance) and the area under the curve (AUC) of each method overall. These numbers are also summarized in Tables 3.1 & 3.2. It is easier to compare and contrast the different methods and approaches here, where the performance of all is visualised.

What we can deduce from Figures 3.18a and 3.18b is that the use of cross-validation significantly improves the performance of the different trained models, especially in the case of Method II. For this method, without cross-validation the performance falls below the marginally acceptable random classifier, whereas with cross-validation the same method is pushed very close to the other two, though with still worse performance. Cross-validation also helps the other two methods, for which the performance improves. We can also conclude that the use of different numbers of folds for the training of the SVM has very small impact on the performance, something that was also commented earlier.

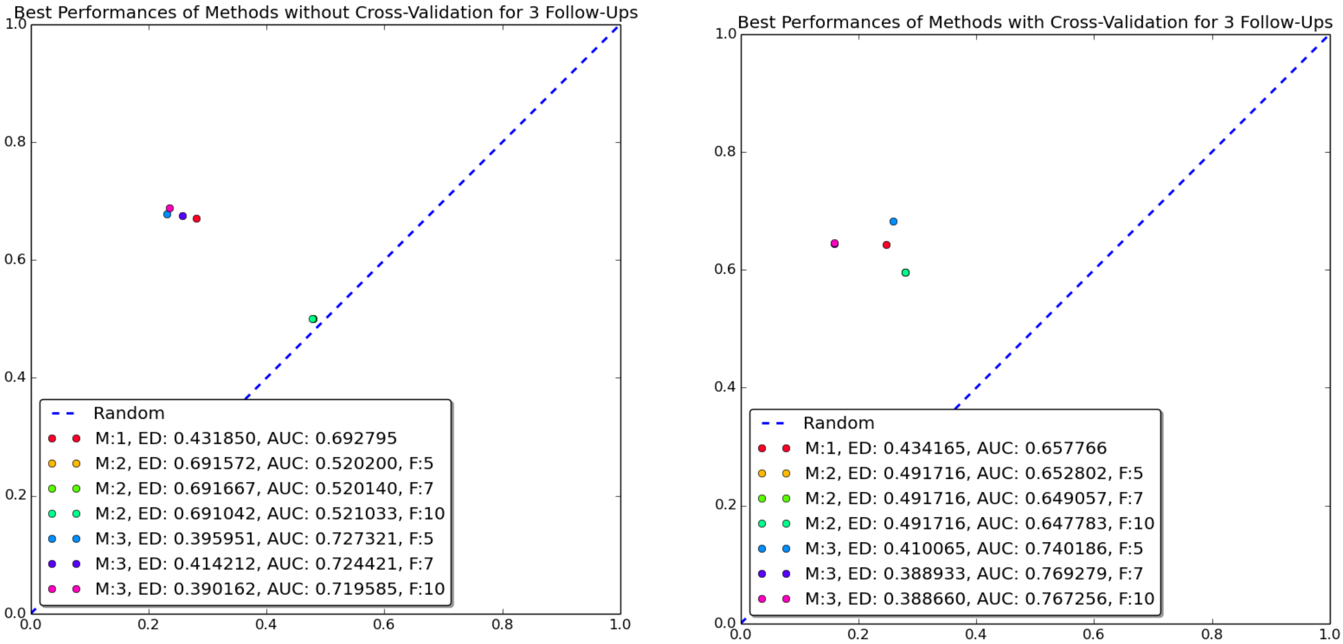
The results of the testing on subjects with 3 follow-ups are improved in this section as well (Method II does not fall below the random classifier, even without cross-validation), something that stresses the importance of the the length of the scan sequences of the subjects.



(a) All Methods Without Cross-Validation

(b) All Methods With Cross-Validation

Figure 3.18: Best Performances of all Methods



(a) All Methods Without Cross-Validation, Subjects with 3 Follow-ups

(b) All Methods With Cross-Validation, Subjects with 3 Follow-ups

Figure 3.19: Best Performances of all Methods, Subjects with 3 Follow-ups

Table 3.1: Summary of the best results of each method with respect to the closest point of the ROC curve to the upper left corner of the ROC space. The table shows the Specificity and Sensitivity that correspond to that point, as well as its Euclidean distance from the upper left corner and the overall area under the curve of the corresponding method (for SVM folds 5, 7 & 10).

METHOD	CV	SPECIFICITY	SENSITIVITY	DISTANCE	AUC
1	NO	0.646	0.665	0.486	0.661
1	YES	0.689	0.655	0.463	0.638
2	NO	0.434, 0.434, 0.425	0.504, 0.504, 0.505	0.752, 0.752, 0.757	0.459, 0.459, 0.456
2	YES	0.596, 0.597, 0.597	0.655, 0.654, 0.654	0.530, 0.530, 0.530	0.626, 0.634, 0.629
3	NO	0.626, 0.622, 0.622	0.672, 0.685, 0.686	0.496, 0.491, 0.490	0.642, 0.648, 0.658
3	YES	0.641, 0.670, 0.659	0.776, 0.715, 0.728	0.422, 0.435, 0.435	0.708, 0.714, 0.706

3.4 Discussion & Future Work

The objective of the thesis was to implement a model that would be able to predict the progression of the condition of MCI patients by examining their longitudinal MRI scans. We can state that a satisfactory model has been implemented, which combines the HMM's ability to study longitudinal sequences and the SVM's strong discrimination capability. It is interesting to study the effect that the different approaches have on the model's performance, as well as the length of the sequences.

At this point, it is imperative to discuss the difficulties that the data-set poses on completing the objective. One important aspect is the size of it. In computer science it is always important to have a plethora of data on which we can train our models, however that is difficult to achieve for medical data, for different reasons. In the case of our data-set, one issue is that, the MRI scan itself is costly and not

Table 3.2: Summary of the best results of each method with respect to the closest point of the ROC curve to the upper left corner of the ROC space. The table shows the Specificity and Sensitivity that correspond to that point, as well as its Euclidean distance from the upper left corner and the overall area under the curve of the corresponding method (for SVM folds 5, 7 & 10). Subjects with 3 Follow-ups

METHOD	CV	SPECIFICITY	SENSITIVITY	DISTANCE	AUC
1	NO	0.719	0.671	0.431	0.692
1	YES	0.753	0.642	0.434	0.657
2	NO	0.521, 0.521, 0.522	0.500, 0.500, 0.500	0.691, 0.691, 0.691	0.520, 0.520, 0.521
2	YES	0.720, 0.720, 0.720	0.595, 0.595, 0.595	0.491, 0.491, 0.491	0.652, 0.649, 0.647
3	NO	0.769, 0.742, 0.764	0.677, 0.675, 0.688	0.395, 0.414, 0.390	0.727, 0.724, 0.719
3	YES	0.740, 0.840, 0.840	0.682, 0.645, 0.645	0.410, 0.388, 0.388	0.740, 0.769, 0.767

easy to obtain, therefore it is not easy to create a large data-set. Additionally, the process of segmenting the scans in order to obtain the volumetric features that we need, performed by Freesurfer, is very time-consuming. Each scan needs $12 - 14$ hours of preprocessing before producing our features, making it also hard to enlarge our dataset.

The same reasons, combined with personal or practical reasons for which a patient would not continue the participation in the MRI scanning process, also affect the number of follow-up scans of each patient. The length of the sequences is very important, because the HMM works on a probabilistic basis, which means that it needs to observe large quantities of data in order for it to be able to stabilize its behaviour, and in this particular case the quantity of the data refers to the number of observations for each subject, rather than the number of the subjects. It is highly possible that with long enough observation sequences, the first method would outperform both other and could produce much better results.

One more issue, that consequently affects the obtained data, is the fact that a definite AD diagnosis can only occur post-mortem with a brain biopsy. There have been occasions, where the patient's diagnosis was proven incorrect by a post-mortem examination, meaning that a diagnosis in vivo cannot be trusted absolutely, at least with the diagnostic means available today. Within the current data-set, it is not possible at the moment to know with certainty, that the provided diagnosis is correct or not. This brings about a type of “chicken & egg” problem: in order to get better data, we need to find a way for a way of diagnosing AD in vivo with certainty, but in order to achieve that we need better data.

There are different ideas and ways to attempt, in order to improve the performance achieved by this thesis, as well as, ways to address the issues discussed in the previous paragraphs, or at least soften their impact. A speed-up modification can be attempted on the Freesurfer package, potentially by parallelizing its procedure, so that the scan processing time can be significantly reduced. This can provide researchers with more and longer longitudinal sequences.

A small change in the data-set can also help with the uncertainty over the diagnoses. A flag can be provided with each scan indicating whether it bears a verified diagnosis or not. Using this flag the model can be adjusted by putting weight factors on the data, and weighing more the subjects with definite diagnoses, or even implementing a semi-supervised method in order to take advantage of more data, even unlabelled.

Additional pre-processing can be applied on the 55-long feature vectors, in the form of clustering or any dimension reducing technique, so that more compact and representative features can be used for the prediction process.

Finally, in the field of neural networks, there are many exciting ways to address the thesis' objective, for example with the use of Recurrent Neural Networks (RNN), that have the ability to model temporal and sequential data, without the demand of fixed length for the input. The RNNs are currently used in certain applications of speech recognition [22]. Deep neural networks (DeepNets) or even convolutional neural networks (ConvNets) could also be ventured. DeepNets and ConvNets have the advantage, because of their complexity and size, that they are able to model more complex data with higher accuracy. A ConvNet is so strong as a model that it can possibly handle the initial MRI scan as input, without any preprocessing, and yet be able to make correct predictions about the course a condition. However an important demand of these types of networks is the size of the data provided, which needs to be very large. This fact renders their use out of reach for medical research, at least for the time being.

3.5 Society & Ethics

A long discussion can be done on the impact of medical research on society and its ethical boundaries. The most common arguments concern the aspect of consent and the protection of the patients' privacy. It is obvious that no research can be possibly done without the informed consent of a patient, or in general any person that will participate in an experiment. With proper care and handling, personal data can be protected and research can be done, when the objective is in the best interest of people.

Moreover, people argue on the matter of programs replacing doctors, which is something that can be unsettling. However, if proper methods are developed they can, instead of replacing a doctor, constitute a significant aid in the process of diagnosing or predicting certain conditions early enough, so as to render them treatable or even avoidable. In the specific case of the thesis' objective, a method that can predict progression to AD can be extremely valuable to a doctor and a patient.

Bibliography

[1] Ying Chen and Tuan D Pham. Sample entropy and regularity dimension in complexity analysis of cortical surface structure in early alzheimer's disease and aging. *Journal of Neuroscience Methods*, 215:210–217, May 2013.

[2] Simon Duchesne, Anna Caroli, Christina Geroldi, D. Louis Collins, and Giovanni B. Frisoni. Relating one-year cognitive change in mild cognitive impairment to baseline MRI features. *Neuroimage*, 47:1363–1370, October 2009.

[3] Farshad Falahati, Eric Westman, and Andrew Simmons. Multivariate data analysis and machine learning in alzheimer's disease with a focus on structural magnetic resonance imaging. *Journal of Alzheimer's Disease*, 41:685–708, March 2014.

[4] Chong-Yaw Wee, Pew-Thian Yap, Dinggang Shen, and Alzheimer's Disease Neuroimaging Initiative. Prediction of alzheimer's disease and mild cognitive impairment using cortical morphological patterns. *Hum Brain Mapp.*, December 2013.

[5] Daoqiang Zhang, Yaping Wang, Luping Zhou, Hong Yuan, Ginggang Shen, and Alzheimer's Disease Neuroimaging Initiative. Multimodal classification of alzheimer's disease and mild cognitive impairment. *Neuroimage.*, April 2011.

[6] Laboratory for Computational Neuroimaging. Freesurfer. Athinoula A. Martinos Center for Biomedical Imaging, 2013. `https://surfer.nmr.mgh.harvard.edu/`.

[7] M. Jenkinson, C.F. Beckmann, T.E. Behrens, M.W. Woolrich, and S.M. Smith. FSL. *NeuroImage*, 62:782–790, 2012. `https://http://fsl.fmrib.ox.ac.uk/`.

[8] Wellcome Trust Centre for Neuroimaging. SPM. `http://www.fil.ion.ucl.ac.uk/spm/`.

[9] Carlos Aguilar, Eric Westman, J-Sebastian Muehlboeck, Patrizia Mecocci, Bruno Vellas, Magda Tsolaki, Iwona Kłoszewska, Hilkka Soininen, Simon Lovestone, Christian Spenger, Andrew Simmons, and Lars-Olof Wahlund. Different multivariate techniques for automated classification of MRI data in alzheimer's disease and mild cognitive impairment. *Psychiatry Research: Neuroimaging*, 212:89–98, May 2013.

[10] Gabriela Spulber, Andrew Simmons, J-Sebastian Muehlboeck, Patrizia Mecocci, Bruno Vellas, Magda Tsolaki, Iwona Kłoszewska, Hilkka Soininen, Christian Spenger, Simon Lovestone, Lars-Olof Wahlund, and Eric Westman. An MRI-based index to measure the severity of alzheimer's disease-like structural pattern in subjects with mild cognitive impairment. *Journal of internal medicine*, 273:396–409, January 2013.

[11] Javier Escudero, John P. Zajicek, and Emmanuel Ifeachor. Machine learning classification of MRI features of alzheimer's disease and mild cognitive impairment subjects to reduce the sample size in clinical trials. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 7957–7960. IEEE, 2011.

[12] Kota Katanoda, Yasumasa Matsuda, and Morihiro Sugishita. A spatio-temporal regression model for the analysis of functional MRI data. *NeuroImage*, 17:1415–1428, November 2002.

[13] Alicia Quiros, Raquel Montes Diez, and Dani Gamerman. Bayesian spatiotemporal model of fmri data. *NeuroImage*, 49:442–256, January 2010.

[14] Ying Chen and Tuan D Pham. Development of a brain mri-based hidden markov model for dementia recognition. *BioMedical Engineering OnLine*, 12:S2, April 2013.

[15] Bing Wang and Tuan D Pham. MRI-based age prediction using hidden markov models. *Journal of Neuroscience Methods*, 199:140–145, July 2011.

[16] Ying Wang, Ssan M. Resnick, and Christos Davatzikos. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, chapter Spatio-temporal Analysis of Brain MRI Images Using Hidden Markov Models, pages 160–168. Springer Berlin Heidelberg, 2010.

[17] Susanne G. Mueller, Michael W. Weiner, Leon J. Thal, Ronald C. Petersen, Clifford R. Jack, William Jagust, John Q. Trojanowski, Arthur W. Toga, and Beckett Laurel. Ways toward an early diagnosis in alzheimer's disease: The alzheimer's disease neuroimaging initiative (adni). *Alzheimer's & Dementia*, 1(1):55–66, July 2005.

[18] Farshad Falahati, Daniel Ferreira, Hilkka Soininen, Patrizia Mecocci, Bruno Vellas, Magda Tsolaki, Iwona Kłoszewska, Simon Lovestone, Maria Eriksdotter, Lars-Olof Wahlund, Andrew Simmons, and Eric Westman. The effect of age correction on multivariate classification in alzheimer's disease, with a focus on the characteristics of incorrectly and correctly classified subjects. *Brain Topography*, 29(2):296–307, October 2015.

[19] Bruce Fischl. Freesurfer. *NeuroImage*, 62(2):774–81, August 2012.

[20] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[21] Andrew Zisserman. Lecture notes, 2015. University of Oxford, Information Engineering, Machine Learning Course.

[22] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778, 2013. URL `http://arxiv.org/abs/1303.5778`.

[23] M Symms, H Jager, K Schmierer, and T Yousry. A review of structural magnetic resonance neuroimaging. *Journal of Neurology, Neurosurgery, and Psychiatry*, 75, September 2004.

[24] IMAIOS SAS. Mri step-by-step, interactive course on magnetic resonance imaging. Online Course. `https://www.imaios.com/en/e-Courses/e-MRI`.

[25] Amiya Sarkar. Understanding the basic principles of nuclear magnetic resonance imaging. Online Blog, June 2010. `http://physiology-physics.blogspot.se/2010/06/understanding-basic-principles-of.html`.

[26] Anne Brown Rodgers. *Alzheimer's Disease: Unraveling the Mystery*. National Institute on Aging, September 2008.

[27] Charles M. Grinstead and J. Laurie Snell. *Introduction to probability*, chapter 11, pages 405–470. John Ewing, 2nd edition.

[28] Susan M. Resnick, Dzung L. Pham, Michael A. Kraut, Alan B. Zonderman, and Christos Davatzikos. Longitudinal magnetic resonance imaging studies of older adults: a shrinking brain. *Journal of Neuroscience Methods*, April 2003.

[29] John P. Cunningham and Byron M. Yu. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17:1500–1509, February 2014.

[30] Lawerence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286. IEEE, February 1989.

[31] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *Journal of Pattern Recognition and Artificial Intelligence*, 15:9–42, 2001.

[32] Narada Dilp Warakagoda. A hybrid ANN-HMM ASR system with NN based adaptive preprocessing. Master's thesis, Norges Tekniske Høgskole, May 2010.

[33] Arne Leijon and Gustav Eje Henter. *Pattern Recognition - Fundamental Theory & Exercise Problems*. KTH - School of Electrical Engineering, 2015. Lecture Notes for the course EQ2340.

[34] Sayed Mohammad Ebrahim Sahraeian and Byung-Jun Yoon. A novel low-complexity hmm similarity measure. *IEEE Signal Processing Letters*, 18(2):87–90, February 2011.

Appendix A

Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) is an imaging technique, which is becoming more and more popular in the medical field both in diagnostics and in research and also in many cases in producing images of non-living objects. Due to its non-invading nature and lack of use of ionizing radiation, it is favoured over other techniques. The high resolution images, the ability to portray different physiologies and other features (functional MRI, real-time MRI etc.) only add to the list of its benefits.

The Science Behind the MRI

The MRI was first used in vivo by the end of the 1970's [23] and has ever since been constantly researched in order to improve its efficiency (speed, image resolution etc.) and to also expand its applicability and features. However, the phenomenon on which the MRI is based, the Nuclear Magnetic Resonance (NMR), was first observed in 1945.

Nuclear Magnetic Resonance

As is widely known, the atoms are composed by the electrons (negatively charged particles) that are orbiting around the atom's nucleus (its core). The nucleus is composed by the protons (positively charged particles) and the neutrons (charge-less particles). The protons (hydrogen nuclei), due to their charge, behave like tiny rotating magnets, producing a microscopic magnetic field around them. This is exactly how we can perceive them in a macroscopic world, by their charge, their nuclear spin and they are commonly represented by a vector that coincides with the rotation axis of the nucleus. The sum of the microscopic magnetic fields generated by all the protons is called **net magnetization**.

Normally, the different fields are arbitrarily oriented, resulting in a **null net magnetization** (charge-less) (Figure A.1). When an external magnetic field (B_0) is applied to the protons, then all the spins align to that external field, the same way that small magnets would align to a similar field. As seen in Figure A.2[1], some spins align with the direction of the field (**parallel**) and some align with the opposite direction (**anti-parallel**) (also known as spin-up and spin-down positions respectively).

While the aligned protons spin, they actually revolve (**precess**) around the direction of the magnetic field (B_0) at an angle, while at the same time they rotate around their own axis. This phenomenon is shown in Figure A.3. The precessional frequency or resonance frequency is called **Larmor frequency** and is proportional to the external magnetic field's intensity.

So, the proton's vector can be broken down into two components, a longitudinal and a transverse component. The rotation of the transverse component is what gives the precession.

[1] Figures A.1 & A.2 found at `http://bio.groups.et.byu.net/mri_training_b_Alignment_in_Magnetic_Fields.phtml`

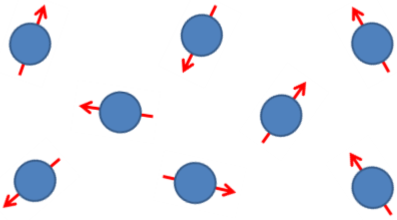


Figure A.1: Null net magnetization

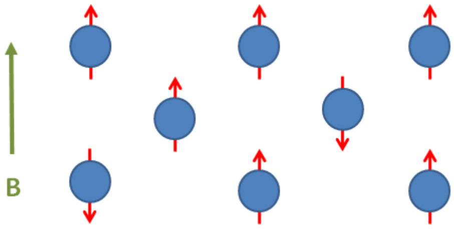


Figure A.2: Nuclei aligned to external field

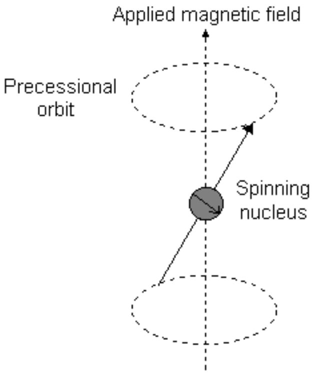


Figure A.3: Larmor frequency

In general protons tend to align parallel to the field in larger numbers than anti-parallel to the field, resulting in a net magnetization of a certain strength along the longitudinal axis (Z axis). On the other hand, since the protons are not rotating in phase, the sum of their transverse spins results in a null magnetization on the XY plane.

At this point if a second magnetic field (B_1), with perpendicular direction to B_0 and with higher intensity, is applied to the protons, they align to it. This means that we now have a new longitudinal net magnetization of a certain intensity (along the direction of B_1) and a new transverse null magnetization. In other words the original longitudinal magnetization becomes null and the original transverse magnetization increases. This is called spin excitation.

After B_1 stops being applied, then the protons return gradually to their initial state of alignment with B_0 (spin relaxation). While relaxation happens, a small but detectable amount of electromagnetic energy is emitted by the protons (**NMR** signal). We can observe this phenomenon by two different points-of-view: the increase of the longitudinal magnetization to its original state and the decrease of the transverse magnetization to null.

When observing these changes, we can calculate the times needed for them to happen. We call $T1$ the time needed for the longitudinal magnetization to reach 63% of its final value and $T2$ the time needed for the transverse magnetization to reach 37% of its original value (lose 63% of its original value). $T2$ is shorter than $T1$ and since it is tissue-specific, it is unrelated to the strength of the applied field, while $T1$ gets longer as the field's intensity increases.

MRI Signal Recording

As said previously, the MRI takes advantage of the NMR. Specifically it takes advantage of the magnetic energy emitted by the protons during relaxation. When the patient enters the MRI machine, a large magnet that envelopes the patient produces the B_0 magnetic field. At the same time an electromagnetic radio frequency (RF) signal transmitter and receiver produces the B_1 magnetic field.

An important factor at this point is that the signal transmitted is oscillating with a specific frequency (**Resonance**[2]). This way, only the protons that spin with the desired frequency will respond to the signal causing the excitation and relaxation phenomena described in the previous section.

When the electromagnetic energy is emitted from the protons, the RF receiver detects and imprints it on an image. Since different tissues have different responses to the RF pulse, they emit energy of different strength. Additionally by knowing the $T1$ and $T2$ of different tissues and adjusting the intensity of the RF pulse accordingly, scientists are able to produces images where different tissues are distinct, or even adjust the contrast of the produced image, according to their current needs.

For example, a tissue with long $T1$ and $T2$ times (like water) is depicted dark when the RF pulse is low (called $T1$-weighted image) and bright when the RF pulse is high (called $T2$-weighted image). On the other hand, a tissue with short $T1$ and long $T2$ times (like fat) is bright in a $T1$-weighted image and gray in a $T2$-weighted image.

[2] Resonance is called the frequency at which one system will interact with another causing it to oscillate with a maximum amplitude. In this particular case, since we have an electromagnetic RF, it is called magnetic resonance.

Table A.1: Relaxation times for various tissues at 1.5 T

	$T1$ **(ms)**	$T2$ **(ms)**
Water	3000	3000
Gray matter	810	100
White matter	680	90
Liver	420	45
Fat	240	85

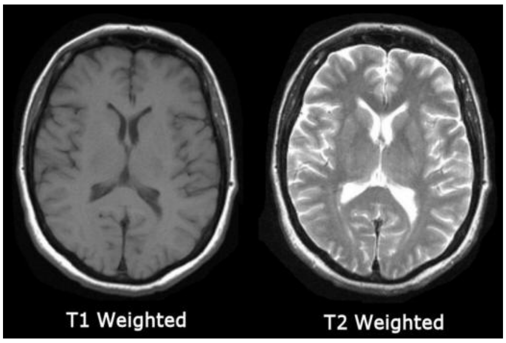


Figure A.4: MRI scan of a brain