Gait sequence modelling and estimation

using Hidden Markov Models

Presented by:
Kouame Hermann Kouassi

Prepared for:
Fred Nicolls
Dept. of Electrical and Electronics Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Electrical and Computer Engineering

November 2, 2017

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Signature:............................
Kouame H. Kouassi

Date:................................

Acknowledgments

Abstract

- Open the **Project Report Template.tex** file and carefully follow the comments (starting with %).
- Process the file with **pdflatex**, using other processors may need you to change some features such as graphics types.
- Note the files included in the **Project Report Template.tex** (with the .tex extension excluded). You can open these files separately and modify their contents or create new ones.
- Contact the latex namual for more features in your document such as equations, subfigures, footnotes, subscripts & superscripts, special characters etc.
- I recommend using the **kile** latex IDE, as it is simple to use.

Contents

Chapter 1

Introduction

1.1 Background to the study

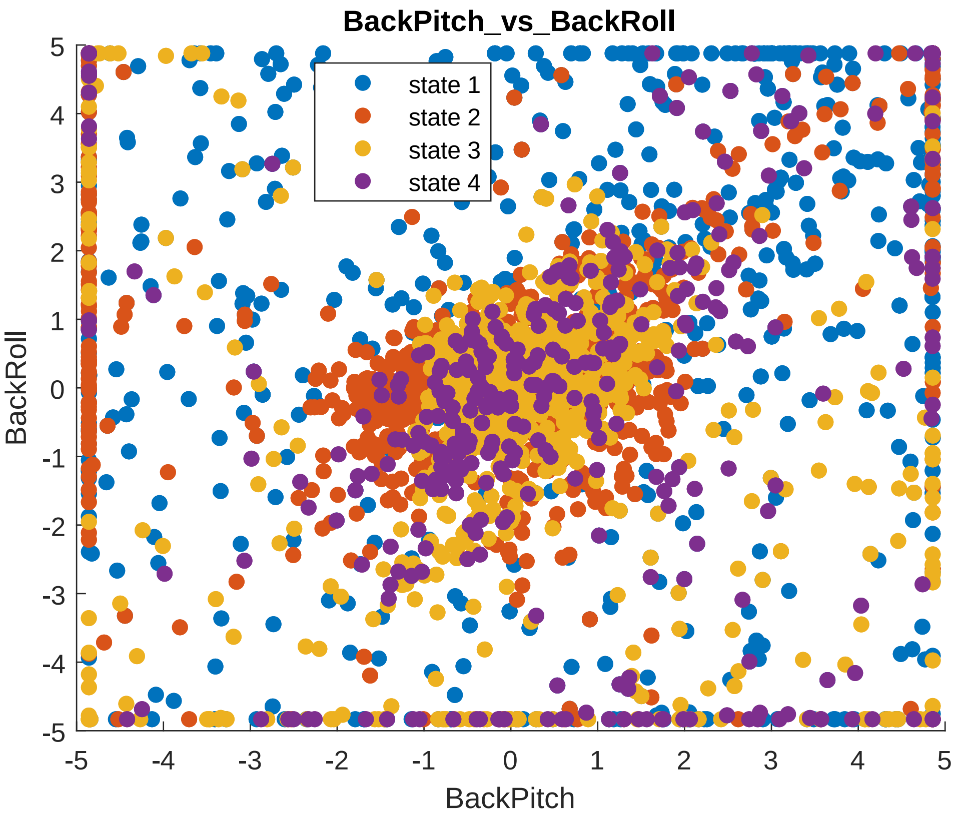
A very brief background to your area of research. Start off with a general introduction to the area and then narrow it down to your focus area. Used to set the scene [1].

Bio-inspired robotics uses nature to inform real-world engineering systems. Research has been conducted at UCT to investigate the manner in which a cheetah uses its tail for stability during high acceleration, quick turns and sudden braking, with an aim to incorporating identified mechanisms into sophisticated robot designs. One way to acquire useful data is to strap an inertial measurement unit (IMU) to an animal, and log the sensor data while certain actions are being performed. We currently have such a dataset of a dog moving, along with corresponding video data.

1.2 Objectives of this study

The objective of this project is to design, implement, and test Hidden Markov Models (HMM) for estimating gait sequence from Inertia Measurement Unit (IMU) data.

so that specific models can be formulated and their parameters estimated and interrogated. The project can be extended to include any other useful analysis of gait patterns from similar sensor measurements



1 - formulate model 2 - estimate its parameters 3 - Interrogate its parameters 4 - Useful analysis of gait patterns from IMU measurements

1.2.1 Problems to be investigated

Description of the main questions to be investigated in this study.

The main questions to be answered are the following:

1. How well can HMM model gait sequence dynamics using IMU data, in the abscence of enough training samples?
2. Can dimensionality reduction cause an increase in performance of HMM models when there is not enough training data?

1.2.2 Purpose of the study

Give the significance of investigating these problems. It must be obvious why you are doing this study and why it is relevant.

1.3 Scope and Limitations

Scope indicates to the reader what has and has not been included in the study. Limitations tell the reader what factors influenced the study such as sample size, time etc. It is not a section for excuses as to why your project may or may not have worked.

1 - Does not include data collection 2 - Focus on design of HMM only 3 - Focus on analysis of the model 4 - Focus on impact of dimensionality reduction

1.4 Plan of development

Here you tell the reader how your report has been organised and what is included in each chapter.

I recommend that you write this section last. You can then tailor it to your report.

Chapter 2

Literature Review

2.1 Gait sequence modelling and estimation

2.1.1 Quadrupede gait modelling

Periodicity

2.1.2 Quadrupede gait estimation

2.2 Case study: Inertia Measurement Unit

2.3 Hidden Markov Models

Hidden Markov Models (HMMs) are doubly embedded stochastic processes with a rich underlying statistical structure. Introduced at the end of the 1960s by Baum and colleagues, they have become one of the prefered techniques in speech recognition after the implementation of Baker and Jelinek in 1970s. HMMs have been successfully applied to various other engineering problems in pattern recognition for classification and fraud detection purposes, amongst others.
The type of HMM depends on the possible connections between the states. Thus, an HMM in which a state can transition to any other state is an ergodic. Other types such

as the Left-Right model or Bakis do not allow all possible transitions between the states.

2.3.1 HMM parameters specification

An HMM is fully specified by the following parameters

1. N, the number of distinct states of the model. Together they form the set of individual states $S = \{S_1, S_2, ..., S_N\}$.
2. T, the number of observations. A sample observation sequence is denoted as $O = \{O_1, O_2, ..., O_T\}$.
3. $Q = q_t$, the set of states with q_t denoting the current state at time instance, t such that $q_t \epsilon S$ and $t = 1, 2, ..., T$.
4. K, the number of distinct observation symbols per state.
5. $V = \{v_1, v_2, ..., v_K\}$, the feature set of K dimensions.
6. $A = \{a_{ij}\}$, the state transition probabilities. a_{ij} denotes probability of transitioning from state S_i to state S_j.
7. $\Phi = \{\phi_j(k\}$, the probability distribution of observation symbols in state j.
8. The initial state distribution, $\pi = \pi_i$

For continuous HMM (CHMM), i.e, HMM with continuous-valued observations, Φ consists in a probability distribution function. Many applications have succefully modelled such distributions with mixtures of Gaussian distributions. As such, ϕ is approximated by a weighted sum of M multivariate Gaussian distributions η. For a given, observation sequence, ϕ and η are therefore given by equations 2.1 and 2.2,

$$\phi(O_t) = \sum_{m=1}^{M} \beta_{jm}\eta(\mu_{jm}, \Sigma_{jm}, O_t), \tag{2.1}$$

$$\eta(\mu, \Sigma, O) = \frac{1}{\sqrt{(2\pi)^K|\Sigma|}} exp(-\frac{1}{2}(O-\mu)'\Sigma^{-1}(O-\mu) \tag{2.2}$$

$$1 \leq j \leq N; 1 \leq m \leq M; \beta_{jm} \geq 0; \sum_{m=1}^{M} \beta_{jm} = 1$$

where β_{jm} is the mixture composition coefficient; μ_{jm}, Σ_{jm}, respectively the mean vector and covariance matrix of state j; M is the number of mixture components and K is the dimensionality of O.

As a summary, the compact specification of a continous valued observation HMM is defined by 2.3 and that of a discrete HMM in 2.4.

$$CHMM = \lambda_C = (A, \beta_{jm}, \mu_{jm}, \Sigma_{jm}, \pi) \tag{2.3}$$

$$DHMM = \lambda_D = (A, b_j(k), \pi) \tag{2.4}$$

Basic assumptions of HMMs theory

HMM theory is built on three basic assumptions listed below.

1. *The Markov assumption*: HMM assumes that the probability of being in the current at any instance of time t, is uniquely dependent on the previous state, at time, t + 1. More specifically, $a_{ij} = P[q_t = S_j | q_{t+1} = S_i]$. This assumption makes it unsuitable for long-range correlation capturing applications.

2. *The stationary assumption*: Furthermore, HMM state transition probabilities are assumed to be time-independent. Thus, the transition probabilities of two distinct time, t_1 and t_2 are identical, $P[q_{t_1} = S_j | q_{t_1 - 1} = S_i] = P[qt_2 = S_i | q_{t_2 - 1} = S_i]$. HMMs can therefore effectively model mechanisms with stationary observations.

3. *The output/observation independence assumption*: The current observation also known as emission symbol is statistically independent of the previous observations. It is "emitted" only by the current state, $P[O|q_1, q_2, ..., q_T, \lambda] = \prod_{t=1}^{T} P[O_t|q_t, \lambda]$.

The three assumptions make an HMM model a relatively simple graphic modelling to be implemented. This simplicity naturally comes with some limitations in modelling more complex problems, which however, may be modelled with higher order HMMs. Futhermore, the three assumptions are very similar to those of a Markov chain. This is because the stochastic process of an HMM pertaining to the hidden states can be reduced to a Markov chain. In fact, an HMM is an extension of a Markov Chain. The essential difference between the two is that, with the former, there is no a one-to-one mapping between the states and the observation symbols.

2.3.2 The three basics problems for HMM design

In , Lawrence argued that an HMM design needs to answer three fondamental problems. They are the *training problem*, the *evaluation problem*, and the *decoding problem*. Each problem and its solution is discuss in greater details next.

The evaluation problem

The evaluation problem is about answering this question: *Given the observation sequence $O = O_1O_2O_T$, and a model λ, how do we efficiently compute $P(O|\lambda$, the probability of the observation sequence?* The naive answer to this question is simply computing the $P(O|\lambda)$ according to equation 2.5:

$$P(O|\lambda = \sum_{q_1}^{q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1q_2} b_{q_2}(O_2)...a_{q_{T-1}q_T}(O)) \tag{2.5}$$

This approach has two issues, it is not only, computationally too expensive because of the exponential complexity with respect to T, but also, intractable for very long sequence. In pactice, $P(O|\lambda)$ is computed by an algorithm called *forward-backward* procedure, which is a more efficient method.

The decoding problem

The decoding problem can be reduced to this interrogation: *Given the observation sequence $O = O_1O_2O_T$, and the model λ, how do we choose a corresponding state sequence $Q = q_1q_2...Q_T$ which is optimal in some meaningful sense i.e, best "explains" the observations?* Simply put, this problem is about deciphering the most likely hidden states that emitted the visible observation sequences. This is done dynamically using the Viterbi algorithm.

The training problem

Given the model, λ, the training problem raises the following question: *how do we adjust the model parameters λ to maximise the $P(O|\lambda)$, the probability of the probability of the observation sequence?* This problem is usually solved by iterative learning algorithms

called expectation-maximisation. Examples of this algorithms are Baum-Welch method or any gradient based method.
When using Baum-Welch algorithm, the parameters are initialised by guesses then re-estimated iteratively to find the parameters with maximum likelihood. This method is vulnerable to local maxima issues. To avoid such cases, it is advice to run it multiple times with different initial values in order to keep the estimation with the highest likelihood value..

First order Markov model - current and predecessor only considered

2.4 k-Nearest Neighbour

2.5 Dimensionality reduction

2.5.1 Motivations for dimensionality reduction

2.5.2 Filter methods

Forward feature subset selection

Similarity index

2.5.3 Wrapper methods

Principal component analysis: PCA

Linear discriminant analysis: LDA

2.5.4 Hybrid filter-wrapper methods

2.6 Sufficiency of Training Data

2.7 Techniques to increase Training Data

2.7.1 Mirroring

Chapter 3

Hidden Markov Model design

This section focuses on the design of the HMM used to test the hyphotheses postulated above.

3.1 Description of available dataset

The available dataset was acquired from a moving dog using Inertia Measurement Units. Two inertial measurements units (IMU) were straped to the front and back of a dog. Each unit has an accelarometer, a gyroscope and a magnetometer. The dataset contains calibrated measurements of a dog running, walking, and trotting then walking; together with the footfalls. The footfall is represented by a binary value that indicates the state of the dog's leg: if it is on or above the ground, at a particular instant in its gait sequence. More specifically, the value 0 means leg up and the value 1 means leg down. The four variables representing the footfalls effectively constitute the ground truth, informing us about the state in which the dog is, at a given time in its movement.

The dataset can be retrieved from nine different matlab files. Each file contains twenty four matlab variables. The variables of interest are listed in the table 3.1.

The observations are continous and the statistical property are assumed to be stationary, i.e, the do not vary over time. In this sta

Observations			
Body part	Accelerometer	Gyroscope	Magnetometer
Front	accFrontX accFrontY accFrontZ	FrontPitch FrontRoll FrontYaw	magFront_cal magFront_cal2 magFront_cal3
— Back	accBackX accBackY accBackZ	BackPitch BackRoll BackYaw	magBack_cal magBack_cal2 magBack_cal3

Table 3.1: IMU measurements and footfall variables in dataset

stationary: statistical property do not vary over time or non-stationary: properties vary over time

pure or corrupted?

3.1.1 Quadrupede Gait sequence modelling

One of the objectives of this project is to effectively model the gait sequence dynamic of the dog from IMU measurements using HMM. Based on the fact that quadrupedes achieve inverted pendulum-like movements like humans, their gait dynamic can be modelled as a succession of latent states observed by measurements such as IMU data. The states representing the footfalls and the observations, the outputs of the accelerometer, gyroscope and the magnetometer. Similar to human gait mechanism, it is sound to assume that the current state of a quadrupede is conditionally dependent on its previous state. This inference combined with the statistical robustness of HMM makes it the best model candidate when the available dataset is not large enough.

HMM model elements: states and observations properties

The problem at hand requires 16 distinct state that make up the state vector S, shown in equation 3.1

$$S = S_i = \{(LF, RF, LB, RB)\} = \{0000, 0001, 0010, ..., 1111\}. \tag{3.1}$$

$$|S| = N = 2^4 = 16$$

$$i = 1, 2, ..., 16$$

The 16 distinct states are derived from the combination of the four binary footfalls. In practice, the dataset may not reveal all the 16 states.

The stream of IMU measurement form the observation sequence. An observation instance is a row vector of K dimensions. The initial K value before any dimensionality reduction is 18, from the 18 IMU measurements. Thus, an observation sequence O is a TxK matrix of continuous values as presented in 3.2. T is the total number of the successive measurements.

$$O = \{Ok_t\} = O1_t, O2_t, ..., O18_t. \tag{3.2}$$

$$k = 1, 2, ..., 18. \tag{3.3}$$

$$t = 1, 2, ..., T. \tag{3.4}$$

Splitting the 16-states HMM in two 4-states HHMs

In order to simplify the problem, it was decided to split the the four legs in two subparts: two front legs and two back legs. This approach for spliting it was based on the two inverted-like movements for a quadrupede such as a dog has investigated in As a result, the initial 16-states HMM becomes, two distinct 4-states HMMs. These two models may be combined to get back the holistic 16-states model. This design decision was motivated by the fact that it is a simpler task to dsicriminate between 4 distinct classses than 16 classes. From here onward, attention will be given to the 4-states HMM model.

Transition between states

This design assumes that a dog can transition from one state to any other possible state. So, for any transtion from S_i to S_j both S_i and S_j may be any of the element of the state space

$$S = \{S1, S2, S3, S4\}$$

.

For instance, if a dog has its left leg above ground and its right leg on ground, at time instance t, it may move to any of the 3 other possible positions or remain in the same state, in the next time instance, t + 1. This consideration yielded in an ergodic HMM where, all the transitions are possible. The graphical model of the simplified HMM is illustrated by figure 3.1

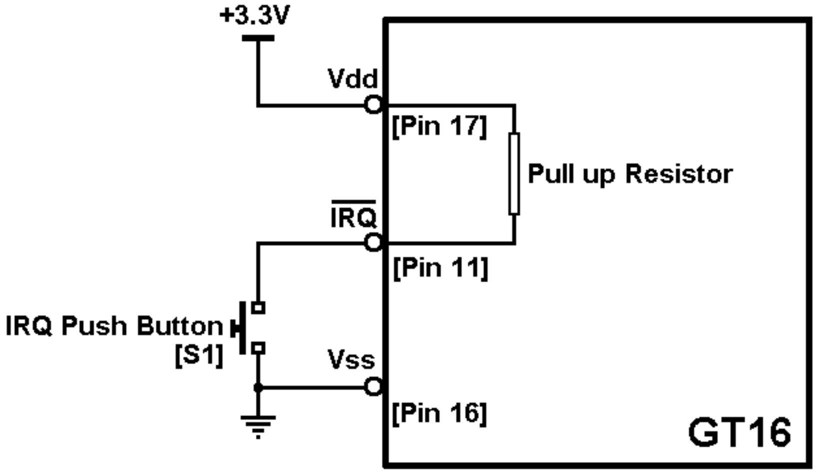


Figure 3.1: Ergodic HMM graphical model showing the hidden states, observation, and transitions between states

3.1.2 Data pre-processing

3.1.3 Model parameters estimation

As a reminder, a continuous HMM model is completely specified by its initial state distribution: π transition matrix: A; the mean covariance matrices: μ, Σ which can be combined into Φ. If the observations are modelled with gaussian mixture distributions, one addition parameter is required for the initial mixture distribution: β. The next sub-sections discuss how each parameters was estimated in this project.

Transition matrix: A

For each of the front and back 4-state HMM, the state transition matrix A, is a 4-by-4 matrix. Two different approaches were considered in the estimation of A. The two methods make use of the expectation maximisation algorithm but differ in the input arguments considered.

1. Approach 1: Exploiting the available ground truth
   This approach takes advantage of the ground truth for a labelled dataset to reduce

the HMM model to a Markov Chain. This is done by making the hypothetical observation sequence identical to the state sequence. Then, using approach used in discrete HMM, the transition matrix can be estimated using maximum likelihood algorithm. For each state, the pseudocount was set to the number of occurences of that state in the training data plus a constant value: $PseudoA = |S_i| + C$. The additional constant C is to avoid having 0 in the transition matrix for states and transitions not reflected in the dataset. It can be chosen empirically. This method is very simple however, it has two limitations. It not only assumes that the number of states is known but also requires the training data to be labelled. Although, the dataset at hand satisfies the two constraints, the second approach which eliminates these possible setbacks, was also considered.

2. Approach 2: This method is the standard approach found in literature using expectation algorithm such as Welch-Baum algorithm. and described in the literature review.

Mean and covariance matrices: μ, Σ and β

The observations were modelled using mixture Gaussian distributions, characterised by a KxMxN mean matrix: μ and a KxKxMxN co-variance matrix: Σ. where: $K = number \quad of \quad features, \quad M = mixture \quad number, \quad and \quad N = number \quad of \quad states.$ The observations were grouped K classes following based on the ground-truth represented by the state sequence. The each group of observation was used as input to the EM algorithm to estimate the μ and Σ. In general, the optimal number of mixture, M is chosen empirically, in this project, it was estimated using akaike information criterion (AIC). So, gaussian mixture models were built using EM algorithm while varying the M from 1 to K, the feature number. Since AIC is a measure of information loss, the model with the minimum AIC best represents the dataset. The number of mixture M, is therefore set to the mixture number of this model. This algorithm is outlined below.

```
function M = best_M(data, K)
  data = training_set
  AIC = zeros(1, K);
  models = cell(1, K);
  for m = 1:K
    model = gauss_mixture(data, m);
    AIC(m) = model.AIC;
  end
```

```
  [minAIC, minAIC_Idx] = min(AIC);
  M = minAIC_Idx;
end
```

The mixture components were considered evenly distributed initially. In order to avoid zero values in the covariance matrix, it was regularized with $10^{-}10$. The maximum number of iteration of the EM algorithm was empirically set to 1000.

Inital state distribution: π

The initial state distribution was estimated using the probability of occurence of each state in the training data sample. Thus, $\pi = \{\pi_i\} = \frac{|S_i|}{T}$

3.1.4 Dimension reduction

Optimal number of features

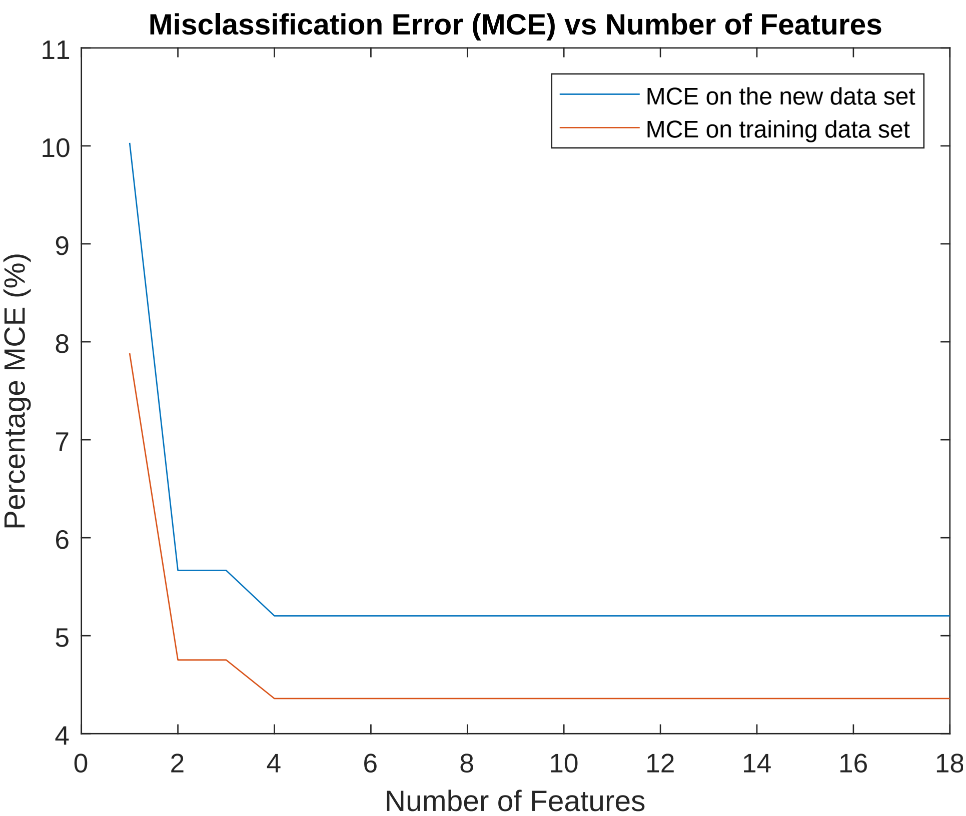


Figure 3.2: Misclassification error vs feature number using separability index and KNN

Optimal PCA component number

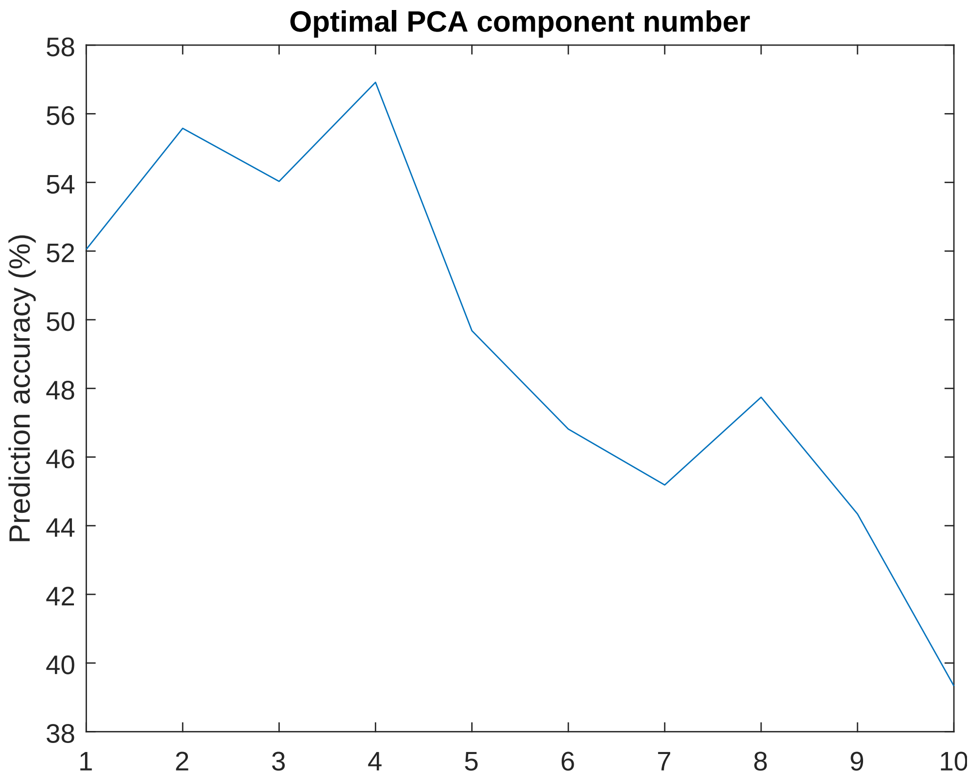


Figure 3.3: Optimal PCA component number

3.2 HMM implementation

Chapter 4

Results

4.1 Experiement 1: The effect of a CHMM' observation dimensionality on its performance

4.1.1 Aim of the experiement

The aim of this experiement is to investigate how the number of features impacts the accuracy of a Hidden Markov Model with continuous emission symbols (CHMM), in the abscence of enough training data. Thus, the hypothesis under investigation is:
In the absence of enough training data, a CHMM with observations of high dimensionality performs poorly.

4.1.2 Experiment apparatus

To perform this expereiment, the following materials are required:

- λ, a continuous Hidden Markov Model specified by $\lambda = (A, \beta_{jm}, \mu_{jm}, \Sigma_{jm}, \pi)$.
- At least two sample data sets training the model and testing it.
- A criterion to rank and select subsets of features.
- A measure to evaluate the performance of the CHMM model.

- Finally, a way to visualise the results of the experiments

4.1.3 Experiment procedure

The expreriment was performed with the steps listed below:

1. Step 0 - Preliminary data pre-processing: This step consisted in the data pre-processing as described in
2. Step 1 - Partitioning data into training and test sets: Here, the dataset was randomly sampled into training and test sets. The training set was relatively small, it was a sequence 539 observations.
3. Step 2 - Feature ranking: The features were sorted in a descending order based on their ability to discriminate the different states of the CHMM. The separability index method described in was used for this purpose.
4. Step 3 - Data subset selection: Select the optimal feature subset, starting with 1 dimension.
5. Step 4 - Model building and training: The CHMM model, λ was built and trained using with training dataset using the optimal feature subset.
6. Step 5 - Model testing and evaluation: The model was tested with the test dataset. The test consisted in decoding the most likely state sequence given a previously unseen sequence of observations. This path prediction was evaluated based on the evaluation criterion presented in
7. Step 6 - Iteration: Step 3 through step 5 were repeated while varying the feature subset size until the maximum size, which is 18 in this case. In each iteration, the prediction accuracy was stored in an array for visualisation.
8. The different accuracies were finally ploted as a function of the observation dimensionality. Moreover, the observations were grouped based on the corresponding hidden state sequence and scattered in a 2-dimensional principal component space. This is to compare the decoded states against the ground-truth.

4.1.4 Experiment results

The results of the experiments are presented in figure 4.1, 4.2, 4.3, 4.4, 4.5, 4.6.
4.1 and 4.2 show how the hidden state decoding performance and the log-likelihood estimated by the CHMM model, train with just 539 observations, varies as the observation dimensionality increases.
4.3 and 4.4, 4.5 and 4.6 illustrate how the estimated state sequences compare to reality for observation sequences of 5-dimensions and 18-dimensions. 5 and 18 dimensions were presented because they are the two extremes in terms of accuracy. Other dimensions may be found in the appendices,

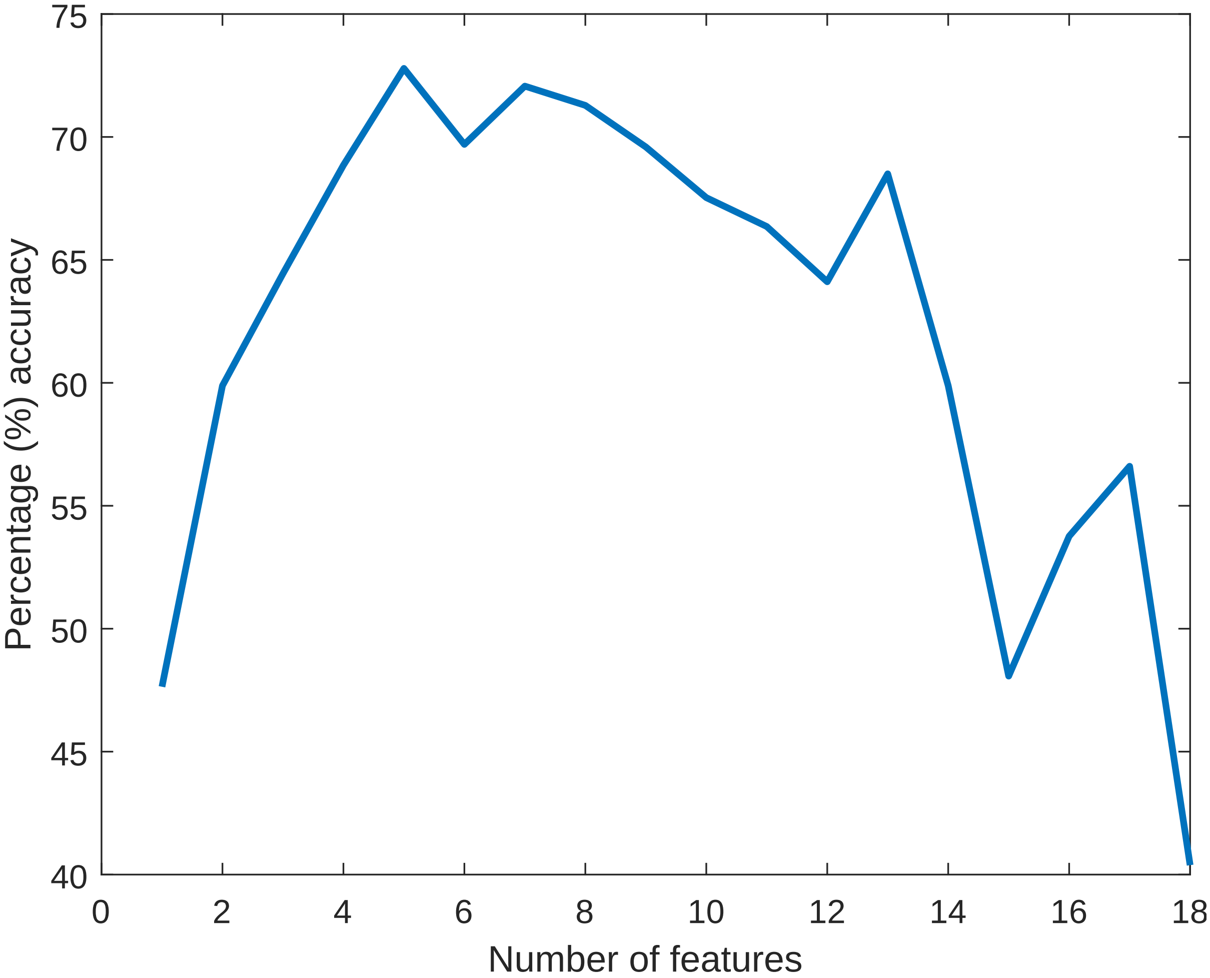


Figure 4.1: The effect of CHMM's observation dimensionality the state sequence decoding accuracy

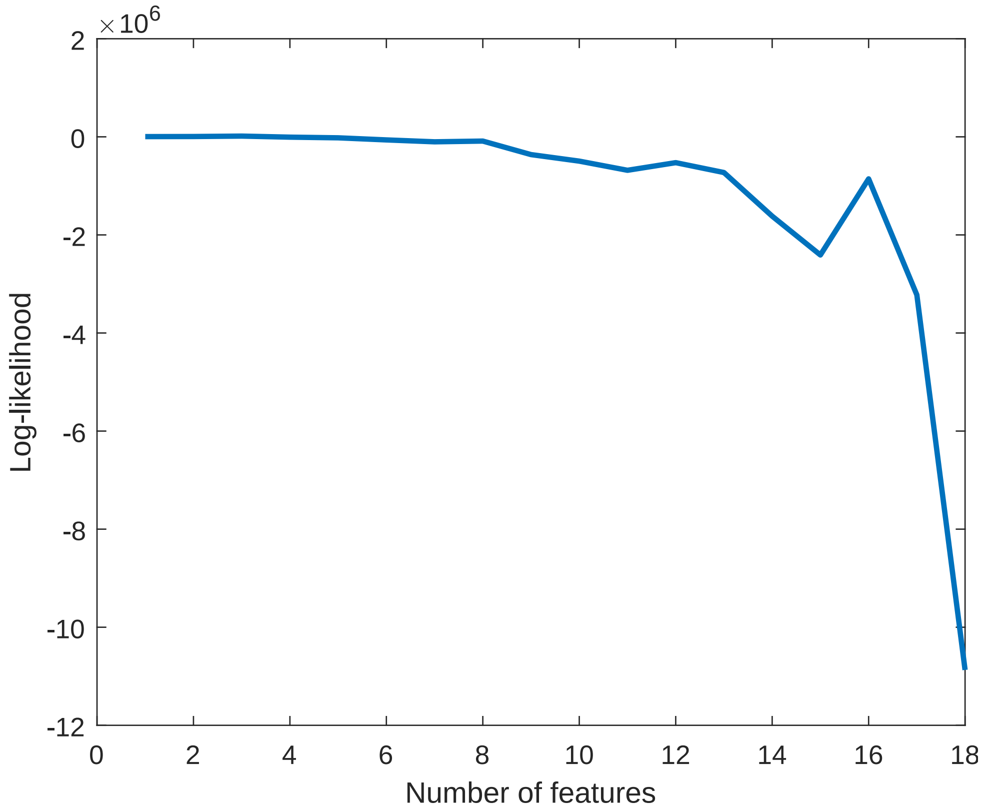


Figure 4.2: The effect of CHMM's observation dimensionality the log-likelihood

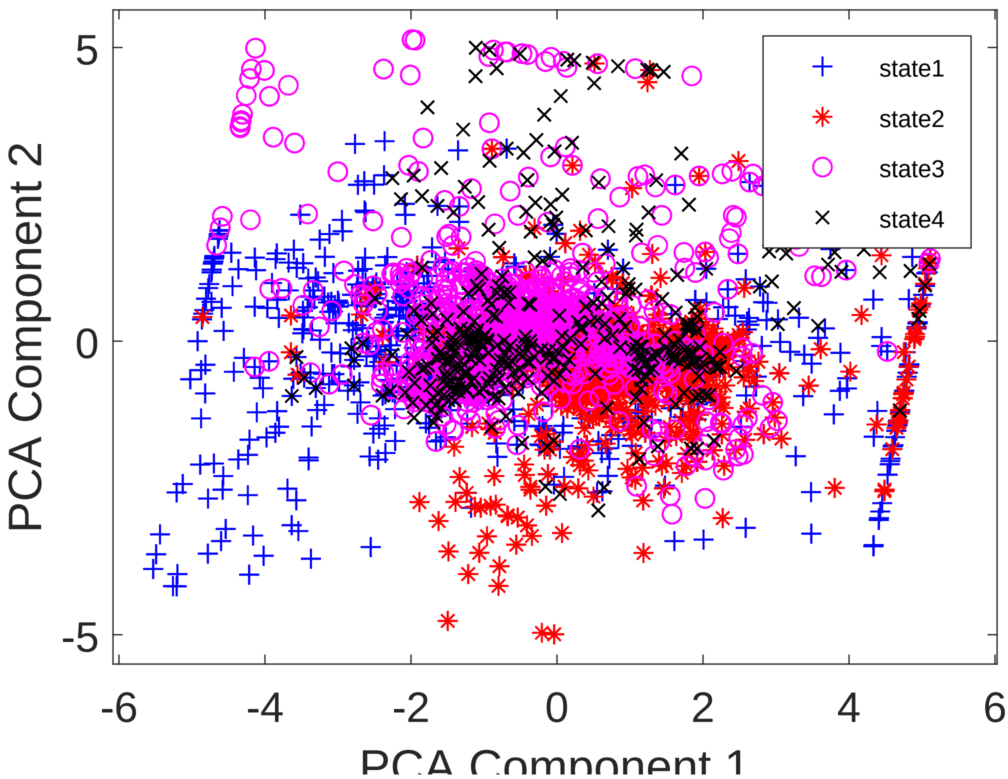


Figure 4.3: Scatter plot of 5-dimensional observations grouped per state based on ground-truth state sequence

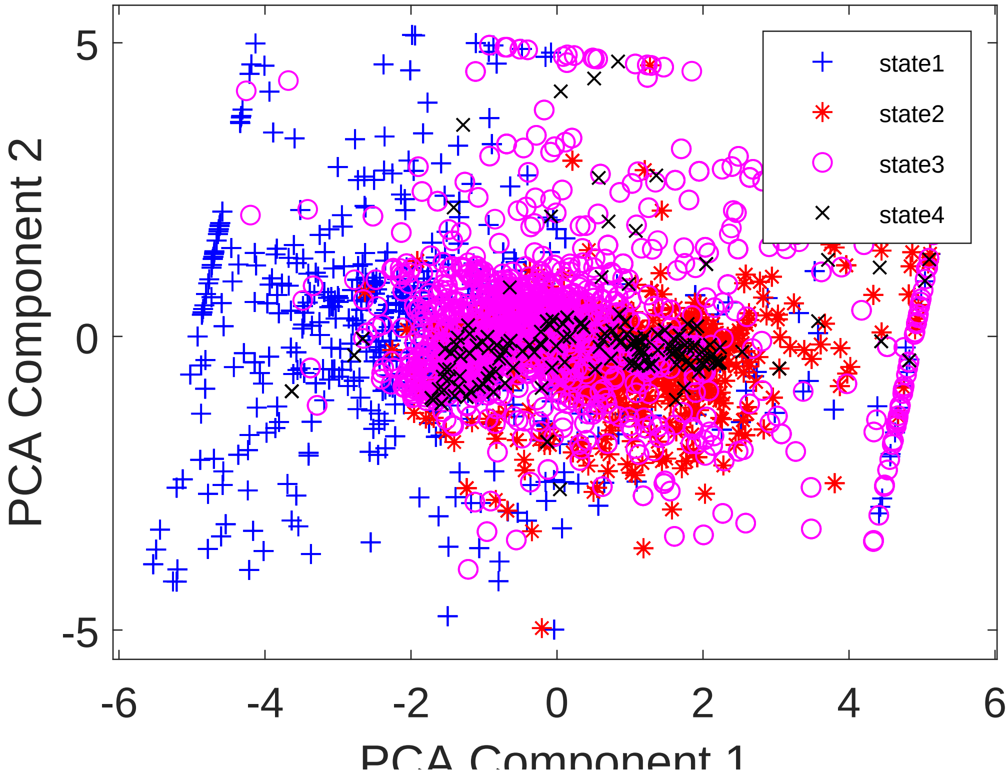


Figure 4.4: Scatter plot of 5-dimensional observations grouped per state based on estimated state sequence

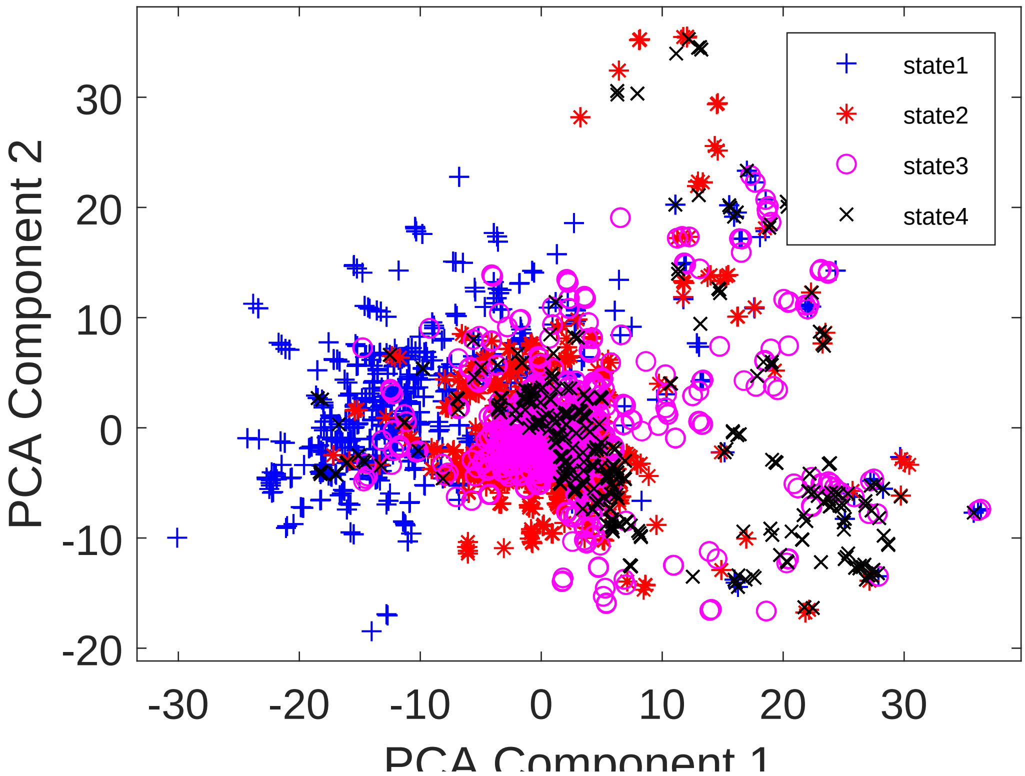


Figure 4.5: Scatter plot of 18-dimensional observations grouped per state based on ground-truth state sequence

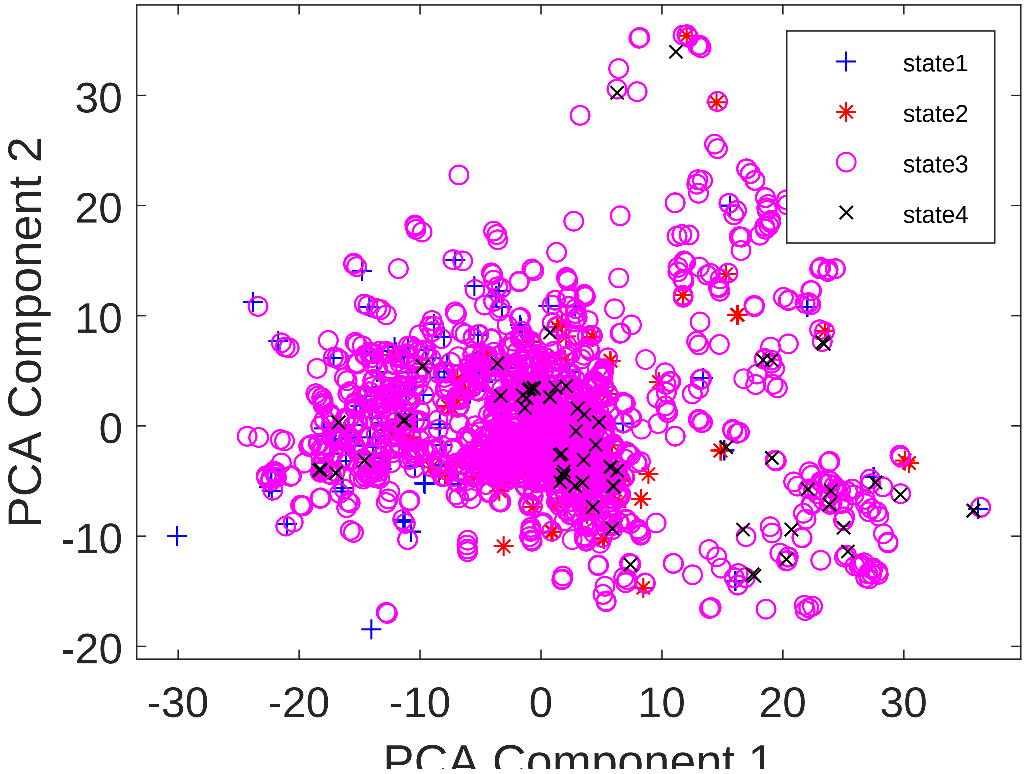


Figure 4.6: Scatter plot of 18-dimensional observations grouped per state based on estimated state sequence

4.1.5 Analysis of results

4.1.6 Conclusions and recommendations of the experiment

4.2 Experiement 2: The effect of dimensionality reduction on CHMM's performance

4.2.1 Aim of the experiement

The aim of this experiment is to investigate the effect of dimensionality reduction on the performance of a continuous Hidden Markov Model (CHMM). The hypothesis under investigation is therefore the following: ***Dimension reduction can cause an increase in a CHMM model performance when there is not enough training data.*** Thus, the performance of the CHMM without and with various dimensionality reduction are compared to test the hypothesis.

4.2.2 Experiment apparatus

The assets needed to perform the experiment are listed below.

- λ, a continuous Hidden Markov Model specified by $\lambda = (A, \beta_{jm}, \mu_{jm}, \Sigma_{jm}, \pi)$.
- Dimensionality reduction methods. Two wrapper and two filter methods were considered. The wrapper methods were Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA). The two filters methods were feature ranking with similarity index and a combination of forward feature selection and similarity index.
- A Performance metric. The metric used were the hidden state decoding accuracy and the log-likelihood of the EM algorithm.

4.2.3 Experiment procedure

The experiment was performed as follows. First, the dataset was partition into two different set for training and testing using random sampling. Using the same traing dataset five different models were built and trained, $\Lambda = (\lambda_{NoReduction}, \lambda_{PCA}, \lambda_{LDA}, \lambda_{SI}, \lambda_{SI-forward})$. Then the different models were tested with the same test dataset and the prediction accuracy as well as the EM algorithm loglikehood were recorded. The training and the testing were repeated while varying the proportion of training set used from 10% to 90% of the total dataset. The prediction accuracies and the EM algorithm likelihoods were finally plotted as a function of the training data size for each model. These findings are presented in the figures 4.7 and 4.8.

4.2.4 Experiment results

Firstly, figure 4.7 how the performances of the five CHMMs compare against each other as the training data size increases. Secondly, the loglikelihoods presented in 4.8 show how effectively can each model recognise an observation sequence generated by the underlying mechanism.

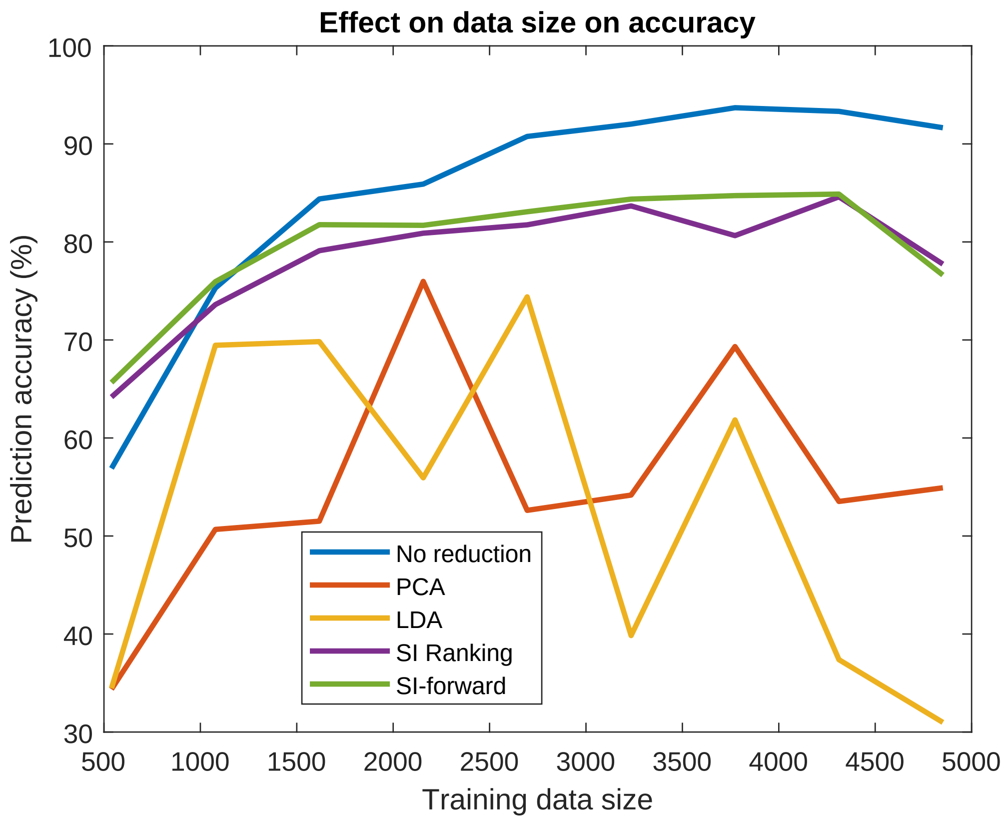


Figure 4.7: The effect of training datasize on the prediction accuracy

4.2.5 Analysis of results

4.2.6 Conclusions and recommendations of the experiment

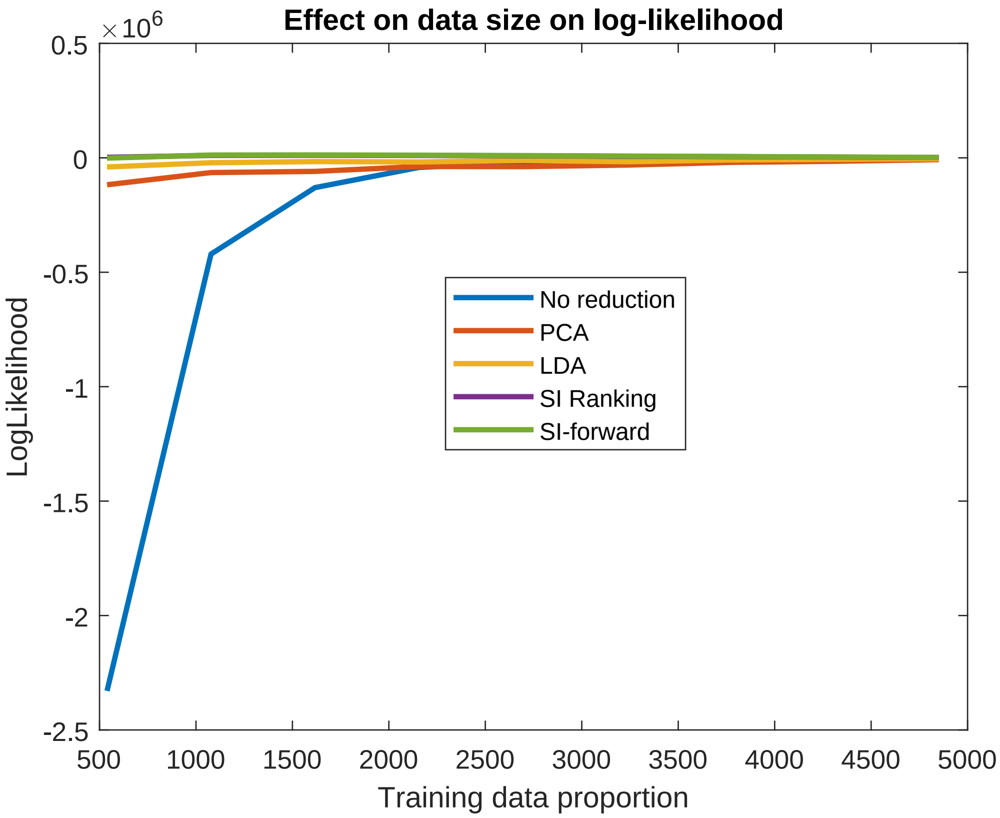


Figure 4.8: The effect of training datasize on the log likelihood

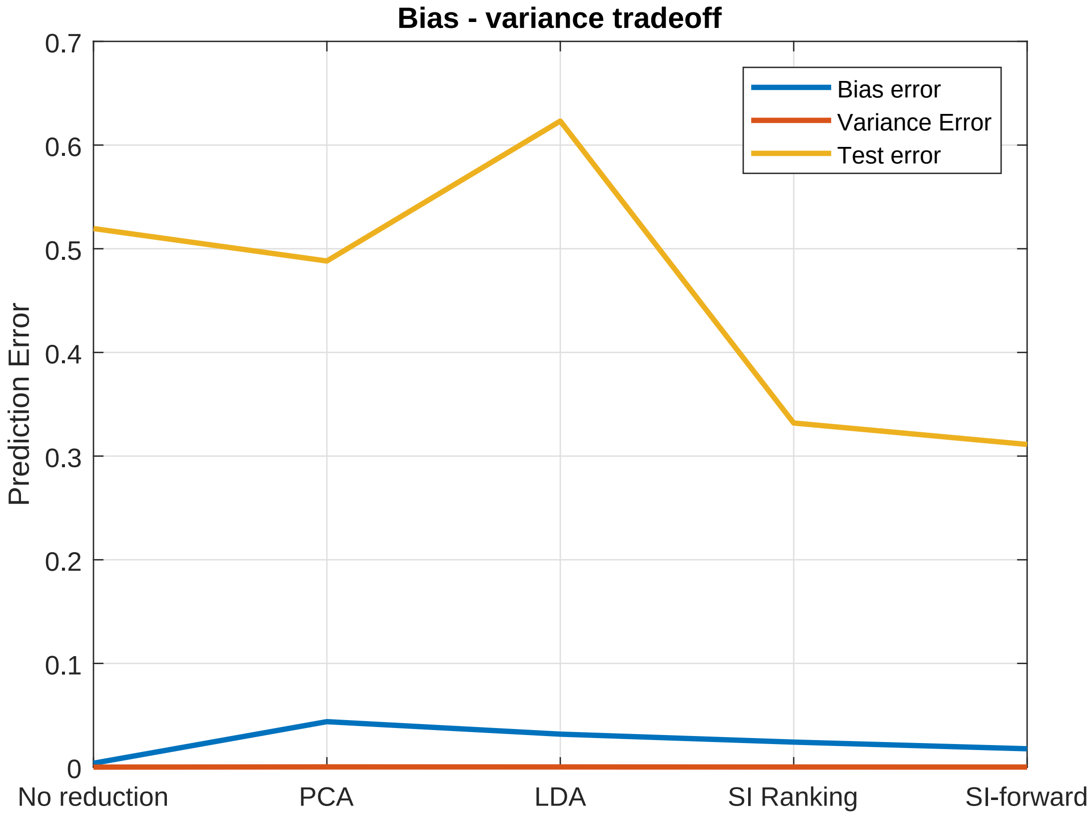


Figure 4.9: Bias-Variance tradeoff analysis

Chapter 5

Discussion

Here is what the results mean and how they tie to existing literature...

Discuss the relevance of your results and how they fit into the theoretical work you described in your literature review.

Chapter 6

Conclusions

These are the conclusions from the investivation and how the investigation changes things in this field or contributes to current knowledge...

Draw suitable and intelligent conclusions from your results and subsequent discussion.

Chapter 7

Recommendations

Make sensible recommendations for further work.

Use the IEEE numbered reference style for referencing your work as shown in your thesis guidelines. Please remember that the majority of your referenced work should be from journal articles, technical reports and books not online sources such as Wikipedia.

Bibliography

[1] M. S. Tsoeu and M. Braae, "Control Systems," *IEEE*, **vol. 34(3)**, pp. 123-129, 2011.

[2] J. C. Tapson, *Instrumentation*, UCT Press, Cape Town, 2010.

Appendix A

Additional Files and Schematics

Add any information here that you would like to have in your project but is not necessary in the main text. Remember to refer to it in the main text. Separate your appendices based on what they are for example. Equation derivations in Appendix A and code in Appendix B etc.

Appendix B

Addenda

B.1 Ethics Forms