IMPORTANT ISSUES FOR BRAIN CONNECTIVITY MODELLING BY DISCRETE DYNAMIC BAYESIAN NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SALİH GEDUK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONIC ENGINEERING

JANUARY 2020

Approval of the thesis:

THESIS TITLE

submitted by **SALİH GEDUK** in partial fulfillment of the requirements for the degree of **Master of Science** in **Electrical and Electronic Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of the Department, **Electrical and Electronics Eng.** _____

Prof. Dr. İlkay Ulusoy
Supervisor, **Electrical and Electronics Eng., METU** _____

Examining Committee Members:

Prof. Dr. Uğur Halıcı
Electrical and Electronics Eng., METU _____

Prof. Dr. İlkay Ulusoy
Electrical and Electronics Eng., METU _____

Prof. Dr. Metehan Çiçek
Medicine, Ankara University _____

Prof. Dr. Umut Orguner
Electrical and Electronics Eng., METU _____

Assoc. Prof. Dr. Yeşim Serinağaoğlu
Electrical and Electronics Eng., METU _____

Date: 30.01.2020

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Salih Geduk

Signature :

ABSTRACT

IMPORTANT ISSUES FOR BRAIN CONNECTIVITY MODELLING BY DISCRETE DYNAMIC BAYESIAN NETWORKS

Geduk, Salih
Master of Science, Electrical and Electronic Engineering
Supervisor: Prof. Dr. İlkay Ulusoy

January 2020, 125 pages

To understand the underlying neural mechanisms in the brain, effective connectivity among brain regions is important. Discrete Dynamic Bayesian Networks (dDBN) have been proposed to model the brain's effective connectivity, due to its nonlinear and probabilistic nature. In modeling brain connectivity using discrete dynamic Bayesian network (dDBN), we need to make sure that the model accurately reflects the internal brain structure in spite of limited neuroimaging data. Based on the fact that there are many dDBN structure learning applications in the recent literature and most of them use very limited amount of data, some facts should be made clear at least for the model convergence which depends on the number of data, the model complexity, and the learning approach. In this thesis, we analyzed the sample complexity of dDBN to find the required number of samples that guarantee successful learning. Firstly, we realized that the theoretical sample complexity for dDBN structure learning is not realistic, practical and applicable in practice. Therefore, we also focused on a practical and systematic approach for estimating the sample complexity for dDBN. Secondly, we evaluated the non-supervised discretization methods for functional magnetic resonance imaging (fMRI) data

which has not been done yet to the best of our knowledge. We generated synthetic fMRI data that possess temporal relations. Then they were used for modeling effective connectivity by dDBN to compare the performance of each discretization method. Thirdly we analyzed the smoothing step of the fMRI data which is necessary to improve the signal to noise ratio. Experiments suggested that smoothing fMRI data with Gaussian function having a standard deviation to be 4 mm is suitable considering effective connectivity via dDBN. Lastly, by considering these results we used dDBN to model the brain connectivity of schizophrenia and control group. The results signify that schizophrenia is a disconnection syndrome.

Keywords: Discrete Dynamic Bayesian Networks, fMRI, Structure Learning, Sample Complexity, Effective Connectivity, Schizophrenia

ÖZ

DİNAMİK BAYESÇİ AĞI İLE YAPILAN BEYİN BAĞLANTILARI İÇİN ÖNEMLİ HUSUSLAR

Geduk, Salih
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği
Tez Yöneticisi: Prof. Dr. İlkay Ulusoy

Ocak 2020, 125 sayfa

Beyindeki altta yatan sinirsel mekanizmaları anlamak için beyin bölgeleri arasındaki etkin bağlantısallığı göz önünde bulundurmak önemlidir. Ayrık Dinamik Bayes Ağları (dDBN), doğrusal olmayan ve olasılıklı doğası nedeniyle beynin etkin bağlantısallığını modellemek için önerilmiştir. Ayrık dinamik Bayes ağını (dDBN) kullanarak beyin bağlantısallığını modellerken, modelin sınırlı beyin görüntüleme verilerine rağmen dahili beyin yapısını doğru bir şekilde yansıttığından emin olmalıyız. Literatürde çok sayıda dDBN yapısı öğrenme uygulamasının bulunmasına ve çoğunun çok sınırlı miktarda veri kullanmasına bağlı olarak, en azından veri sayısına, model karmaşıklığına ve öğrenme yaklaşımına bağlı olan model yakınsaması için bazı gerçekler açıkça belirtilmelidir. Bu tezde, başarılı bir öğrenmeyi garanti eden gerekli sayıda örneği bulmak için dDBN'nin örnek karmaşıklığını analiz ettik. İlk olarak, dDBN yapı öğrenmesi için teorik örnek karmaşıklığını bulduk. Bununla birlikte, teorik örneklem miktarı gerçekçi, pratik ve dDBN için geçerli değildir. Bu nedenle, dDBN için örnek karmaşıklığını analiz etmek için pratik ve sistematik bir yaklaşıma odaklandık. Ayrıca, bilgimiz dahilinde henüz yapılmayan fMRI verileri için denetimsiz ayrıklaştırma yöntemlerini de

değerlendirdik. Zamansal ilişkilere sahip sentetik fMRI verileri oluşturduk. Daha sonra, bu veri her bir ayrıklaştırma yönteminin performansını karşılaştırmak için dDBN tarafından etkin bağlantısallığı modellemek için kullanıldı. Üçüncü olarak, sinyal-gürültü oranını iyileştirmek için gerekli olan fMRI verilerinin yumuşatma aşamasını analiz ettik. Deneyler, standart sapması 4 mm olan Gauss fonksiyonu ile fMRI verilerinin yumuşatılmasının, dDBN ile yapılan etkin bağlantısallık göz önüne alındığında uygun olduğunu göstermiştir. Son olarak, bu sonuçları dikkate alarak, şizofreni ve kontrol grubunda beyin bağlantısını modellemek için dDBN kullandık. Sonuçlar, şizofreninin bir kopukluk sendromu olduğunu göstermiştir.

Anahtar Kelimeler: Ayrıkçı Dinamik Bayes Ağları, fMRI, Yapı Öğrenimi, Örnek Karmaşıklığı, Etkin Bağlantısallık, Şizofreni

To my dear family…

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere appreciation and gratitude to my supervisor Prof. Dr. İlkay Ulusoy for her continuous support, criticism and invaluable guidance throughout my thesis study.

I would like to express my gratitude to Prof. Dr. Metehan Çiçek, Dr. Hikmet Emre Kale and Dr. Sertaç Üstün for their advices and valuable information that enable me to learn about functional Magnetic Resonance Imaging data.

For their comments and criticism, I would also like to thank the examining committee members; Prof. Dr. Uğur Halıcı, Prof. Dr. Umut Orguner, and Assoc. Prof. Dr. Yeşim Serinağaoğlu.

I am thankful to TÜBİTAK, the National Scientific and Technological Research Council of Turkey, for granting me their M.S. studies scholarship.

Finally, but forever I owe my loving thanks to my family, my mother Saadet, my father Kadri, my brother Mustafa and my sisters Kerime, Esengül, Gülşen and Yeter for their love, support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

BN	Bayesian Network
DBN	Dynamic Bayesian Network
dDBN	Discrete Dynamic Bayesian Network
fMRI	Functional Magnetic Resonance Imaging
BOLD	Blood Oxygenation Level-Dependent
DCM	Dynamic Causal Modelling
EEG	Electroencephalogram
BDeu	Bayesian Dirichlet equivalence with a uniform prior
BIC	Bayesian Information Criteria
GED	Gene Expression Data
TDT	Target Discretization Threshold
TSD	Transitional State Discretization
EWD	Equal Width Discretization
EFD	Equal Frequency Discretization
HRF	Hemodynamic Response Function
VAR	Vector Autoregressive
ROI	Region of Interest

CHAPTER 1

INTRODUCTION

1.1 Motivation

By functional magnetic resonance imaging, brain regions involved in various cognitive tasks can be detected [1]. Considering multiple processes that occur in brain regions that interact with each other, extracting brain connectivity from fMRI data during a specific task can help us to understand brain functioning. In fMRI, brain activity is measured by time-series signals based on blood oxygenation level-dependent (BOLD) contrast. One of the most important connectivity approaches using these time series is undoubtedly effective connectivity [2]. Effective connectivity reveals the causal interactions between brain regions. Dynamic Bayesian Networks are appropriate to model the brain's effective connectivity due to their non-deterministic behavior. Due to the complexity of modeling, two DBN methods are applicable, one is Gaussian DBN [3], where brain regions are modeled with linear gaussian relations, second one is discrete DBN (dDBN), where non-linear modeling is possible by discretizing the data and using multinomial distributions over the network parameters [4]–[6].

A dDBN is specified by two components: a structure (graph or model), which represents the conditional independencies between random variables and parameters, which represent the conditional probability distributions among these random variables. When we model the brain with dDBN, the nodes correspond to the brain regions, and the structure refers to the effective connectivity of the brain.

A series of steps are followed to model the effective connectivity of the brain using dDBN. Figure 1-1 gives each step for modeling the effective connectivity of the brain

by dDBN. The first step is preprocessing where smoothing is applied to the raw fMRI data to get rid of errors due to scanning procedure. Secondly, the time-series of the identified brain regions (i.e., the region of interests, ROIs) are obtained. This determination is either done by expert knowledge or by generalized linear model technique to find activated regions in the brain. Then 4-dimensional fMRI data is transformed into 1-Dimensional data for each region. Then, these time series are discretized, where data is converted to a finite number of discrete values. Finally, the discrete time-series are used as input to the dDBN learning procedure and the brain effective connectivity is modeled.

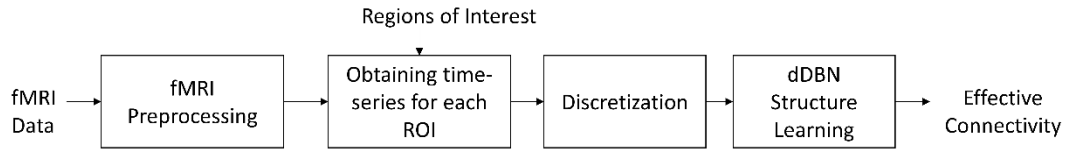

Figure 1-1: Steps to obtain effective connectivity by dDBN

In this thesis, three important steps are investigated and clarified to make sure that the modeling is done correctly. The first one is the required number of data samples to be able to model the data by the dDBN learning procedure. Although motivation was for fMRI data, the required sample size is in general related to model not the type of data used. Therefore, assessment of the number of samples is investigated independently from fMRI data. The second one is the discretization method to convert fMRI data into a discrete set of states. Such a discretization has a significant impact on the correctness of modeling. The information in the continuous values should be kept while the number of quantized states should be as low as possible for the computational complexity of the learning procedure. Although some discretization methods are used in dDBN connectivity studies with fMRI data in the literature [4]–[6], we have not yet found any studies which search the best discretization method for this purpose. The last issue is about the smoothing process which is done in the fMRI preprocessing step. Smoothing is the only option for fMRI data to improve the signal to noise ratio. However excessive smoothing may cause to lose the spatial information that fMRI data possess. In this thesis, we considered

an approach considering the properties of dDBN and time-series to find a suitable smoothing.

In most of the effective connectivity studies, researchers did not consider the issues discussed in this thesis. However, they are very crucial for the correctness of the modeling.

1.2 Contributions

In this thesis, the most important and greatest contribution is that all the steps of the dDBN method for brain modeling are examined and every important issue is solved. Consequently, considering the studies of this thesis, the brain effective connectivity can be done by using fMRI data and the hidden sides of the brain can be illuminated.

This thesis contains various contributions in terms of the sample complexity of dDBN. Although our motivation relies on fMRI data, sample complexity analysis is in general related to discrete Dynamic Bayesian Networks. First of all, as far as we know this study is the first for putting forward the sample complexity for dDBN on model discovery. A practical approach rather than a theoretical one was applied to see the effect of sample size for learning the structure of dDBN. Theoretical results are not applicable for real applications, such as the effective connectivity of the brain by fMRI, because they state the need for immense sample size in order to find the structure correctly. However, our practical results state that less number of samples is enough to discover the correct structure. In addition, experimental results are used to practically assess the sample complexity with respect to network parameters. $O(K^{p+1})$ is found to be the sample complexity for binary and ternary valued dDBNs, where K indicates the cardinality of the network, and p indicates the maximum number of parents in the network. Another contribution is to evaluate how the BDeu score is affected by the number of samples and what kind of structures are learned as a result of the dDBN learning procedure considering the imaginary sample size used as a prior belief for BDeu metric. This contribution to the literature will enable researchers to use dDBN more accurately in their studies. dDBN is used in various

areas such as economics, bioinformatics and neuroscience. This contribution will advance the studies to use the dDBN technique in modeling.

We had two main objectives for the discretization case. The first was to evaluate the state-of-the-art non-supervised discretization methods to model the effective connectivity of the brain with dDBN using fMRI data and to determine the best among them. The discretization methods used in this study are mostly explained in [7]–[9]. Our second aim was to use variation between successive time points in the discretization and to show that using the differential information performs better. To achieve these goals, first of all, we produced synthetic fMRI data from 1000 different connectivity models. Then, we discretized this data with all of the discretization methods and used them in dDBN learning. We compared the ground-truth models and the learned models with appropriate error metrics. It was observed that the use of the derivative rather than the fMRI data itself was more informative in dDBN modeling. Moreover, we tested discretization techniques using real fMRI data, and similar results were obtained with synthetic data. Discretization methods was only evaluated for fMRI data in this study. Hence any area using this technique should also consider and evaluate discretization methods for the data they use. The discretization technique is rather data-dependent, modelling method only can be used as an evaluation metric to find the best discretization method.

Another contribution is related to the smoothing step of fMRI preprocessing. The results of the discretization step suggested that scanner noise has a negative effect on the discretization. Spatial smoothing is the only issue that enhances the signal to noise ratio. To do that fMRI data was smoothed for various sigma values of Gaussian filter. Then the resulted dDBN models were analyzed and it was found that for smoothing, sigma should between 4-7 mm, and 4 mm found to be more promising.

1.3 The Outline of the Thesis

This thesis contains four main studies. The first one is the sample complexity analysis of discrete dynamic Bayesian networks. The second one is the evaluation of discretization methods for fMRI data. The third one is to find suitable spatial

smoothing for fMRI data considering the dDBN method. The last one is to find the effective connectivity of the brain by using the data belonging schizophrenia and control. Since we have a total of four sub-studies, we will introduce each study separately. In Chapter 2 the brain connectivity approaches focusing on effective connectivity will be explained. In chapter 3 discrete dynamic Bayesian networks and structure learning are explained. In chapter 4, sample complexity analysis of discrete dynamic networks with its results and discussions is provided. In chapter 5 evaluation of discretization methods with literature review, results and discussion are explained. In chapter 6, the smoothing step of the fMRI preprocessing is explained. In chapter 7, the effective connectivity approaches are explained with an application on real fMRI data. Lastly, we conclude this thesis.

CHAPTER 2

BRAIN CONNECTIVITY

The main property of the brain's working mechanism is the segregation and integration of the information processed. The paradigm considered in neuroscience studies is that the interconnectivity between brain regions is directly related to optimal information processing. Functional interactions between regions of the brain are observed by the synchronized activation between both local and distant regions. In other words, the brain is a complex structure that can be spatially distant from each other but functionally interacts with each other. Brain connectivity is generally studied under three headings [10].

- Structural connectivity: This connectivity examines the anatomic connectivity of distant neuron assemblies connected by axonal pathways of the brain regions [11]. The information sent in the axons is transmitted to other regions via synaptic connections [12]. We call all these axon pathways of the brain as white matter. This connectivity is expected to be more stable and the same for every person since it shows the direct structural property of the brain. It is more stable and permanent than other connectivity methods.
- Functional connectivity: It shows whether the neurons that are considered spatially separate have similar activation patterns during any functional task [2], [13]. It shows the statistical dependence between different brain regions in information processing. Therefore, this method is based on statistical measurement methods such as correlation, covariance, and coherence.
- Effective Connectivity: This is the connectivity that possesses the effect of one neural system to another neural system [2], [13]. It shows the temporal relationship between brain regions. It is defined as a directional map of

connectivity between brain regions. This temporal causality is usually obtained by time series analysis of data from brain regions.

2.1 The Methods for Effective Connectivity

Although there are several methods proposed to study effective connectivity of the brain using fMRI and EEG data, in this section, commonly used effective connectivity methods are explained. These methods are Granger Causality, Structural Equation Modelling, Dynamic Causal Modelling and Bayesian Networks.

2.1.1 Granger Causality

Suppose we have two time series, x and y. Our goal is to find the causality between x and y. If a time series x provides predictive information about the future of time series y better than past values of y, x is said to Granger-cause y [14], [15]. An autoregressive model is used to find this causality. In this section, we will explain how to calculate causality only for two variables. The same rule may apply to more than two time series. A univariate autoregressive model will be generated first.

$$x(t) = \sum_{k=1}^{p} a_k x(t-k) + u_1(t)$$
$$y(t) = \sum_{k=1}^{p} b_k y(t-k) + v_1(t) \quad (2.1)$$

In the given equation, a_k represents the linear relation between the time series x at a particular time point t and its k previous values. k indicates the index of temporal dependency. This equation is fitted separately for x and y, then u_1 and v_1 vectors are obtained as error of prediction. The magnitude of these vectors shows how suitable our data is for the given model. Secondly, x and y are then fitted to a bivariate autoregressive model which is expressed by the following equations.

$$x(t) = \sum_{k=1}^{p} a_k x(t-k) + c_k y(t-k) + u_2(t)$$
$$y(t) = \sum_{k=1}^{p} b_k y(t-k) + d_k x(t-k) + v_2(t) \quad (2.2)$$

In equation 2.2 C indicates the linear relation between x(t) and y(t-k). u_2 and v_2 show the prediction error due to the fitting process. In the Granger causality method, the variances of the error vectors are used to check the strength of the effective connectivity. The following equations show variance calculations.

$$\sigma_{x|x} = var(u_1)$$
$$\sigma_{y|y} = var(v_1)$$
$$\sigma_{x|xy} = var(u_2) \quad (2.3)$$
$$\sigma_{y|yx} = var(v_2)$$

Then Granger causality of y over x and x over y are calculated by the following equation.

$$F_{Y\to X} = \frac{\sigma_{x|x}}{\sigma_{x|xy}}$$
$$F_{X\to Y} = \frac{\sigma_{y|y}}{\sigma_{y|yx}} \quad (2.4)$$

2.1.2 Structural Equation Modelling

The most important aspect of this technique that differs from other techniques in finding effective connectivity is that it considers the anatomical connectivity of the brain [16]. While the structural connectivity of the brain is used as a priori information, a connectivity model is formed using the covariance between brain regions. We will only give introductory information about the model. The model is expressed by the following equation.

$$y = By + \Gamma x + \varepsilon \quad (2.5)$$

y is an m*1 vector representing dependent variables. x is an n*1 vector of independent variables, ε is the error vector, B is a m*n coefficient matrix for dependent variables and Γ is the coefficient matrix for the independent variables x. The diagonal elements of the matrix B are 0 since we think variables do not influence themselves. In order to evaluate the model, covariances of the x and ε are investigated.

$$\Phi = E[xx^T] \qquad \Psi = E[\varepsilon\varepsilon^T] \tag{2.6}$$

Φ is defined as the covariance matrix of x and Ψ is defined as the covariance matrix of error term ε. If Z is a vector containing all the variables in the network which is described in equation 2.7, then the covariance matrix of Z can be defined in equation 2.8, where Z is the n* p matrix of p variables in the network for each n observation.

$$Z = [x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n] \tag{2.7}$$

$$\Sigma_{obs} = \frac{ZZ^T}{N-1} \tag{2.8}$$

The covariance matrix from the model can be calculated by the following expression.

$$\Sigma_{mod} = \begin{bmatrix} \Phi & (I-B)^{-1}\Phi \\ ((I-B)^{-1}\Phi)^T & (I-B)^{-1}(\Gamma\Phi\Gamma+\Psi)((I-B)^{-1}\Phi)^T \end{bmatrix} \tag{2.9}$$

The goal of this method is to minimize the difference between these two covariance matrices. In this minimization, the number of unknowns which is related to B,Ψ,Γ and Φ are more than the number of equations. Therefore, this method needs prior information about the model in order to find the unknown parameters. Thus, this model has to point to the existence of some causal relations among the variables. The remaining parameters are found by fitting the model so that the difference between covariance matrices defined in equations 2.8 and 2.9 is minimized. By using the maximum likelihood method, the effective connectivity between the variables is found by the following expression where tr(.) denotes the trace of a matrix.

$$F = \log|\Sigma_{mod}| + tr(\Sigma_{obs}) - \log|\Sigma_{obs}| - p \tag{2.10}$$

2.1.3 Dynamic Causal Modelling

This modeling addresses causal interactions between brain regions by creating and testing realistic models of interactive neural areas [17]. For this reason, DCM needs extreme prior information. DCM finds the couplings between brain regions and also aims to predict how these are affected by the changes in the experiment. DCM first begins by modeling brain regions that are supposed to interact, and adds a model of how a signal that can be measured by fMRI can form neuronal activity using BOLD response. Modeling is performed with a hemodynamic response function that describes how this neuronal activity transforms the BOLD signal. The DCM method, which models the connectivity with a Bayesian approach, solves the interaction between brain regions with the following equation.

$$\frac{dz}{dt} = \left(A + \sum_{j=1}^{m} u_j B^{(j)} \right) z + Cu \tag{2.11}$$

Here t is the continuous-time, u shows the input given during the experiment. Matrix A shows the interaction between brain regions, independent of the experiment, whereas matrix B shows the interaction resulting from the experimental input. C matrix shows the effect of the experiment input directly on the brain region. The parameters A, B and C in this equation are the parameters to be estimated in the learning part of the model. Therefore, the experimental design has a significant impact on DCM analysis. However, DCM is also used to analyze the effective connectivity for the resting-state fMRI where there is not any experimental input [18].

The computational complexity of this method is immense. Generally, it is investigated by some predetermined models. These specific models are tested by DCM analysis and the best fitting one out of the predetermined models is chosen as the model for the connectivity of the brain, with a Bayesian approach. Then, on the

best model, the strength of the effective connectivity is calculated and expressed to represent how much a brain region affects another brain region.

2.1.4 Bayesian Networks

Bayesian networks are probabilistic graphical models that show the independencies between certain random variables. This model is often preferred for fMRI and EEG data[4], [5], [19]. For effective connectivity studies, dynamic Bayesian networks are preferred over BN. Because this model investigates the temporal independencies between random variables. Dynamic Bayesian Networks are appropriate to model the brain's effective connectivity due to their non-deterministic behavior. Due to the complexity of modeling, two DBN methods are applicable, one is Gaussian DBN [3], where brain regions are modeled with linear Gaussian relations, second one is discrete DBN, where non-linear modeling is possible by discretizing the data and using multinomial distributions over the network parameters [4]–[6]. The connectivity between the brain regions is assumed to be linear in most of the effective connectivity methods such as Linear Gaussian Model, Partial Directed Coherence and Granger Causality, Structural equation modeling, but this linear relationship may not be valid for the brain. Therefore, dDBN, which is a non-linear method, is one step ahead of other effective connectivity methods.

One of the disadvantages of DCM in multiple brain regions is the excess computational needed. Therefore, not all models can be tested. But for ease of calculation, Bayesian networks allow us to test any model. Secondly, the modeling process does not depend on the experimental design. The experimental design is, of course, very critical for fMRI studies, but this method does not require experimental design for analysis. For this reason, BNs stands out one more step.

The detailed background information about discrete Bayesian Networks is provided in the following chapter.

CHAPTER 3

DISCRETE DYNAMIC BAYESIAN NETWORKS

3.1 Introduction

Bayesian networks are directed and acyclic graphical models to represent the joint probability distribution over a set of random variables [20]. A graph is represented by a set of nodes V= {i: i =1, 2, …, n} and edges E= {e_{ij} :e_{ij} = 1 if the *i-th* node is in the parent set of the *j-th* node}. Each node in the graph represents a random variable and edges show the causal independencies between certain variables. Given the parents of any node in the graph, that node is independent of the non-descendants, which is the nature of the probability and graph theory. If a certain amount of data is provided, both the structure of the underlying Bayesian network and the conditional probability distributions of the variables could be modeled. For Bayesian Network that model discrete random variables (discrete Bayesian Network), the causal relationship between a node and its parents are parameterized by conditional probability tables which explicitly describe the probability of the *i-th* node having any particular discrete state given the state of its parents.

Dynamic Bayesian network (DBN) is a graphical model that represents the causal characteristics of the variables over time [21]. Discrete Dynamic Bayesian networks is a specialization of DBN that models the temporal processes between discrete-valued random variables. dDBN is divided into columns of nodes where each column represents the observation of variables for a particular time frame. The edges are only designed to connect nodes between these columns, and edges are always from the previous state to the next state, i.e. edge of a dDBN is designed to point forward in time.

Some simplifying assumptions are used for the sake of dDBN complexity and convergence during model learning. The first assumption is that the time series is

stationary which means that the conditional distribution is the same for all time points. The second assumption is that the model obeys first-order Markov property:

$$P\big(x_i(t)\big|x_1(t),\dots,x_n(t),x_1(t-\Delta t),\dots,x_n(t-\Delta t),\dots,x_1(t-k\Delta t),\dots,x_n(t-k\Delta t)\big) = P\big(x_i(t)|pa_i(t-\Delta t)\big)$$

Here x_i is the *i-th* variable, n is the total number of variables, t is the discrete time, Δt represents the time delay to model the causal relationships, k refers to the order of the model, pa_i represents the parent set of the *i-th* node.

dDBN structure learning targets to find the present edges, e_{ij}, and the conditional probability distribution of the variables based on the existing edges. The graphical structure G is learned from the dataset D:

$$G^* = argmax(P(G|D)) \tag{3.1}$$

By using the Bayes rule

$$P(G|D) = \frac{P(D|G)P(G)}{\sum_G P(D|G)P(G)}$$

and taking the logarithm of both sides and ignoring the denominator, since it is just a constant, the score can be defined as:

$$score(G:D) = log\big(P(D|G)\big) + log(P(G))$$

The right-hand side of the expression is the sum of the likelihood and the prior information about the structure. Generally, the prior is taken as the uniform distribution over the structures, and this concludes that maximizing the total score is the same as maximizing the likelihood:

$$P(D|G) = \int P(D|G,\theta)P(\theta|G)d\theta$$

Here θ is the parameter set that defines the conditional distributions over the random variables of the given structure G. Several distributions have been used for the prior $P(\theta|G)$. Cooper and Herskovits [22] take the prior as uniform distribution for each parameter of θ and apply the well-known K2 score. Heckerman et al. [23] use

Dirichlet distribution with parameters α (imaginary sample size) and obtain the following expression for the likelihood:

$$logP(D|G) = \log\left(\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}\right)$$

$$logP(D|G) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\log\left(\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})}\right) + \sum_{k=1}^{r_i} log\left(\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}\right) \quad (3.2)$$

Where q_i denotes the total number of parental configurations of i-th node, r_i represents the total discrete states that i-th node can take, n is the number of nodes, N_{ij} represents the number of samples that i-th node is observed given the parental configuration represented by j, N_{ijk} is the total number of samples that i-th node take one of its discrete state k given the parental configuration represented by j, α_{ij} and α_{ijk} are the prior distributions for N_{ij} and N_{ijk} and they are specified by α/q_i and $\alpha/q_i r_i$ respectively. This equation is called the Bayesian Dirichlet equivalence with a uniform prior (BDeu).

3.2 BDeu and BIC Scores

Suppose that the data D consist of M samples, when M→∞ we have that[24]:

$$logP(D|G) = L(\theta_G : D) - \frac{\log M}{2} Dim[G] + O(1) \quad (3.3)$$

Where $L(\theta_G:D)$ is the maximum log-likelihood of parameters of the graph, θ_G and $Dim[G]$ is the model dimension, or the number of independent parameters in G. Without the last term $O(1)$, remaining expression is called Bayesian Information Criteria (BIC) which is also commonly used as a scoring method for Bayesian networks. If only the first term, the log-likelihood score, is used to find the best structure, some limitations will be faced. For example, log-likelihood tends to prefer

networks with more parents as the samples of variables increase [24]. However, adding the second term Dim[G] decreases the score of complex structures. This leads to a tradeoff between fit to data and model complexity: as the dependence between a variable and its' parents increases, we get higher score due to the likelihood term, however as the network gets more complex, we get lower score due to the second term in equation 3.3.

The likelihood score can be decomposed as follows:

$$L(\theta_G : D) = M \sum_{i=1}^{n} \boldsymbol{I}\left(x_i; pa_i^G\right) - M \sum_{i=1}^{n} \boldsymbol{H}(x_i) \tag{3.4}$$

Where $\boldsymbol{I}(x_i; pa_i^G)$ is the mutual information between the random variable x_i and their parents, and $\boldsymbol{H}(x_i)$ is the entropy of variable x_i. Using equation 3.3 and 3.4 we can write the BIC score as follows:

$$logP(D|G) = M\left(\sum_{i=1}^{n} \boldsymbol{I}\left(x_i; pa_i^G\right) - \sum_{i=1}^{n} \boldsymbol{H}(x_i)\right) - \frac{\log M}{2} Dim[G] \tag{3.5}$$

Suppose that we try to find the difference of BIC scores between two graphs, namely G_1 and G_2.

$$score_{bic}(G_1 : D) - score_{bic}(G_2 : D) = \Delta M - \frac{\log M}{2}\left(Dim[G_1] - Dim[G_2]\right)$$

$$\Delta = \sum_{i=1}^{n} \boldsymbol{I}\left(x_i; pa_i^{G_1}\right) - \sum_{i=1}^{n} \boldsymbol{I}\left(x_i; pa_i^{G_2}\right) \tag{3.6}$$

Equation 3.6 is composed of two terms: the first term is the difference due to the likelihood, the second term is the difference of scores due to the model dimension. The first term changes linearly with the number of samples M, however, the second term changes logarithmically. This affects the structure obtained during learning when different numbers of data samples are used from the same underlying

probability distribution which is to be modeled. In order to analyze the effects of different configurations, G_1 and G_2, on BIC metric, let's discuss some special cases:

- Let G_1 be the actual structure from which the data is sampled, in other words, it is the true structure G*, and G_2 is any graph that does not contain all of the temporal relations between random variables that the true structure does. As $M \to \infty$ Δ will be positive and high because G_2 contains different temporal relations than G*. Therefore, the first term of equation 3.6 starts to dominate the second term, as a result of the fact that the linear term increases faster than the logarithmic term. Therefore, the BIC score of G1 will be higher than that of G2.
- Let G1 and G2 both contain the same temporal relations as the true structure G*, however, both have a higher number of edges than G*. In this case, the likelihood term of the equation 3.6 will converge to 0 since both graphs indicate the same temporal relations with the true structure, whereas the second term will be different for G1 and G2 if they differ in terms of model dimension. Hence, the structure with a lower number of parameters, Dim[G], will get the highest score. This result concludes that the graphs which have the same temporal relations as G* but have a higher number of edges will get lower score than G*.

We conclude from these configurations that when the amount of data $M \to \infty$ G* is the structure that maximizes BIC score and all structures other than G* have strictly less scores. This property is called the consistency of a score, hence BDeu metric is a consistent score because for $M \to \infty$ BDeu score is approximated as BIC score; see equation 3.3 [24].

Another important property of BIC and BDeu metrics is the score decomposability: the total score of a certain structure can be written as the sum over family scores of individual nodes in the structure. By using score decomposability, we can write the equation 3.2 and 3.3 as:

$$score(G:D) = logP(D|G) = \sum_{\mathrm{i}}^{\mathrm{n}} FamScore(x_i|pa_i^G:D)$$

Where the family score of each node is defined for the BDeu score:

$$FamScore(x_i|pa_i^G:D) = \sum_{j=1}^{q_i} \log\left(\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})}\right) + \sum_{k=1}^{r_i} log\left(\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}\right)$$

Score decomposability provides efficient learning algorithms since it allows local search [25]. Maximizing the overall score can be reduced to several optimization problems with only maximizing the individual family scores. For a given node, all possible parent combinations are traced and the one with the highest score is taken as the family of that node. This method is applicable for DBN because adding any edges to the graph of a DBN will not violate the directed acyclic graph property of the graph; since edges of DBN show the independence between the time slices. For Bayesian Networks, this method is applicable if and only if the order of the nodes is predetermined [24], [25].

CHAPTER 4

SAMPLE COMPLEXITY ANALYSIS OF DISCRETE DYNAMIC BAYESIAN NETWORKS

4.1 Introduction

In the literature, researches on the adequacy of the number of samples consider mostly the conditional probability distributions of Bayesian networks (BNs). They examine the error between the actual distribution from which the data was sampled and the distribution learned from the sampled data using Hoeffding's inequality [24], [26]–[29]. All these studies use the Kullback-Leibler distance between the original and the learned model distributions to decide on the sample complexity. Zuk et al. [30] go beyond finding the right distribution and show the relationship between the correct structure with lower bounds and the number of samples. They argue that the amount of the samples should be more for finding the correct structure than finding the correct distribution. They demonstrate this by using the Hoeffding's inequality and the relative entropy distance, and state that the probability of the correct structure's score being greater than the score of any other structure is a function of the number of samples. They concentrate on Bayesian Networks with binary random variables and state the bounds on the probability of learning a wrong structure when Bayesian Information Criteria (BIC-score) is used. Ghoshal and Honorio [31] study on the information-theoretic limits of learning the structure of Bayesian Networks with discrete and continuous random variables and show that the minimum number of samples by any procedure to recover the correct structure grows with the number of random variables, for non-sparse Bayesian Networks. Dai et al. investigate the relation between the sample size and the error on model discovery (structure learning) [32]. The synthetic data generated from known models of various complexity are used, and the effect of sample size on two learning procedures is

searched. Their results show that increasing model complexity requires more samples to discover the model correctly. They also investigate the effect of the weak link (edge) on the model discovery and find out that finding a weak link through learning requires more samples. They have not analyzed the effect of parent size and number of random variables systematically. But their results help to understand that the number of samples is critically important to discover the correct structure. Brenner and Sontag [33] propose a new scoring method for Bayesian Networks, which has a sample complexity of the order O (n^2), where n is the number of binary nodes in the network. In addition, they compare their method with Bayesian Information Criteria metric (BIC) and Max-Min Hill Climbing (MMHC) method and show that their scoring metric requires less number of samples for discovering the correct structure through the learning procedure.

In order to define the sample complexity of dDBNs, we firstly started with the theoretical studies which define the sample complexity of Bayesian networks (BN) for structure learning. We used approximate methods to obtain a theoretical sample complexity for dDBNs and showed that it is not practical to use the theoretical approaches for dDBNs where the number of samples needed to learn the correct structure practically is far less than the theoretic sample size. Therefore, we developed an experimental method by posing the hypothesis that the practical sample complexity would be less than the theoretical ones for dDBN. We produced synthetic data for binary and ternary valued random variables, to see the effect of the cardinality on sample complexity. The structures from which the synthetic data was produced were carefully selected to observe the effect of the number of nodes and the number of parents accurately. We then examined the effect of the number of samples on structure learning with the BDeu score where the error is defined as structural Hamming distance between the learned structure and the ground-truth structure from which the data was generated. We then examined the relationship between the error due to the number of samples and the dDBN parameters such as parent size, cardinality and node numbers. Finally, we reached a practical definition of sample complexity for binary and ternary valued dDBN.

4.2 Theoretical Sample Complexity of dDBN

In this section, the relationship between the model discovery for the dDBNs and the number of samples, i.e., sample complexity, will be explained. Let G^* be the real structure and P_{B^*} be the corresponding probability distribution from which the data is sampled. Our goal is to find the relationship between the score of any graph G and the actual structure G^*. We will discuss the results presented in Zuk et al. [30] where they use relative entropy distance and examine the probability of the correct structure's score being smaller than the score of any other structure, given the number of samples. In this study, graph G has been examined considering two cases: Graphs that are not I-maps for P_{B^*}, and graphs which are I-maps for P_{B^*}, yet have a higher dimension than G^*. Since the second case is not valid for dDBN, we will only consider the first case and specify the effect of the number of samples. They conclude the sample complexity study for binary random variables with the following expression:

$$P(S_M(G*) < S_M(G)) \leq \binom{n}{2} n 2^{n+3} e^{-\sigma^2 M/3} \tag{4.1}$$

In this expression, S_M is the scoring function of the Bayesian Network, n is the number of random variables, M is the number of samples, and σ is the following expression:

$$\sigma = \min\left\{\frac{\gamma^n}{2}, \frac{IC_B}{2^{n+2}|n\log\left(\frac{\gamma}{2}\right)+1|}\right\} \tag{4.2}$$

$$IC_B = \min_{i,j}\left\{\min_{s\subset\{x_1,\dots,x_n\}\setminus\{x_i,x_j\}}\left\{I_{P_{B^*}}(x_i,x_j|S)\right\}\right\} \tag{4.3}$$

In this expression γ is the minimum conditional probability distribution in P_{B^*}, and IC_B is the minimum information content in P_{B^*}. Equation 4.1 gives the probability of maximizing a wrong structure with respect to the parameters of BNs. If we leave M

in this inequality alone and the probability is assumed to be smaller than δ, we get the following inequality:

$$M \geq \frac{3}{\sigma^2} ln\left(\frac{\binom{n}{2}\, n\, 2^{n+3}}{\delta}\right) \tag{4.4}$$

For equation 4.4, if σ gets smaller, the minimum required sample size M increases. To be able to get a general expression for M we need to consider the worst case, hence we need to find the minimum σ value using equation 4.2. In this equation, γ determines whether the first expression or the second expression should be taken for minimum σ. γ shows the lowest conditional probability distribution and we cannot make any assumptions for the minimum of this value. The exponential representation of γ in the first expression shows that this expression is more dominating than the second one since γ can take random values in the interval [0, 1/K], where K is the cardinality of random variables. As a result, the first expression should be taken for σ. The proof of this decision is provided in Appendix A. So, the following approximate expression is considered for the minimum number of samples:

$$M \cong \frac{12}{\gamma^{2n}} ln\left(\frac{\binom{n}{2}\, n\, 2^{n+3}}{\delta}\right) \tag{4.5}$$

The most effective part of this equation is γ because it is the lowest probability distribution, and the highest value it can get in a binary network is ½. Even if γ is ½, the minimum number of samples (M) increases proportionally to 2^{2n}. This confirms the need for an extremely high amount of data. Table 4-1 shows the required number of samples for learning the structure which includes binary-valued random variables based on equation 4.5. These values are not practical because obtaining these amounts of samples is far from reality. Our practical results indicate that much less sample size is enough to learn the correct structure for dDBNs.

Table 4-1: M values for various γ and n for δ=0.1

		n				
		3	4	5	6	7
γ	0.5	6,65E+03	3,17E+04	1,45E+05	6,41E+05	2,80E+06
	0.3	1,43E+05	1,89E+06	2,39E+07	2,94E+08	3,57E+09
	0.1	1,04E+08	1,24E+10	1,41E+12	1,56E+14	1,71E+16

4.3 Effect of Number of Samples on Structure Learning

In this section, the effect of sample size on structure learning of dDBN is explained by practical experiments using synthetic data. First of all, various synthetic data was generated based on a network model which consists of binary or ternary valued discrete random variables with known conditional probability distributions. In order to cover all possible parental relationships, each variable in the model has a different number of parents. For example, for a three-node graph, one node has three parents, i.e., has connections from all of the nodes, one node has two parents and the remaining node has a single parent. As an example, Table 4-2 and Figure 4-1 show connectivity relations among six random variables. Node number 6 has a single connection from node number 1, whereas node number 1 has connections from all six nodes. Using this kind of structure provides us to see the effect of the parent size and node size separately in the sample complexity analysis.

Table 4-2: An example of connectivity for a six-variable network, from rows to columns. If there is a connection, the cell has value 1.

nodes	1	2	3	4	5	6
1	1	1	1	1	1	1
2	1	1	1	1	1	0
3	1	1	1	1	0	0
4	1	1	1	0	0	0
5	1	1	0	0	0	0
6	1	0	0	0	0	0

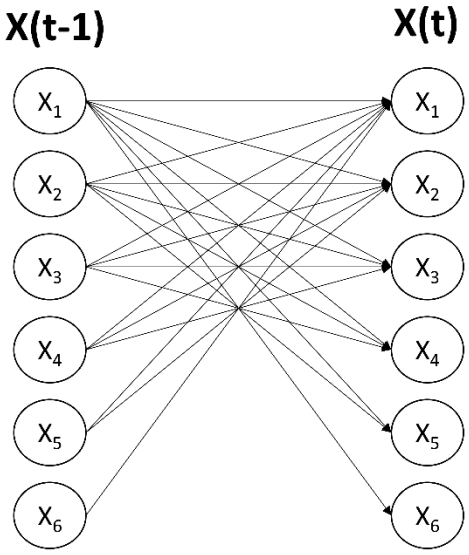

Figure 4-1: Nodes and edges of a six-variable network, each node having a different number of parents.

For binary-valued random variables, synthetic data was created from networks having node numbers changing from three to ten, based on the approach for possible structures defined above. For each different network structure, one hundred different time series were generated, each of which was created from a network having different conditional probability distributions. No constraint on the probability distributions was made, which is more fair. Similarly, synthetic data was produced for the structures which include ternary valued nodes, with one difference, data was generated from the structures which have node number starting from three up to eight. Because learning the minimum number of samples for more than eight nodes requires very high computation power. The structures used for synthetic data creation were kept as ground truth.

In chapter 3 it was described that finding the best structure for a dDBN is the same as finding the best parent combination for the individual nodes. Learning parents of the first node in Table 4-2 does not depend on whether parents of other nodes are learned or not, due to score decomposability of the BDeu metric. This leads to the freedom of analyzing each node separately. As a result, the error between the ground truth structure and the learned structure (in terms of average structural Hamming

distance) was recorded separately for each node. The following equation shows the structural error, where G_i is the learned structure of the *i-th* node, G_i* is the ground-truth structure for that node and n is the total number of nodes.

$$error = \frac{1}{n}\sum_{i=1}^{n} |G_i - G_i^*| \tag{4.6}$$

4.3.1 Data size analysis

In order to investigate the effect of sample size on the convergence of structure learning for dDBN, learning was performed for various sample sizes, where the imaginary sample size is considered as 1. For example, for a six-node network consisting of binary random variables, to analyze the effect of data size for node number 1, structure learning was performed when the number of samples is increased from 10 up to 100.000 in an exponential manner. The same procedure was also performed for the networks having ternary valued variables, but this time the length of the generated data was determined so that learning successfully finds the ground-truth structure. During structure learning, if error dropped to 0, which means the structure was found perfectly, the algorithm was terminated to save from computation time, and the length of the data at the termination time was recorded as to be sufficient. Figures 4-2 and 4-3 show the mean structural error of the hundred-time series versus the number of samples for binary and ternary nodes having various number of parents. It is observed that there should be a minimum number of data samples to discover the model correctly. Explicit analysis of this figure will be explained in more detail in the next chapter by considering the imaginary sample size of the BDeu metric. Also, in Tables 4-3 and 4-4, the minimum required number of samples that are needed for learning the structure of networks having various node numbers and various parent numbers are listed, for binary and ternary variables, respectively. In order to obtain the minimum number of samples for an error to be 0.1, we used linear interpolation between successive error values.

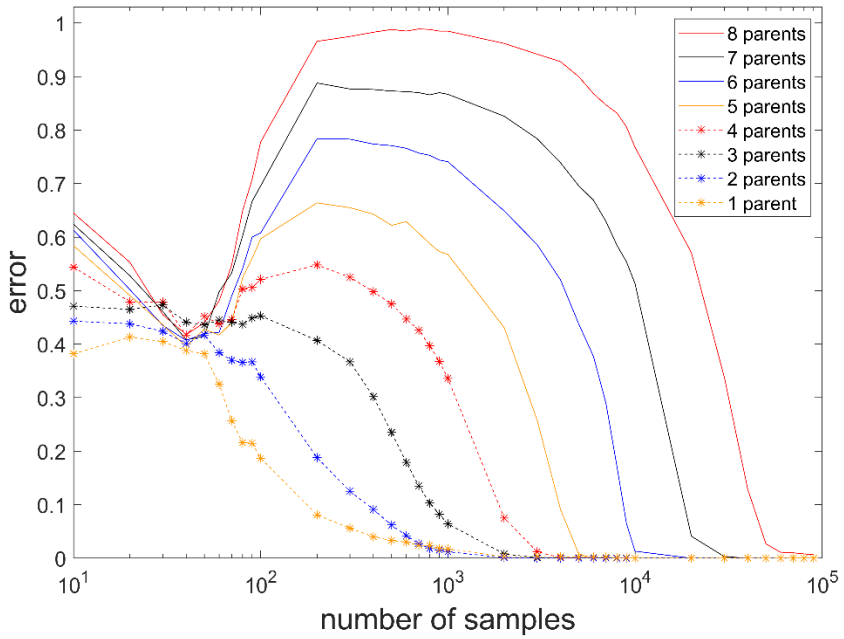

Figure 4-2: Mean error vs number of samples for various parent sizes for an 8-node network with binary nodes.

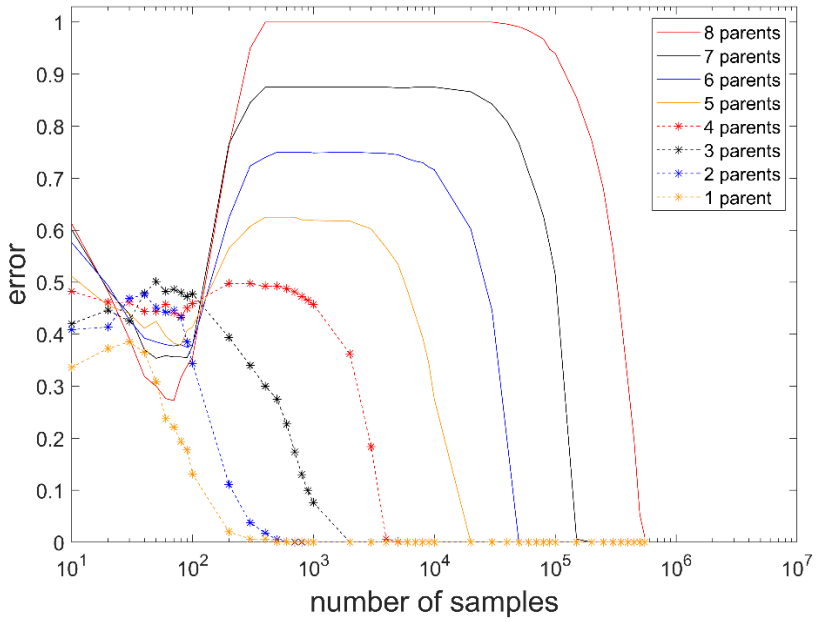

Figure 4-3: Mean error vs number of samples for various parent sizes for an 8-node network with ternary nodes

Table 4-3: Minimum required number of samples for various number of nodes and parent sizes, to find the correct dDBN structure with a mean error smaller than 10 percent for binary-valued random variables.

number of nodes \ parent size	1	2	3	4	5	6	7	8	9	10
3	216	440	480	-	-	-	-	-	-	-
4	162	197	414	720	-	-	-	-	-	-
5	68	175	344	653	1455	-	-	-	-	-
6	75	154	262	546	1175	2308	-	-	-	-
7	56	108	223	447	1000	1962	4444	-	-	-
8	57	78	176	411	953	1915	4322	9192	-	-
9	78	91	160	384	856	1935	4089	8870	19221	-
10	83	95	182	374	814	1904	3946	8673	18750	42843

Table 4-4: Minimum required number of samples for various number of nodes and parent sizes, to find the correct dDBN structure with a mean error smaller than 10 percent for ternary valued random variables.

number of nodes \ parent size	1	2	3	4	5	6	7	8
3	87	368	1378	-	-	-	-	-
4	58	344	1184	3819	-	-	-	-
5	57	282	989	3670	17200	-	-	-
6	62	221	963	3606	17391	44955	-	-
7	93	204	908	3530	16943	45035	148776	-
8	128	215	896	3472	16364	44872	140786	483482

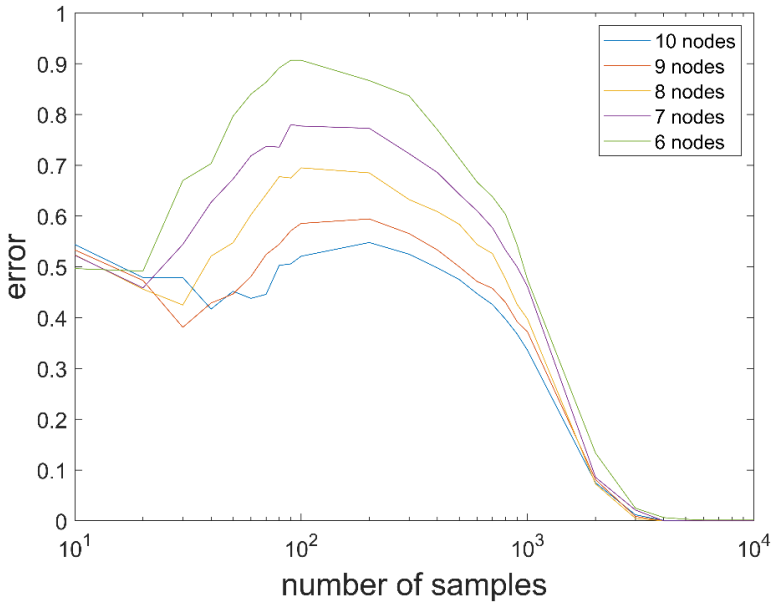

Figure 4-4: Mean error vs number of samples for various node numbers where each node has the same number of parents. This figure is for binary-valued networks, and each node has six parents.

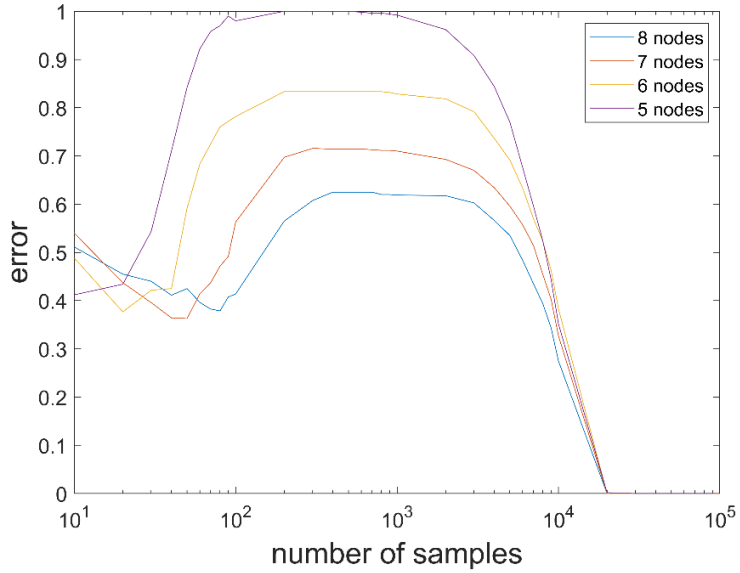

Figure 4-5: Mean error vs number of samples for various node numbers where each node has the same number of parents. This figure is for ternary-valued networks, and each node has five parents.

Figures 4-4 and 4-5 illustrate an example of the average structural Hamming distance between the actual and learned structures for various number of nodes that have the same number of parents, 6 for binary and 5 for ternary. Here our purpose is to see the effect of node number on structure learning where the number of parents is kept constant for all nodes. This result concludes that the minimum number of samples to guarantee a successful dDBN structure learning depends on the parent size, not the number of random variables.

4.3.2 The expression for practical sample complexity

In this section, the minimum number of samples that is required for the convergence of dDBN structure learning is expressed as a function of the network parameters. These parameters consist of the number of variables n, the cardinality of random variables K, and the maximum number of parents p. We used the results presented in Tables 4-3 and 4-4 to fit an expression of these parameters. The expression of practical sample complexity represented only when imaginary sample size is 1.

Let's start with the simple expression,

$$M = \min_{M} \frac{|G(M) - G^*|}{n^2} > \epsilon \tag{4.7}$$

where M is the length of time series, $G(M)$ is the structure found by using M samples of data, G^* is the ground truth structure, n is the number of random variables. This equation implies that our objective is to find M, which is the minimum required number of data samples that guarantees the structure is found correctly with an error ϵ. We divided the structural error term by n^2 to normalize it. Note that a structure can be represented by a $n \times n$ matrix with n^2 edges.

First of all, this equation can be divided into sub-optimization problems, one for each node in the network. In other words, the minimum number of samples to find parents of a random variable i can be expressed independently as follows:

$$M_i = \min_{M_i} \frac{|G_i(M_i) - G_i^*|}{n} > \epsilon \tag{4.8}$$

In this expression, division n^2 is replaced by n, because there are now only n edges in G_i. In order to find the minimum number of samples for the overall network, we need to find the highest M_i:

$$M = max\left(\min_{M_i} \frac{|G_i(M_i) - G_i^*|}{n} > \epsilon\right) \tag{4.9}$$

Define K_i as the cardinality of random variable x_i, i.e., x_i can take K_i possible discrete values. Thus, in a discrete Dynamic Bayesian network, the number of possible parent configurations for the *i-th* node (PC_i) can be defined as:

$$PC_i = \prod_{j \in \{pa_i\}} K_j \tag{4.10}$$

Here pa_i, is the set of parents of node i. Hence, learning $P(x_i|pa_i)$ depends on the observations of x_i and pa_i, this leads to C_i possible configurations:

$$C_i = K_i * PC_i = K_i * \prod_{j \in \{pa_i\}} K_j \tag{4.11}$$

If every random variable on the network has the same cardinality K, this equation can be simplified as,

$$C_i = K^{p_i+1} \tag{4.12}$$

where p_i is the number of parents of the variable x_i.

Secondly, we continued with the following assumption: learning the structure of a dDBN with an error term ϵ, approximately depends on an error term λ times C_i. In this term, λ still may depend on the network parameters as well, but we found that

this exponential relation is suitable to represent the practical sample complexity. Although we did not have a theoretical proof for this assumption, it holds for the practical results. Therefore Equation 4.8 can be written as,

$$\min_{M_i} \frac{|G_i(M_i) - G_i^*|}{n} > \epsilon \cong \lambda(\epsilon, n, K, \dots) * K^{p_i+1} \tag{4.13}$$

which reduces equation 4.9 to,

$$M = max(\lambda * K^{p_i+1}) \tag{4.14}$$

$$M = \lambda * K^{\max\{p_i\}+1} \tag{4.15}$$

Equation 4.15 emphasizes that the required number of samples for a network depends on cardinality *K*, the maximum number of parents and an error term λ. Here, there is only an unknown λ, and we found it by using the practical results.

In the third step, we used the results presented in Tables 4-3 and 4-4 to find the unknown parameter λ. These results were obtained using the synthetic data which was generated for random variables with cardinality 2 and 3, respectively. In Tables 4-5 and 4-6, we listed the minimum required length *M* obtained from Tables 4-3 and 4-4 where the maximum parent size was taken as equal to the node number in the network.

Table 4-5: Minimum data length for various parent size with K=2 and ϵ=0.1

	parent size							
p	3	4	5	6	7	8	9	10
M	480	720	1.455	2.308	4.444	9.192	19.221	42.843

Table 4-6: Minimum data length for various parent size with K=3 and ϵ=0.1

	parent size					
p	3	4	5	6	7	8
M	1.378	3.819	17.200	44.955	148.776	483.482

Finding λ is an optimization problem which finds the best fit of Equation 4.15 to the data presented in Tables 4-5 and 4-6:

$$\underset{\lambda}{\operatorname{argmin}} \ |M - \lambda K^{\mathrm{p}+1}|$$

Note that, for simplicity, p is used for $max\{p_i\}$. The main problem in this expression was that the relationship between the required data length and the parent size is exponential. Hence using any curve fitting method would tend to fail as a result of the fact that minimization is mostly affected by larger values of p. To overcome this problem, we took K logarithm of the minimization problem. Therefore, every value of p affected the minimization process with equal weights.

$$\underset{\lambda}{\operatorname{argmin}} \ |log_K M - log_K \lambda - p - 1| \qquad (4.16)$$

For binary case (K=2) λ was found as 20.7116, and for ternary case (K=3) it was found as 20.4071. Figure 4-6 shows the plots of minimum data length versus parent size based on Tables 4-5 and 4-6, as well as the plots of Equation 4.15 with the computed λ values.

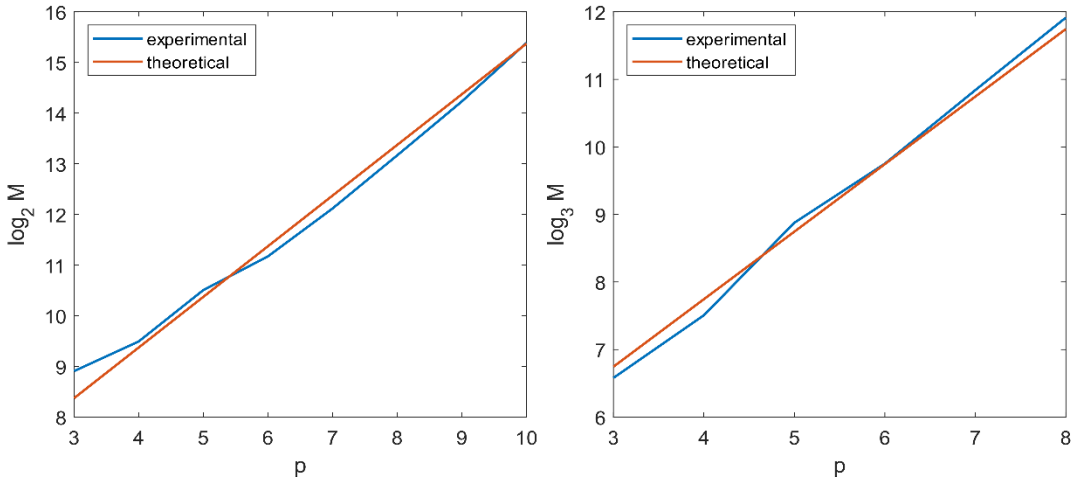

Figure 4-6: Experimental and theoretical plots of minimum data length versus parent size, left for the binary case and right for the ternary case.

Note that this optimization process is a linear regression on p values, and suppose that $f(p)=a*p+b$. One difference here is that the linear coefficient a is 1, hence the aim of this optimization is only to find the constant b that is added to the linear term. This is due to the previous assumption. Figure 4-6 underlines that the assumption is correct because of the prominent consistency between the theoretical linear relation and the experimental results presented in Figure 4-6. Even though we put a restriction on the fitting process by predetermining the parameter a, the fitting was quite successful.

In order to verify the assumption in more detail, we checked the goodness of fit of the optimization process for different error values. R^2 is a suitable metric to check the goodness of fit for linear regression. Suppose we have a distribution over (x,y) variables, and the aim is to find a linear function $f(x)=y=ax+b$ that fits the (x,y) pairs. R^2 was computed by the following equation:

$$R^2 = \frac{\sum(\hat{y} - \bar{y})^2}{\sum(y - \bar{y})^2} \tag{4.17}$$

where $\bar{y}$ is the mean of all y values, $\hat{y}$ is the computed values by using the function $f(x)$. R^2 takes values in the interval [0 1]. When R^2 is 1 it means that model $f(x)$ fits the data perfectly. When it is 0 the model does not reflect the relation between x and y. For each error values, Tables 4-5 and 4-6 were recomputed using linear interpolation between successive error values. Then the same curve fitting approach was investigated and corresponding λ and R^2 were computed. Table 4-7 gives the corresponding lambda and R^2 for different error values. This table signifies two important results. Firstly, λ still depends on the cardinality of the random variables even for the given assumption over complexity. The same lambda values were not obtained considering the same errors. Secondly, the fitting process to the expression described in Equation 4.15 was perfect. R^2 was obtained as near to 1 for each error and cardinality values. Therefore, assumption over sample complexity in Equation 4.15 is verified. Despite determining a in the linear regression as 1, getting R^2 close to 1 signifies that the minimum number of samples for the convergence dDBN structure learning is proportional to K^{p+1}.

Table 4-7: Complexity coefficient λ and R^2 for different errors

	K=2		*K=3*	
error	*λ*	R^2	*λ*	R^2
0,03	30,6	0,936	23,3	0,996
0,04	27,8	0,951	22,8	0,996
0,05	25,3	0,960	22,3	0,995
0,06	24,0	0,971	21,8	0,994
0,07	23,0	0,976	21,5	0,994
0,08	22,2	0,981	21,1	0,994
0,09	21,5	0,985	20,8	0,994
0,1	20,7	0,987	20,4	0,993
0,11	19,9	0,990	20,0	0,993
0,12	19,3	0,991	19,7	0,992
0,13	18,8	0,992	19,3	0,991
0,14	18,2	0,993	18,9	0,990
0,15	17,7	0,994	18,6	0,989
0,16	17,2	0,994	18,2	0,988
0,17	16,8	0,995	18,0	0,988
0,18	16,4	0,996	17,7	0,988
0,19	16,1	0,996	17,5	0,987
0,2	15,8	0,996	17,3	0,987

4.3.3 Effect of Imaginary Sample Size on Sample Complexity

In this section, the aim is to investigate the effect of imaginary sample size *α,* on the sample complexity of dDBN. Several studies conduct that imaginary sample size has a significant impact on the model discovery of Bayesian Networks. Steck and Jaakkola show that as the imaginary sample goes to zero, deletion of an edge is more likely to occur in the structure learning of Bayesian Networks [34]. The learned

graph becomes an empty graph when α goes to zero. In the same study, they also demonstrate that the number of edges in a network increases when the prior term increases. Silander et al. conduct practical experiments on structure learning to find an optimal alpha value [35]. They show that learned structure is highly sensitive to the chosen alpha value. In order to solve this problematic effect of the prior term, they propose a Bayes method for determining the optimal alpha. Steck provides an analytical approximation to the optimal alpha value in a predictive sense [36]. The data properties that have the main effect for determining optimal alpha value are provided by considering this approximation. Ueno analytically investigated the behavior of the BDeu metric when alpha goes to zero and infinity [37], [38]. The sensitivity of model discovery to alpha is investigated, and it is shown that when alpha goes to zero BDeu favors an empty graph. If alpha tends to infinity, BDeu favors a complete graph. In his studies, by considering the issues faced with prior term alpha, Scutari experimentally and theoretically show that the BDeu score is not accurate when data is sparse, which is the case when the number of samples is less than the appropriate amount [39], [40]. He proposes a new scoring method, Bayesian Dirichlet sparse, which is more suitable for sparse data. Because of this significant effect of the imaginary sample size in model discovery, we also conducted several experiments to see the effect of it for dDBN.

Figure 4-7 shows the structural error for a ternary valued network consisting of five variables where only the mean error of the node that has five parents is shown. This figure illustrates the error between the true structure from which the data was sampled, and the structure found with dDBN learning using this data. Note that the imaginary sample size for this figure was 1. The graph seems to have three regions. In the first region, the error is around 0.5. It means that when data size is very small, dDBN structure learning ends up with a structure as if it was chosen randomly and does not contain any information about the actual structure. In the second region, the error is highest and stays so for the number of samples M from 70 to 1000. The actual structure from which the data was generated contains all the edges, i.e., fully connected. Getting structural error to be 1 means that the structure found by the

dDBN learning did not include any 1's, hence the learning procedure tries to obstruct any edges, and the BDeu score of the empty structure is higher than any other possible structures. If the amount of data is further increased, in the third region, the algorithm starts to add some parents to the structure and error starts to decrease. When a sufficient amount of data is provided, all parental relations are found correctly by the dDBN structure learning, and error reaches 0. The detailed explanation of this figure with the theoretical analysis of the BDeu score is provided in Appendix B.

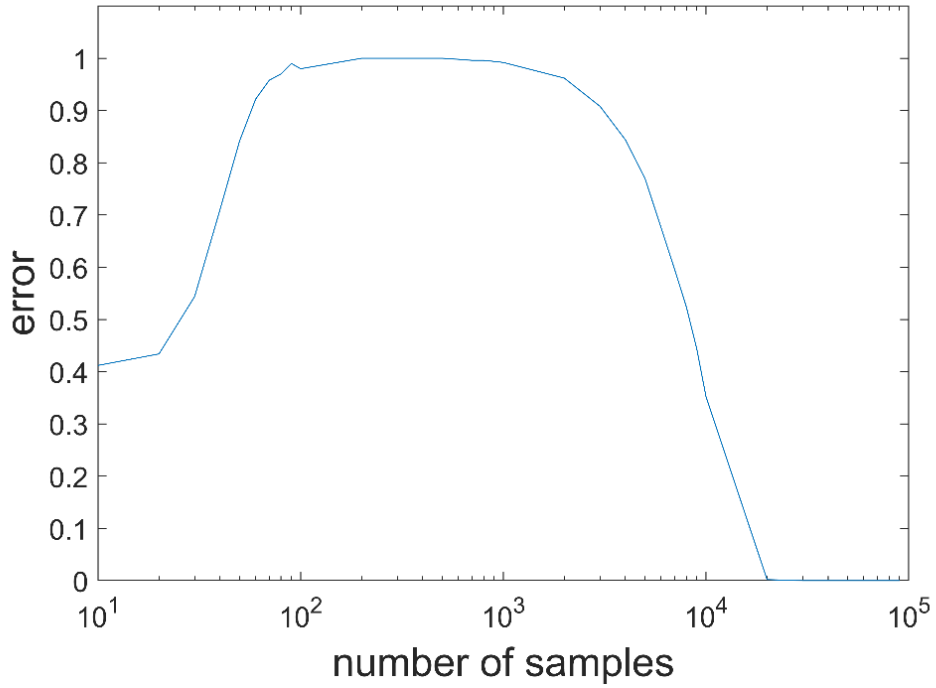

Figure 4-7: Mean error vs number of samples for a node which has five parents in a network of five ternary variables.

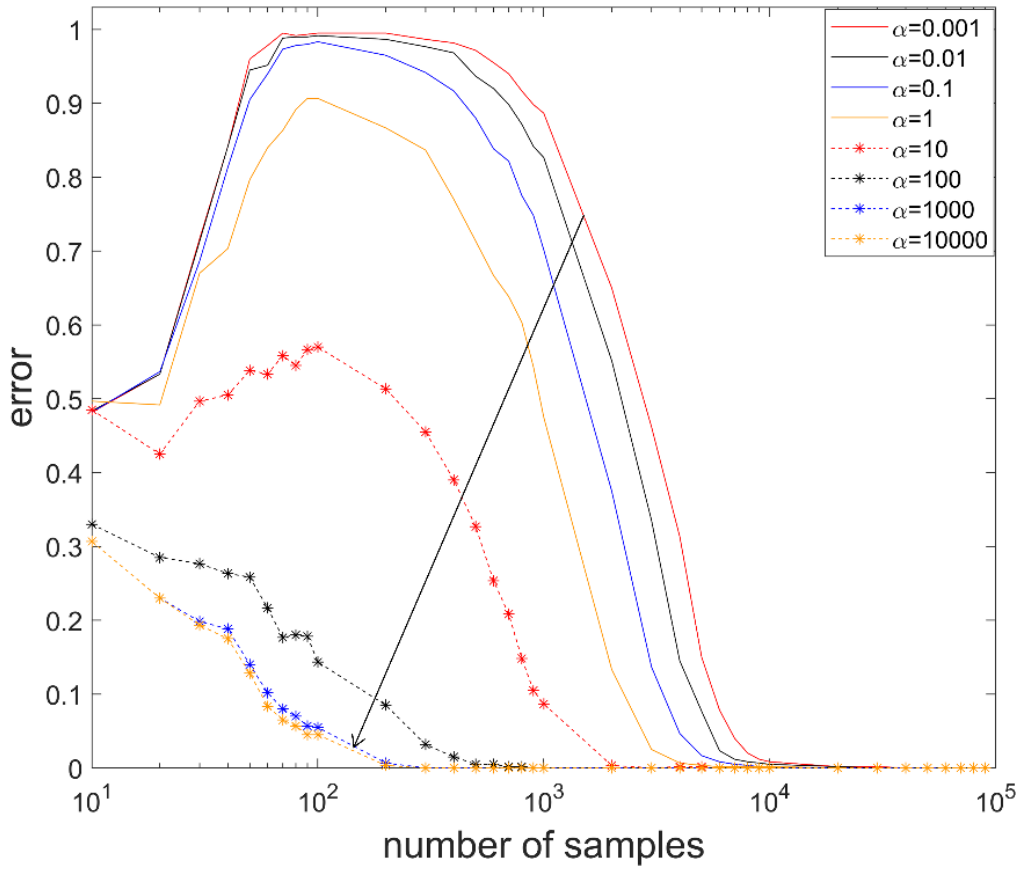

Figure 4-8: Mean error vs number of samples for different alpha values used for six-node binary-valued network. It presents the results for a node that has six parents. The arrow shows the direction of increase in the imaginary sample size, for the easiness of illustration.

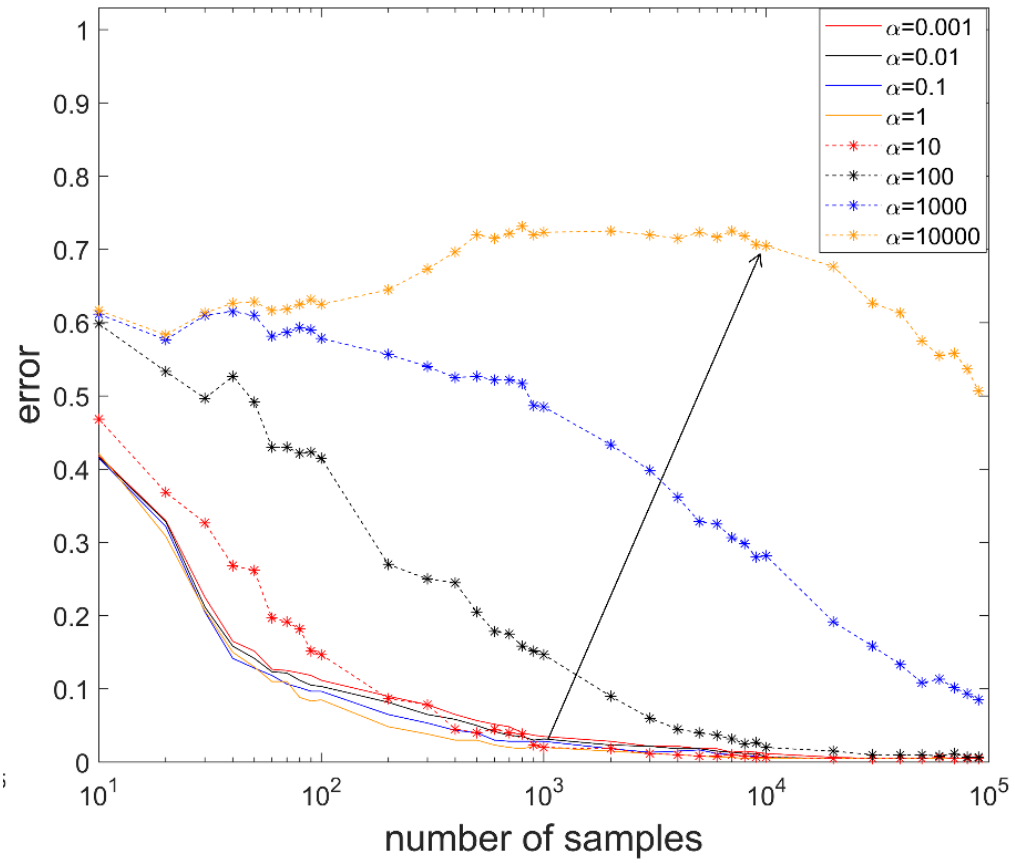

Figure 4-9: Mean error vs number of samples for different alpha values used for six-node binary-valued network. It presents for a node that has one parent. The arrow shows the direction of increase in the imaginary sample size, for the easiness of illustration.

Figures 4-8 and 4-9 illustrate the effect of the imaginary sample size on model discovery when different numbers of data samples are provided. Only the error of the structure of two nodes is provided in these figures, which is considered to represent general cases. For other nodes, the figure has provided in Appendix D. Figure 4-8 is designed for a node that has six parents out of six nodes, and Figure 4-9 is for a node that has one parent. There are three significant results of this analysis. First of all, for smaller imaginary sample sizes, the BDeu metric is likely to obstruct edge addition to the model. Considering Figure 4-8, when the same amount of data samples is provided, the error is higher for smaller imaginary sample sizes. Since the ground-truth structure of this node contains only 1's, the increase of the error means that the learned model does not contain an edge. These results are compatible with the literature experimentally and theoretically [34], [35], [38]. The second result is that when imaginary sample sizes are higher than the number of data samples, BDeu prefers to add edges to the network. Consider Figure 4-9 for the imaginary sample size 10.000. The error increases when more data samples are provided but up to nearly 10.000. The ground-truth structure of the node was [1,0,0,0,0,0], containing a single edge. Therefore, the increase in error means that BDeu prefers to add 1s to the learned model. The third result is that both increasing and decreasing the imaginary sample size, increases the required number of samples to learn the model correctly. When the imaginary sample size is small, BDeu prefers not adding edges. So, to be able to fit the data to the correct model, the required sample size should be high to overcome the property of BDeu that blocks edge addition to the model. When the imaginary sample size is higher, BDeu more likely overfits the data to a model that has unnecessary edges. Therefore, higher amount of data is needed to reflect the model correctly.

This section concludes that the imaginary sample size has a significant effect on the learned model. Therefore, better learning requires an optimal imaginary sample size. Steck and Jaakale face this issue and propose a Bayesian approach to determine the optimal value but do not investigate the problem [34]. Steck experimentally and theoretically provides how to find the optimal value by Bayesian approach and

performs tests on datasets [36]. Silander et al. also propose to be Bayesian on alpha by marginalizing out *P(G)* from *P(G|α)* to find the most probable graph, which may only be applicable to small datasets [35]. Being Bayesian on alpha increases the computational complexity for the learning procedure. However, finding the optimal alpha as a function of network parameters rather than data itself would be more practical and useful.

Since the choice of alpha changes, the required number of samples, an optimal α would indicate a need of less number of samples for the convergence of structure learning. However, our results signified that by providing enough data samples the is discovered correctly for not overrated imaginary sample sizes. For example, for imaginary sizes 0.01, 0.1, 1, 10, 100, both node 1 and node 6 were modeled successfully. Imaginary sample size is considered as 1 in most of the studies. In our simulations taking alpha as 1 always performed reasonably (see Figures 4-2, 4-3, 4-8 and 4-9). Therefore, practical sample complexity was found for α equals 1 in previous section. In this thesis, we left finding an optimal alpha value for dDBN by sample complexity point of view as future work.

4.4 Discussion

Discrete dynamic Bayesian Networks are expressed by two main components: a structure and a parameter set. Structure or model represents the temporal causal independencies between the random variables. Parameters indicate the conditional probability distributions between the random variables based on the structure. The dDBNs can be learned from data. Therefore, the number of samples is provided in the data plays a crucial role to be confident about the learned structure after performing the dDBN structure learning procedure. This study is the first that conducts the sample complexity for discrete Dynamic Bayesian Networks as far as we know.

In this study, we examined the effect of the number of samples on the structure learning for dDBNs. We gathered our results in three headings. First of all, the amount of data has a very important effect on the learning of the correct structure. It was shown that if the amount of data is less than it should be, the learned structure is entirely unrelated to the actual structure. Figures 4-6, 4-8 and 4-9 show that until the amount of data reaches a particular value, the learning procedure maximizes the empty structure for small imaginary sample sizes. In other words, the learning algorithm concludes that there are no dependencies between the random variables, although there is. However, when the amount of data is increased further, the correct structure is learned completely. These results show the importance of the amount of data, i.e. the number of observations, to find the exact model. Secondly, the results were shown to be directly related to the BDeu score. BDeu score maximizes different types of structures depending on the number of samples and the imaginary sample size. For smaller imaginary sample sizes, it gives a random structure as a maximum scored structure when the number of samples is very small. In other words, the learning method using the BDeu score does not provide any information about the independence relations when the number of samples is not enough. A further increase in the number of samples resulted in the learning of an empty structure. This was observed even though there were dependencies between the random variables. That is, the BDeu score started to reject the dependencies due to the increase in the number of samples. This was observed to a certain threshold; when the number of samples exceeded it, the BDeu score maximized the actual structure and all the independence relations was found correctly. For higher imaginary sample sizes BDeu score prefered to add edges to the learned structure. These results are compatible with recent studies [35], [37], [38]. Finally, practical sample complexity for dDBNs was expressed as a function of the network parameters for the imaginary sample size as 1. The minimum number of samples required to recover the correct structure by using the BDeu score is $O(K^{p+1})$, for binary and ternary valued networks where K is the cardinality of the random variables and p is the maximum number of parents present in the network.

The choice of imaginary sample sizes is important for learning the model. Nonetheless, the simulation results of this chapter signified that by providing sufficient data, the problematic effect of imaginary sample size can be negated. But the optimal imaginary sample size would provide a smaller number of samples for the convergence structure learning. In this thesis, we left this issue as future work. In addition, we believe that this study will have a repercussion in the applications of dDBN and that most researchers should carry out their research considering the results of this study. Especially in neuroscience applications, dDBN will be used more effectively considering issues discussed in this study. Researchers using dDBN should consider the effect of the number of samples on structure learning and make modeling in the light of this study. In this way, the results found will be more consistent and more reliable.

CHAPTER 5

EVALUATION OF DISCRETIZATION TECHNIQUES FOR FUNCTIONAL MAGNETIC RESONANCE IMAGING DATA

5.1 Introduction

In general, data discretization is the conversion of continuous features to a set of discrete states. There are several reasons for discretization. Firstly, the learning process from discrete data is more effective and efficient [41], [42]. Secondly, it reduces the required number of samples for the convergence of the learning procedure [43]. Moreover, since data is simplified, the process of learning is much faster in general [44]. Nonetheless, the choice of discretization is not a trivial task, and any discretization method implies information loss from data [44]. The discretization of continuous data has been an important and long-standing problem for machine learning applications [45], [46]. Discretization methods are diverse depending on the application: dynamic vs. static, supervised vs. non-supervised, direct vs. incremental, etc.

The discretization method for effective connectivity with dDBN using fMRI data in a study by Rajapakse et al.[4] is to consider the mean, maximum, and minimum values of the signal. Firstly, time-series are transformed to zero-mean. Then if a value in the time series is higher than one-third of the maximum value, it is discretized as '1'. If a value is smaller than one-third of the minimum value, it is discretized as '-1'. Else it is discretized as '0'.Burge et al. and Dang et al, [5], [6] used the equal width discretization method where the data is transformed into k levels by splitting the data according to its minimum and maximum value. In none of these studies, the discretization methods are investigated and evaluated explicitly for the appropriateness of the resulting models.

dDBN is not only used for fMRI data; it is used in many fields from biomedical data to economics [47], [48] and the Gene Expression Data (GED) as well. In some of these studies, the methods of discretization are also evaluated. A survey of discretization methods is present in the study of Gallo et al. [49]. Maderia and Oliveira [9] explicitly explain and propose many non-supervised methods. Li et al. [8] compare the methods mentioned in [49] for GED data.

5.2 Functional Magnetic Resonance Imaging (fMRI)

Functional MRI data is obtained by measurement based on the level of oxygen in the blood. This level is defined as blood oxygen level-dependent contrast (BOLD). The magnitude of this signal depends on the cerebral oxidative metabolic rate (CMRO2) blood flow, oxygen extraction rate [50]. BOLD signal and electrical activation measurements give us the following important information, the BOLD signal indicates local field potential rather than neural spikes [51]. The spatial resolution of functional MRI is very high compared to other neurologic data such as electroencephalogram (EEG) data. A voxel related to the lowest measurement unit in an fMRI image can be about 3 * 3 * 3 mm^3. This data is a 4-dimensional data, 3 dimensions are x, y, z space coordinates and the fourth dimension is time. The signal obtained during fMRI scanning for a certain stimulus is called the hemodynamic response. Figure 5-1 gives the hemodynamic response function vs time. FMRI data is processed and analyzed by using a variety of tools such as Statistical Parameter Mapping software implemented in Matlab, BrainVoyager, FMRIB Software Library. Local activation in each voxel is modeled using the multiple linear parametric modeling with the general linear model technique using the BOLD signal.

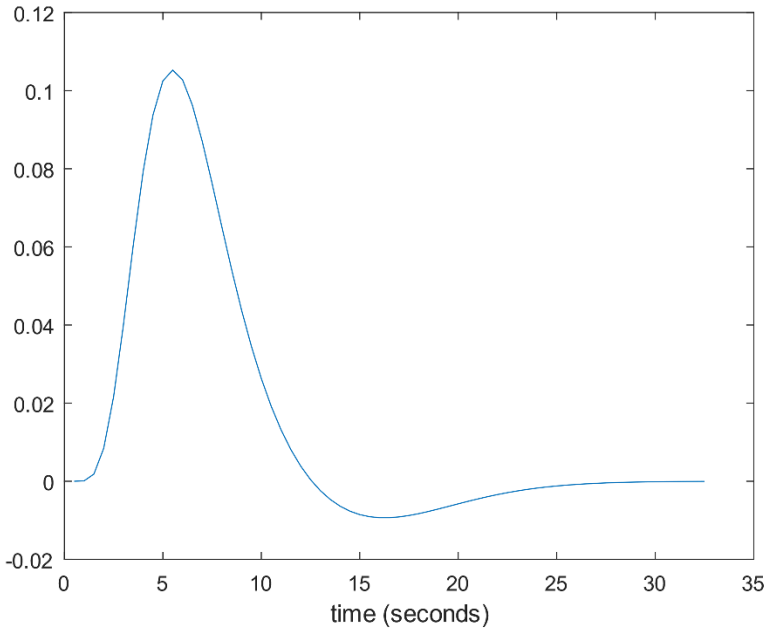

Figure 5-1: Hemodynamic response function

5.3 Discretization Techniques

First, we introduce some definitions and basic concepts. Let x denote the single time-series, and $x[t]$ represents the value for x at time point t. Define $\bar{x}$ as the average of x, x^h and x^l as the highest and lowest values that x can take respectively, and σ represents the standard deviation of x. Moreover, we define d as the discretized version of the time series x and $d[t]$ represents the discrete level of $x[t]$. The discretization methods will be explicated in three categories. The first method is binary discretization, where time series is represented by two discrete states, the second one is ternary discretization where time series can only be discretized to three states, and the last one is multilevel discretization where data can be discretized to any number of levels.

5.3.1 Binary Discretization Methods

The discretization of a data point is to classify the data into two: one is 'activation', and the other one is 'inhibition'. For fMRI data, this method could be feasible if we consider the BOLD response. The BOLD response demonstrates the activity of the brain regions. Therefore, the binarization of an fMRI signal is meaningful from the perspective of neurophysiology since a brain region could be denoted as 'active' or 'de-active' during a specific task. Binarization is usually done by finding a threshold to classify the data into two.

5.3.1.1 Discretization Based on Mean (Mean2)

In this method, each time point is binarized by using the mean of the time series as a threshold δ. Then discretization is done as follows [9]:

$$d[t] = \begin{cases} 1 & if\ x[t] > \delta \\ 0 & otherwise \end{cases} \tag{5.1}$$

5.3.1.2 Discretization Based on Mid-Range (mid-Range)

The only difference between this method and the discretization based on mean is the threshold used in the expression. The threshold for this method is chosen as the midpoint of the data, which is the median. [9].

5.3.1.3 Discretization Based on Max - X% Max (Max-X)

The threshold is fixed with respect to the maximal value observed for the time-series. A percentage of X is reduced from this value and chosen as the threshold, (1-%X)x^h. [9].

5.3.1.4 Discretization Based on Top %X (Top-X)

In this method, time series are split into two sets by finding a threshold that puts %X highest values to one set and remaining ones to another [9].

5.3.1.5 Target Discretization Threshold (TDT)

In this method, data is divided into two states, namely S_1 and S_2 with the following constraint [7]:

$$\min_{S_1,S_2 \subset S} (var(S_1) + var(S_2)) \quad (5.2)$$

Subject to:

- S is the set of sample values for time-series x
- $S_1 \cap S_2 = \emptyset$, $S_1 \cup S_2 = S$, $|S_1|>1$ and $|S_2|>1$
- $var(S_1)$ and $var(S_2)$ are the variances of S_1 and S_2

The sum of the variances of each subset S_1 and S_2 are minimized. This method is similar to K-means clustering, where K is 2. The following steps can be applied for the implementation:

1. Sort the elements of S on an array L.
2. Search for the element e such that *var(L(1..e))+ var(L(e+1 ..|L|))* to be minimum.
3. Save [L(e)+L(e+1)]/2 as the threshold, then use it for discretization expressed in equation 1.

5.3.1.6 Transitional State Discretization (TSD)

This method is proposed to discretize the gene expression data (GED) where the variations between the time points are used [52]. GED data is standardized to mean

of zero and unity variance, and then each gene profile is discretized using the following scheme:

$$d[t] = \begin{cases} 1 & if\ \ x[t] > x[t-1] \\ 0 & otherwise \end{cases} \tag{5.3}$$

In this method, the length of the resulting discrete time series is reduced by a one-time point.

5.3.1.7 Extended TSD

Erdal et al. [53] develop a method related to TSD but introduce a threshold for discretizing the data points. The threshold is computed as follows; the standard deviation of time point 0 is calculated, *std(0),* then a parameter α is provided to scale *std(0).* In order to use it for fMRI data, we used standard deviation, *std* of the time series.

$$d[t] = \begin{cases} 1 & if\ \ x[t] - x[t-1] > \alpha * std \\ 0 & otherwise \end{cases} \tag{5.4}$$

5.3.2 Ternary Discretization Methods

The aim of ternary discretization is to represent the data point by three discrete states {-1, 0, 1}. These states mean 'DownRegulated', 'NoChange', 'UpRegulated' respectively. Several methods are conducted for ternary discretization.

5.3.2.1 Mean and Standard Deviation (mean-std α)

This method combines the mean $\bar{x}$ and standard deviation σ to discretize the data. Let α be a parameter used to manage the deviation from the mean of the data and then, the discretization is performed as follows [9].

$$d[t] = \begin{cases} -1 & if\ x[t] < \bar{x} - \alpha\sigma \\ 1 & if\ x[t] > \bar{x} + \alpha\sigma \\ 0 & otherwise \end{cases} \quad (5.5)$$

5.3.2.2 Mean and Maximum-Minimum (Mean-MaxMin)

This method is used in neuroscience studies to discretize the EEG and fMRI data for effective connectivity modeled by discrete Dynamic Bayesian Networks [4], [19]. The method uses mean $\bar{x}$ with maximum x^h and minimum x^l of time-series x, then performs the following expression.

$$d[t] = \begin{cases} -1 & if\ x[t] < \bar{x} - (\bar{x} - x^l)/3 \\ 1 & if\ x[t] > \bar{x} + (x^h - \bar{x})/3 \\ 0 & otherwise \end{cases} \quad (5.6)$$

The method thresholds the data from its maximum and minimum value by taking the one-third of their difference with mean. This method can be generalized by considering the techniques explained in the binary discretization section: Max-X and Top-X. Hence, two possible new discretization methods for ternary discretization could be proposed.

5.3.2.3 Discretization Based on Max Min (MaxMin-X)

The method starts by finding the average $\bar{x}$, then subtracting each time point from mean to obtain zero-mean time series. After that following expression is performed to discretize the data:

$$d[t] = \begin{cases} -1 & if\ x[t] < (1 - \%X) * x^l \\ 1 & if\ x[t] > (1 - \%X) * x^h \\ 0 & otherwise \end{cases} \quad (5.7)$$

5.3.2.4 Discretization Based on Top and Down (TopDown-X)

In this method, time series are split into three sets by finding two thresholds: one puts %X highest values to first set, the other one puts %X lowest values to the second set, remaining ones to the third set.

5.3.2.5 Discretization Based on Mean and Time (Mean-Time)

In this method, time-series is discretized into three levels by following two steps [54], [55]. First, *x* is discretized into two-level {0,1} by using the mean2 method described in Section 2.1.1. Then this discretized *d'* is re-classified based on the following scheme:

$$d[t] = d'[t] - d'[t-1] \tag{5.8}$$

Data is converted to three discrete levels {-1,0,1} where 'increase or rising' treated as '1', '0' means 'No change or constant', and '-1' means 'decrease or falling'.

5.3.2.6 Method Proposed by Ji and Tan (Ji-Tan)

In this method, discretization is performed by considering the variations between successive time points [56]. Ji and Tan considered that these variations are important and meaningful whenever they exceed a certain threshold. First, they transform a time-series *x* into another series *x'* such that:

$$x'[t] = \begin{cases} \dfrac{x[t]-x[t-1]}{|x[t-1]|} & if\ x[t-1] \neq 0 \\ 1 & if\ x[t-1] = 0\ \wedge x[t] > 0 \\ -1 & f\ x[t-1] = 0\ \wedge x[t] < 0 \\ 0 & f\ x[t-1] = 0\ \wedge x[t] = 0 \end{cases} \tag{5.9}$$

Then the final discretized time series *d* is obtained considering a threshold δ>0:

$$d[t] = \begin{cases} -1 & if\ x'^{[t]} < -\delta \\ 1 & if\ x'[t] > \delta \\ 0 & otherwise \end{cases} \quad (5.10)$$

5.3.3 Multilevel Discretization Methods

In multilevel discretization, the time-series x is transformed in many discrete levels.

5.3.3.1 Equal Width Discretization (EWDX)

In this method, the aim is to divide the data into k intervals using maximum and minimum data values. Interval of the discretization is calculated as $w=(max-min)/k$ where k is the discretization level, and each cut point x_r is calculated as $x_{r+1}=x_r+w$, with x_0 being the minimum data value. After this step, each interval $[x_r\ x_{r+1}]$ is assigned to a discrete level [49].

5.3.3.2 Equal Frequency Discretization (EFDX)

This method aims to divide the data into k intervals, where each interval has the same number of data points. Suppose we have N number of data points, and our purpose is to divide the data into k discrete levels, each discrete level must contain N/k number of data points. After defining the intervals, each data point is assigned to its discrete state [9].

5.3.3.3 K-means discretization (K-means)

This method aims to discretize the data into K level by using a clustering approach [57]. The groups are calculated by maximizing the similarity within the elements of each cluster. K-means clustering uses the squared Euclidian distance as a similarity measure and tries to partition each element in x with the minimum of WCSS (within-cluster sum of squares).

5.3.3.4 Bidirectional K-means discretization (biK-means)

This method is an extended version of K-means clustering [57]. Suppose we have a vector *X: [x_1... x_m ... x_n],* each x_m represents *m*-th time-series and *n* is the number of nodes. The aim is to discretize the data into *k* splits. To do that, two clustering approaches are performed: the first one is the k-means clustering, which is performed for each time-series-x_m with the number of clusters to be *k+1*. Second clustering is performed for every time point of *X, X[t]* where nodes are clustered using k-means with the number of clusters to be *k+1*. Applying two clustering methods gives two discrete states for each $x_m[t]$*:* one for each time-point and the second one for each time-series. Let these two clusters to be d_1 and d_2 respectively, note that $d_1, d_2 \in \{0 \dots k+1\}$. Then final discrete state $d_m[t]$ is determined by the following rule:

$$d_m[t] = p \mid p^2 \leq d_1[t] * d_2[t] < (p+1)^2 \qquad (5.11)$$

Table 5-1: An example of biK-means discretization with k=3, suppose that d_1 and d_2 are found by applying k+1 clustering on X[t] and x_m. The discretization state of the variable $x_m[t]$ is shown for each possible d_1 and d_2.

	d_2			
d_1	*1*	*2*	*3*	*4*
1	1*1=1→$d_m[t]$=1	1*2=2→$d_m[t]$=1	1*3=3→$d_m[t]$=1	1*4=4→$d_m[t]$=2
2	2*1=2→$d_m[t]$=1	2*2=4→$d_m[t]$=2	2*3=6→$d_m[t]$=2	2*4=8→$d_m[t]$=3
3	3*1=3→$d_m[t]$=1	3*2=6→$d_m[t]$=2	3*3=9→$d_m[t]$=3	3*4=12→$d_m[t]$=3
4	4*1=4→$d_m[t]$=2	4*2=8→$d_m[t]$=3	4*3=12→$d_m[t]$=3	4*4=16→$d_m[t]$=3

5.3.4 Properties and External parameters of the methods

Some methods have external parameters that should be set by experts. Table 5- [2-4] lists the properties of the discretization methods. The values of the external parameters are also provided. In addition, the reason for the robustness of each method is explained in the tables.

Most of the methods use the value of a time series for a particular time point t in which the variation between time points is not investigated. However, some methods are designed to discretize the value of *x(t)* by using both *x(t)* and *x(t-1)*. Table 5-[2-4] shows the methods use the variation between time points.

Table 5-2: The properties of binary discretization methods and the values of the external parameters

Name of Method	**The variation between time points**	**External parameter**	**Robustness and Reason**
Mean2	X	-	**Robust**: No external parameters
Mid-Range	X	-	**Robust**: No external parameters
Max-X	X	-	**Not Robust**: A parameter is used for deciding on the discretization threshold. Maximum of the time-series used as the base value for the threshold. The maximum value of a time-series heavily depends on data type and size.
Top-X	X	X=25,50,75	**Robust**: Although there is a parameter to decide for thresholding, the threshold is used to divide the data according to a percentage of the number of samples. The threshold obtained from this parameter does not depend on the properties of data such as the number of samples, the number of nodes. Once the best
TDT	X	-	**Robust**: No external parameters
TSD	✓	-	**Robust**: No external parameters
Extended TSD	✓	α=0.25, 0.50, 1.00, 1.50	**Robust**: Not robust like TSD. But once the best value for α is chosen, it can be used any time; the method does not depend on any constraints such as the number of samples, the number of nodes.

Table 5-3: The properties of ternary discretization methods and the values of the external parameters

Name of Method	**The variation between time points**	**External parameter**	**Robustness and Reason**
Mean-std	X	α=0.25, 0.50, 1.00, 1.50	**Robust**: Once the best value for α is chosen, it can be used any time, the methods do not depend on any constraints such as the number of samples, the number of nodes.
MaxMin-X	X	X=67,50,33	**Not Robust**: The threshold is chosen by the maximum and minimum value of the time-series, which means that the method is data-dependent.
TopDown-X	X	X=10,20,30,40	**Robust**: The decision of the parameter does not depend on data. Once the parameter is chosen, it can be used for data.
Mean-Time	✓	-	**Robust**: No parameters
Ji-Tan	✓	δ=1/3,1/2,2/3	**Robust**: The decision of parameter does not depend on data constraints.

Table 5-4: The properties of multi-level discretization methods and the values of the external parameters

Name of Method	**The variation between time points**	**External parameter**	**Robustness and Reason**
EWDX	X	-	**Not Robust**: The data is discretized according to its maximum and minimum values, which makes this method to be not robust.
EFDX	X	-	**Robust**: No parameters
K-means	X	-	**Robust**: No parameters
biK-means	X	-	**Robust**: No parameters

5.4 The use of derivative for discretization

In this thesis, one of our hypotheses was using variation between time points to have a better performance in the discretization of fMRI data to model effective connectivity by dDBN. To evaluate this hypothesis, we also used the derivative of the generated synthetic fMRI data in the discretization methods. We compared the cases where fMRI data was directly used, or derivative of fMRI data was used in discretization.

The reason behind our hypothesis is the linear property of the hemodynamic response. The brain is as a linear system with impulse response to be the hemodynamic response. Several studies show that BOLD response of the brain is linear if the period of the stimuli is higher than 4 sec for visual stimuli and 6 sec for audio stimuli. Although the period depends on the stimuli type, results signify that for a period greater than a certain threshold, this property holds [58]–[60]. For higher frequencies of the stimuli, the hemodynamic response behaves nonlinearly. Most of the researchers consider this issue when designing an fMRI task in order not to violate the linear property of the HRF.

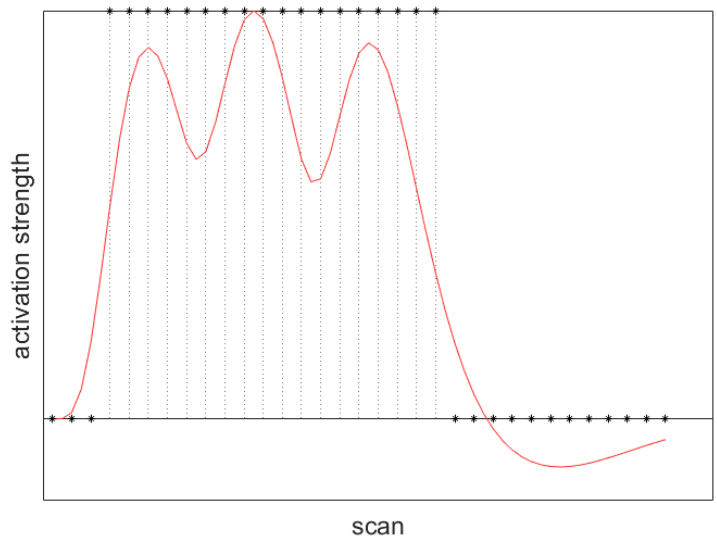

Figure 5-2: Discretization of an HRF response to the stimuli where the time difference between successive stimuli is 6 seconds. The discrete signal is the mean2 discretization of the corresponding signal

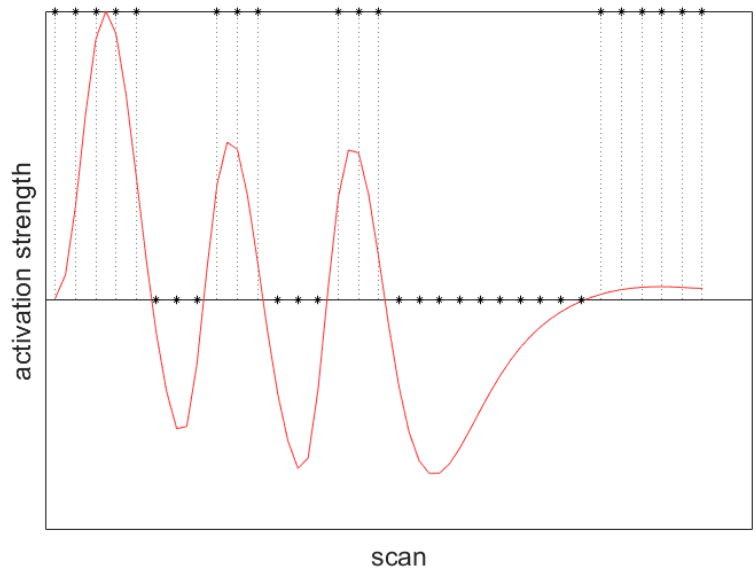

Figure 5-3: Discretization of the derivate of HRF response to the stimuli where the time difference between successive stimuli is 6 seconds. The discrete signal is the mean2 discretization of the corresponding signal

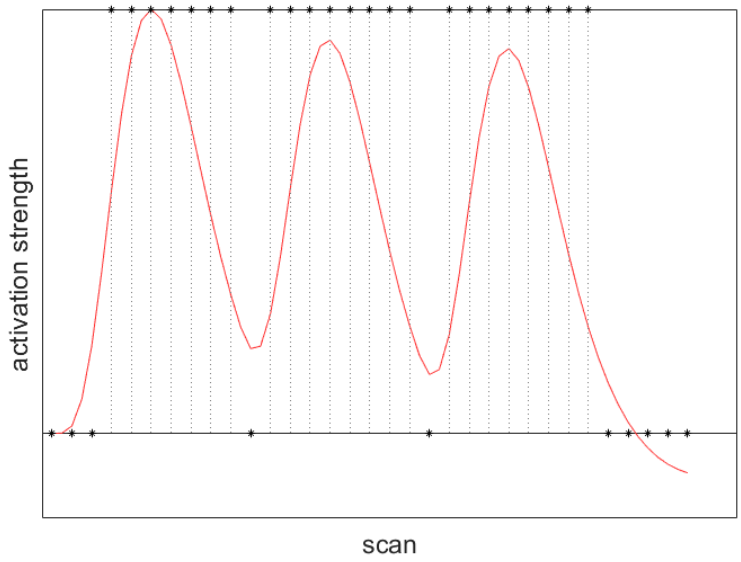

Figure 5-4: Discretization of an HRF response to the stimuli where the time difference between successive stimuli is 8 seconds. The discrete signal is the mean2 discretization of the corresponding signal

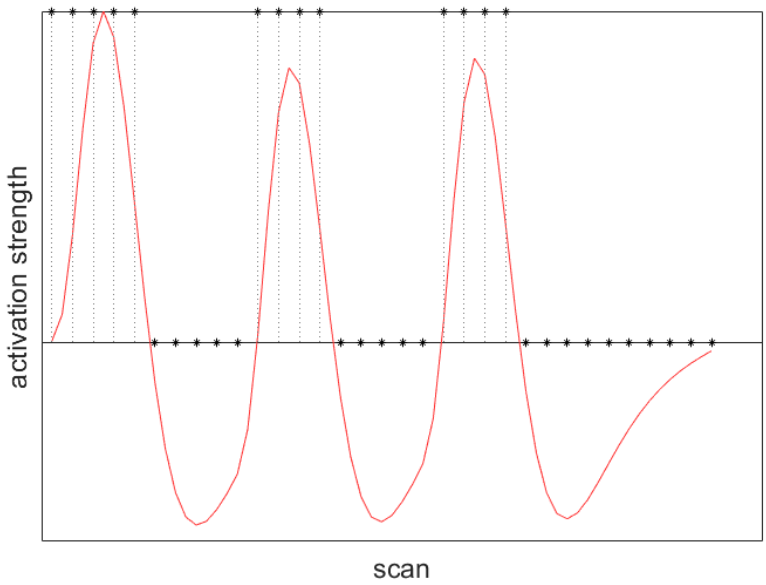

Figure 5-5: Discretization of the derivate of HRF response to the stimuli where the time difference between successive stimuli is 8 seconds. The discrete signal is the mean2 discretization of the corresponding signal

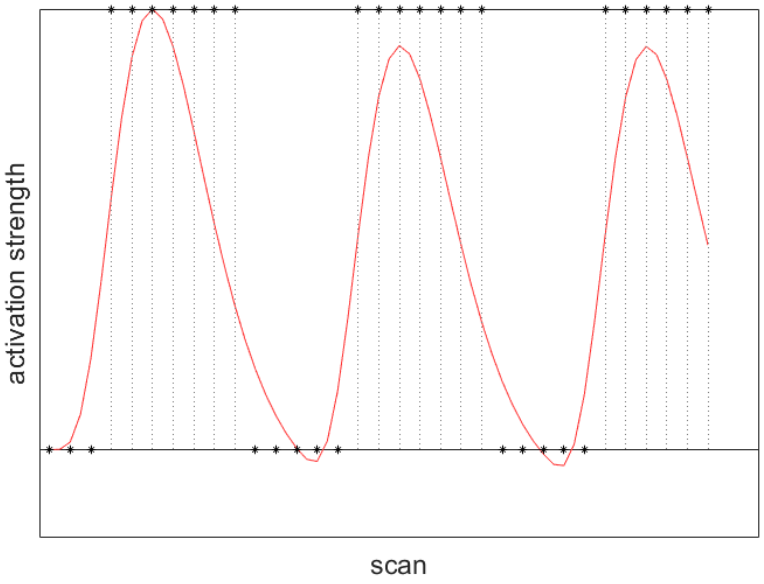

Figure 5-6: Discretization of an HRF response to the stimuli where the time difference between successive stimuli is 8 seconds. The discrete signal is the mean2 discretization of the corresponding signal

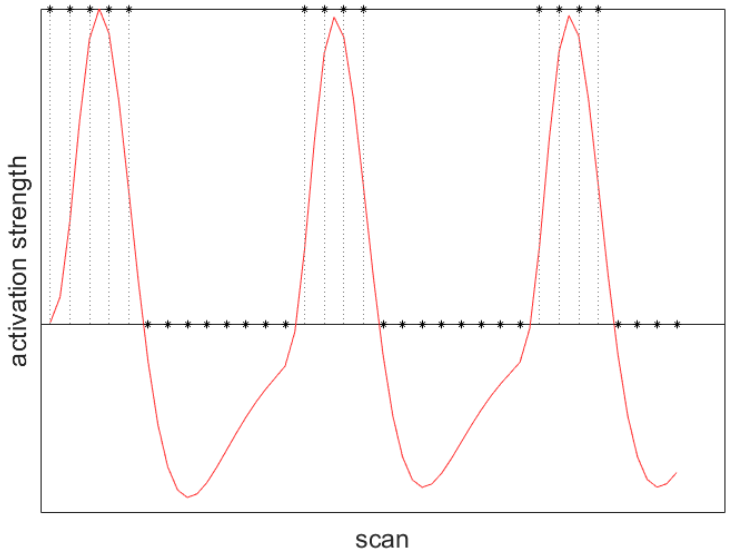

Figure 5-7: Discretization of the derivate of HRF response to the stimuli where the time difference between successive stimuli is 8 seconds. The discrete signal is the mean2 discretization of the corresponding signal

Figure 5-[2-7] illustrates an example of the effect of discretization on HRF response. When the period of the stimuli is 6 seconds the discretization method is not able to discretize the HRF response effectively in order to denote the change of the stimuli. When the period is 8 seconds, discretization starts to sense the stimuli and differentiate them, and when it is 10 seconds, discretization is fully capable of showing each stimulus in the discretized signal. However, when the derivative of the HRF response is discretized the discretization method was able to denote each stimulus regardless of the period of the stimuli. This is due to the linear property of the HRF response. When the time difference between successive stimulus is lower, discretization methods give less information for the HRF response. On the other hand, using variation between time points, derivative in this case, provides much more information, since the information loss due to time difference is omitted.

5.5 Generating Synthetic fMRI data

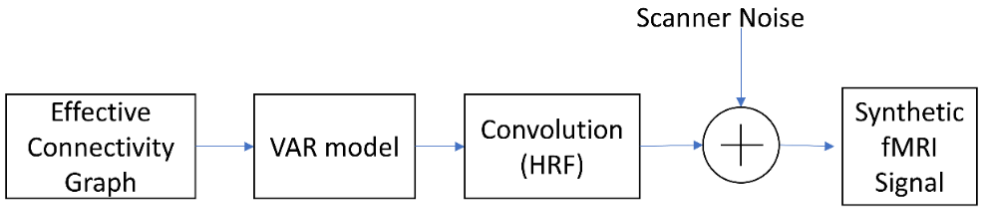

Figure 5-8: Flowchart for generating synthetic fMRI time-series

The vector autoregressive (VAR) model was used to create temporal relationships [4], [6], [15], [61]. Let x be an n-dimensional time series which obeys the first-order VAR model. Every time point of a time series i, x_i is represented by the following expression.

$$x_i(t) = \sum_{j=1}^{n} A_{ji} x_j(t-1) + \in \tag{5.12}$$

In this expression A is a matrix that shows the linear relation between time series. A_{ji} represents the temporal linear relation between i-th and j-th time series.

The dimension of the vector is not critical for the discretization methods. However, larger values would require a larger number of samples to train the dDBN structure because the increase in the model complexity requires many more samples to recover the model through learning [32]. Therefore, we chose the number of time-series to be 6. Hence A was a 6*6 matrix, and there were a total of 36 temporal relations.

Generating the VAR data starts by choosing an appropriate A. Since we would use this data to run the dDBN learning, we firstly generated a new structure A' showing the direction of the temporal relation, which is described as 0's or 1's. A_{ij}'=0 means that there is no edge from *j-th* node to *i-th* node, and 1 means that the *i-th* node is affected by *j-th* node. We had some constraints on A'. The first constraint was that the number of edges was chosen as half of the number of elements that the matrix has. This means that there were a total of 18 connections and 18 non-connections in the generated VAR model. The second constraint was the number of parents for each time series. We had generated the synthetic data such that each node had the same number of parents, which is 3. The parent set of each node was selected randomly. Then, we generated the VAR matrix A by using the generated connectivity matrix A'. On the one hand, 0's in A' remained the same, while 1's in A' were replaced by a random number which is in the interval [-1 -0.5] U [0.5 1]. We did not use smaller numbers which are comparable with 0. Because treating smaller linear relations as a connection between certain nodes might not be correct. After that, we checked the eigenvalues of A, to be sure about the stability of the generated VAR data. Hence the unstable matrices were not used for creating the VAR data. Then A was used to generate the synthetic data by two steps. Firstly, $x_i(0)$ was generated by random from a normal distribution with 0 mean and unity variance. Then the values for other time points were generated by using equation 5.12, where $\in$ is chosen as the Gaussian white noise process of zero mean and unity variance. The number of samples was selected to be 5000. 1000 such multivariate time series were generated to get a good statistical comparison. The ground-truth matrix A' was saved to be used later in the performance analysis.

At the next step, the VAR data were convolved with hemodynamic response function (HRF) to obtain the synthetic fMRI time-series. The HRF was canonical which is a mixture of two Gamma functions. SPM8 TOOLBOX already has a built-in function called *spm_hrf* which generates the HRF data in the discrete domain. We used TR=2s for the sampling of HRF. Note that previous studies also added scanner noise to the generated synthetic fMRI data. They aimed to see the effect of the scanner noise on learning [4], [6]. In this chapter, we also evaluated the effect of scanner noise. However, we only analyzed this effect on the best 11 methods to decrease the computation time.

5.6 Results and Discussion

dDBN learning procedure was applied to the synthetic data, and the best structure found by the learning procedure was saved. Then the following evaluation metrics were calculated. Note that each metric was calculated as an average of 1000 synthetic data.

True positives TP: The average number of correct interactions inferred i.e the connections were the same for the inferred structure and the ground-truth structure.

False Positive (FP): The average number of incorrect interactions inferred. The ground-truth structure did not possess an edge, but the inferred structure did.

True Negative (TN): The average number of correct non-interactions inferred i.e both the ground-truth structure and inferred structure did not possess an edge

False Negative (FN): The average number of incorrect non-interactions inferred. The ground-truth structure possessed an edge, but the inferred structure did not.

Recall or True Positive Rate (TPR): TP/(TP+FN)

False Positive Rate (FPR): FP/(FP+TN)

Precision or Positive Predictive Value (PPV): TP/(TP+FP)

Accuracy: (TP+TN)/(Total)

Before presenting the results, some important issues should be clarified. First, when DBN learning fails to recover the correct model, the expected value for TP, FP, TN, and FN would be same, meaning that we would get 9 for each of these metrics; and the expected value for Recall, FPR, Precision, and Accuracy would be 0.5. Secondly, when the learning procedure fully recovers the correct structure that data was sampled from, which means perfect learning, the expected value for TP and TN would be 18, and for FP and FN, it would be 0. Because there was a total of 36 edges in the ground-truth structure, and each structure had an equal number of dependencies and independencies. The *A'* matrix was designed such that there were 18 1's and 18 0's. Also, Recall, Precision, and Accuracy would be 1 and FPR would be 0 for this case. Hence for an easy way of deciding on the goodness of the discretization methods, we had decided to use the following criteria. If the discretization method had an accuracy higher than 0.75, it was denoted as successful.

Table 5-5:The comparison of binary discretization methods

Name	TP	FP	TN	FN	Recall	FPR	Precision	Accuracy
TSD	15,64	6,63	11,37	2,36	0,87	0,37	0,70	**0,75**
midRange	14,47	7,52	10,48	3,53	0,80	0,42	0,66	0,69
top50	14,46	7,53	10,47	3,54	0,80	0,42	0,66	0,69
mean2	14,45	7,53	10,47	3,55	0,80	0,42	0,66	0,69
TDT	14,47	7,66	10,34	3,53	0,80	0,43	0,65	0,69
TSD25	14,86	8,24	9,76	3,14	0,83	0,46	0,64	0,68
TSD50	13,66	8,69	9,31	4,34	0,76	0,48	0,61	0,64
top25	14,25	9,65	8,35	3,75	0,79	0,54	0,60	0,63
top75	14,20	9,68	8,32	3,80	0,79	0,54	0,59	0,63
max75	13,84	9,80	8,20	4,16	0,77	0,54	0,59	0,61
TSD100	11,15	8,21	9,79	6,85	0,62	0,46	0,58	0,58
max50	9,54	7,10	10,90	8,46	0,53	0,39	0,57	0,57
TSD150	9,63	7,74	10,26	8,37	0,54	0,43	0,55	0,55
max25	5,40	4,12	13,88	12,60	0,30	0,23	0,57	0,54

Table 5-6: The accuracy comparison of the binary discretization methods using the time-series and its derivative

	mean2	midRange	max25	max50	max75	top25	top50	top75	TDT
time-series	0,69	0,69	0,54	0,57	0,61	0,63	0,69	0,63	0,69
derivative	**0,75**	**0,75**	0,55	0,59	0,65	0,67	**0,75**	0,67	**0,75**

Table 5-5 shows the evaluation metrics for each binary discretization method by sorting the methods according to their accuracy. The external parameters are given with the proposed method's name. For example, top25 means that Top-X method was applied for discretization and 25 was used for the external parameter X. The same applies to Max-X method also. The Extended TSD is expressed by TSD only, and the parameter is added to the right side of the name. For instance, TSD150 shows that the Extended TSD method is used with the external parameter to be 1.50.

Each method has an accuracy of more than 0.5. Therefore, although some methods are less accurate than others, at least they were capable of finding some correct dependencies. However, most of the methods were not able to find the correct structure effectively, and their accuracy is near 0.5. Binary discretization is not a perfect approach to discretize a continuous time-series, because expressing continuous signal by just only two levels reduces the information that data possess. For most of the methods, although they have different aspects of discretization, they were classified as unsuccessful. Nonetheless, Transitional State Discretization was the best among binary discretization methods, and 0.75 accuracy is achieved by this method. Note that this method differs from other methods expressed in table 5-5 because this method uses the variation between time points to discretize the data. In order to understand the effect of the variation between time points on discretization, Table 5-6 gives the comparison of accuracy for each method using the time-series and derivative of it. The most important result of this table is for all methods, using the derivative of the time-series gave a better performance than using time-series itself. Therefore, the hypothesis proposed in this study is strengthened by the results presented in table 5-6. In addition, the accuracy of Max-X method was the lowest comparing with others in Tables 5-5 and 5-6, note that this method was described as a non-robust method in table 5-2. Robustness is an important property for a method, and we conclude from this result that non-robust methods give a lower performance.

Table 5-7: The comparison of ternary discretization methods

Name	TP	FP	TN	FN	Recall	FPR	Precision	Accuracy
ji_tan33	13,67	2,92	15,09	4,33	0,76	0,16	0,82	**0,80**
ji_tan50	13,31	3,33	14,67	4,69	0,74	0,19	0,80	**0,78**
ji_tan67	12,80	3,70	14,30	5,20	0,71	0,21	0,78	**0,75**
mean_std50	12,46	4,54	13,46	5,54	0,69	0,25	0,73	0,72
mean_std25	11,96	4,05	13,95	6,04	0,66	0,23	0,75	0,72
topdown40	11,96	4,07	13,93	6,04	0,66	0,23	0,75	0,72
topdown30	12,49	4,07	13,40	5,51	0,69	0,26	0,73	0,72
topdown20	12,34	5,24	12,76	5,66	0,69	0,29	0,70	0,70
mean_std100	12,07	5,47	12,53	5,93	0,67	0,30	0,69	0,68
maxmin67	11,70	5,54	12,46	6,31	0,65	0,31	0,68	0,67
topdown10	11,41	5,54	12,46	6,59	0,63	0,31	0,67	0,66
mean_std150	10,71	5,38	12,62	7,29	0,60	0,30	0,67	0,65
mean_time	11,16	6,28	11,73	6,84	0,62	0,35	0,64	0,64
maxmin50	9,70	5,04	12,96	8,30	0,54	0,28	0,66	0,63
maxmin33	7,45	3,87	14,13	10,55	0,41	0,22	0,66	0,60

Table 5-8: The accuracy comparison of the ternary discretization methods using the time-series and its derivative

	mean_std25	mean_std50	mean_std100	mean_std150	maxmin67	maxmin50
time-series	0,72	0,72	0,68	0,65	0,67	0,63
derivative	**0,84**	**0,84**	**0,78**	0,71	**0,75**	0,68

	topdown10	topdown20	topdown30	topdown40	maxmin33
time-series	0,66	0,70	0,72	0,72	0,60
derivative	0,74	**0,80**	**0,84**	**0,84**	0,63

Table 5-7 shows the evaluation metrics for each ternary discretization method by sorting the methods according to their accuracy. For each method, if an external parameter was used, the parameter was expressed by the right side of its name. Also,

table 5-8 gives the comparison of accuracy for each method using the time-series and derivative of it. We discuss the results of ternary discretization methods in three headings. First of all, ternary discretization methods gave more accurate results than binary discretization methods. It is intuitively correct because splitting a time-series to more levels reduces the information loss due to discretization. Secondly, robust methods explained in table 5-3 performed better; the accuracy of the following methods are better than others despite changing external parameters: Ji-Tan, mean-std, and Top-Down. Robustness is one of the most important criteria for a method because these types of methods are less dependent on the properties of data such as sample size, maximum and minimum values of the data; these properties may change for different types of experiments and conditions. Thirdly similar to the result discussed for binary discretization methods, the methods use variation between time points is better than other methods. Although different external parameters were used for the Ji-Tan method, this method was better compared with the methods not using variation between time points. One exception is the mean-time discretization; less accuracy is obtained despite it uses the variation between time points. Besides, Table 5-8 signifies that using derivate of the time-series gives better discretization performance; for all methods using the derivative of the time-series outperformed the time-series itself. Therefore, like in binary discretization methods, ternary discretization methods also corrected the hypothesis proposed; variation between time points reduces the information loss due to discretization for fMRI data.

Table 5-9: The comparison of multi-level discretization methods

Name	Level	TP	FP	TN	FN	Recall	FPR	Precision	Accuracy
EFD3	3	12,40	4,39	13,61	5,60	0,69	0,24	0,74	0,72
3means	3	12,51	4,80	13,20	5,49	0,70	0,27	0,72	0,71
EFD2	2	14,47	7,52	10,48	3,53	0,80	0,42	0,66	0,69
2means	2	14,44	7,55	10,46	3,56	0,80	0,42	0,66	0,69
EWD2	2	14,40	7,79	10,21	3,60	0,80	0,43	0,65	0,68
EWD3	3	11,79	5,73	12,27	6,21	0,66	0,32	0,67	0,67
bi2means	2	10,64	7,81	10,19	7,37	0,59	0,43	0,58	0,58
bi3means	3	5,78	3,84	14,16	12,22	0,32	0,21	0,60	0,55

Table 5-10: The accuracy comparison of the multi-level discretization methods using the time-series and its derivative

	EWD2	EWD3	EFD2	EFD3	2means	3means	bi2means	bi3means
time-series	0,68	0,67	0,69	0,72	0,69	0,71	0,58	0,55
derivative	0,73	0,74	**0,75**	**0,85**	**0,75**	**0,83**	0,61	0,58

Table 5-9 gives the comparison of the multi-level discretization methods, for each method two levels were compared, binary and ternary. Because, increasing the level of the discretization requires larger samples for the convergence of dDBN learning. Hence, we did not increase the level of discretization, which is not practical because sample sizes are limited for fMRI data. There are several results obtained from multi-level discretization analysis. First of all, except for the biKmeans and EWDX, ternary discretization was better than binary discretization. Getting better accuracy for ternary discretization was expected. The reason for the biKmeans gave the inverse of the expectation is the following; this method makes two Kmeans clustering. The first one for each time point, where nodes are clustered; the number of nodes was six in our case. Secondly, each time-series is clustered, same as the Kmeans method. For the first clustering, we only had six nodes, clustering six

samples to three clusters may arise problems, and these could lead to lower discretization performance for ternary. Clustering six samples to two clusters would give better quality then clustering to three clusters. Hence binary discretization gives better than ternary for this method. The second result is about robustness. Even though the biKmeans method was described as a robust method in Table 5-4, robust methods gave a better performance than non-robust methods. EWDX method is the only non-robust method for multi-level discretization; its performance was lower than other methods except biKmeans. Robustness could be the reason for its ternary discretization to have lower accuracy than binary. More importantly, its ternary discretization performed lower than binary discretization of EFDX and Kmeans discretization. Therefore, robust methods perform better fMRI discretization for dDBN learning. The third result is about the effect of derivative on the discretization. Like binary and ternary discretization methods, using the derivative of time-series gave better accuracy than using time-series itself for all methods presented for multi-level discretization shown in Table 5-10.

Table 5-11: The list of best ten discretization methods according to their accuracy. "der" means that firstly the derivative of the synthetic data was computed then discretization methods were applied.

Name	level	TP	FP	TN	FN	Recall TPR	FPR	Precision PPV	Accuracy
der + EFD3	3	14,95	2,45	15,55	3,05	0,83	0,14	0,86	**0,8473**
der + topdown40	3	14,59	2,23	15,77	3,41	0,81	0,12	0,87	**0,8432**
der + mean_std25	3	14,58	2,23	15,77	3,43	0,81	0,12	0,87	**0,8429**
der + mean_std50	3	14,94	2,60	15,40	3,06	0,83	0,14	0,85	**0,8427**
der + topdown30	3	14,89	2,66	15,34	3,11	0,83	0,15	0,85	**0,8397**
der + 3means	3	14,77	2,92	15,08	3,23	0,82	0,16	0,83	**0,8291**
ji_tan33	3	13,67	2,92	15,09	4,33	0,76	0,16	0,82	**0,7988**
der + topdown20	3	14,28	3,53	14,47	3,72	0,79	0,20	0,80	**0,7985**
der + mean_std100	3	13,92	3,88	14,12	4,08	0,77	0,22	0,78	**0,7789**
ji_tan50	3	13,31	3,33	14,67	4,69	0,74	0,19	0,80	**0,7772**

Table 5-11 lists the best ten discretization methods. The overall comparison for every method is provided in the Appendix C. The best methods listed in the table have three discrete levels. Binary discretization did not give higher accuracy, which is expected; because information loss due to discretization is lower for the ternary case. More importantly, all methods in the table use variation between time points; either method uses it directly like the Ji-Tan method, or methods use the derivative of the time series, then performs the discretization. This result concludes that the hypothesis proposed in this study is corrected by simulation methods; using variation between time points increases the performance of discretization; and for modeling brain connectivity by dDBN using fMRI data, the derivative of the fMRI signal is more informative than the signal itself. In addition, robustness is a key property for a method to give a higher performance for discretization; listed methods in table 5-11 were classified as a robust method in tables 5-[2-4].

Table 5-12: Effect of scanner noise on the accuracy of the discretization methods

Name		standard deviation σ							
	Level	**0**	**0,2**	**0,4**	**0,6**	**0,7**	**0,8**	**0,9**	**1**
TSD00	2	0,750	0,739	0,704	0,663	0,647	0,633	0,621	0,612
ji+tan33	3	0,799	0,766	0,709	0,663	0,642	0,627	0,610	0,597
ji+tan50	3	0,777	0,752	0,705	0,661	0,646	0,626	0,610	0,599
mean_std25	3	0,720	0,707	0,681	0,653	**0,643**	0,633	0,624	0,618
der+mean_std25	3	0,843	0,793	0,714	0,662	**0,638**	0,623	0,606	0,593
mean_std50	3	0,720	0,709	0,685	0,660	0,647	**0,639**	0,626	0,619
der+mean_std50	3	0,843	0,810	0,729	0,669	0,647	**0,628**	0,611	0,599
mean_std100	3	0,683	0,675	0,660	0,640	0,631	0,621	**0,615**	0,609
der+mean_std100	3	0,779	0,755	0,703	0,656	0,635	0,622	**0,605**	0,593
topdown20	3	0,697	0,689	0,669	0,647	0,636	**0,628**	0,619	0,612
der+topdown20	3	0,799	0,775	0,716	0,662	0,642	**0,624**	0,609	0,595
topdown30	3	0,719	0,708	0,684	0,659	0,647	**0,639**	0,626	0,619
der+topdown30	3	0,840	0,809	0,731	0,670	0,647	**0,629**	0,612	0,598
topdown40	3	0,719	0,707	0,681	0,654	**0,642**	0,632	0,624	0,618
der+topdown40	3	0,843	0,793	0,714	0,662	**0,639**	0,622	0,607	0,593
EFD3	3	0,722	0,712	0,687	0,659	0,646	**0,639**	0,627	0,621
der+EFD3	3	0,847	0,810	0,728	0,670	0,646	**0,628**	0,610	0,598
3means	3	0,714	0,704	0,682	0,658	0,646	0,636	**0,625**	0,617
der+3means	3	0,829	0,802	0,729	0,669	0,646	0,636	**0,612**	0,599

Table 5-12 gives the effect of scanner noise on the discretization. Rather than evaluating all discretization methods explained in section 2, we only analyzed the best ten methods listed in table 5-11 and TSD which was the best among all binary discretization methods. Two main results are obtained by evaluating the effect of scanner noise. First, for all methods increasing the scanner noise decreases the accuracy of the discretization methods. This is expected because an increase in the noise level increases the information loss on the HRF while discretizing the data. Secondly, variation between time points is more sensitive to the scanner noise. When scanner noise increases further a threshold, using the time-series itself outperforms

using variation between time points. For example, EFD3 with derivative was obtained the best method among other techniques, see table 5-11. When the standard deviation of the scanner noise is higher than 0.7, der+EFD3 is less accurate than EFD3. The standard deviation for which using time-series itself was more informative is illustrated by bolding the corresponding accuracy of the methods in Table 5-12. The reason behind this behavior is the increase in the effect of scanner noise when performing the difference between successive points. Suppose we have w[t] which is a white noise with 0 mean σ standard deviation. When we perform a difference filter on this noise, we get a new white noise with 0 mean and $\sigma\sqrt{2}$ standard deviation. Therefore, despite using variation between time points gives better accuracy when we consider the linear property of HRF response, an increase in the scanner noise decreases the accuracy. Hence, we have a trade-off between using variation between time-points and increasing the effect of scanner noise.

5.7 Testing Discretization Methods in Real fMRI Data

We compared discretization methods using only the data belonging to the control group in openfMRI data. The corresponding ROI's and time series generation are explained in chapter 7. In this section, our aim is only to show that the discretization methods explained in this section are able to discretize the real fMRI data. Since we do not have ground-truth connectivity in this comparison, we made the comparison using EFD3 with derivative, which was chosen as the best discretization method for synthetic fMRI data. We made a comparison using a total of 8 different methods. According to the use of a method, fMRI data was discretized, considering also the derivative of data if a method needs it. We obtained a connectivity graph for each discretization method separately for a total of 121 subjects. The connectivity graph of der-EFD3 was accepted as ground-truth. If this method is considered to be the best method, it can be taken as ground truth. We have posed our hypothesis as follows, if we obtain a ranking similar to the ranking obtained as a result of tests made with synthetic data for real fMRI data, this will show the accuracy of the

methods performed. Therefore, the results of the methods compared with der-EFD3 should show a similar result to those made in synthetic data. It is not important to get the same result exactly, but the similarity should be as high as possible.

Table 5-13: The accuracy obtained by taking the der-EFD3 ground-truth for the methods specified for the real fMRI data in the table on the left, the results on the right show the accuracy for the synthetic data of the same methods. In both tables, the results are presented by sorting them according to accuracy.

method	accuracy according to real fMRI data
der-topdown40	0,942
der-mean-std50	0,935
der-3means	0,887
jitan33	0,872
TSD00	0,753
der-mean2	0,752
der-EWD3	0,720
topdown40	0,674
EFD3	0,668
mean-std50	0,657
3means	0,641
EWD3	0,615
mean2	0,549

method	accuracy according to synthetic data
der - topdown40	0,843
der - mean_std50	0,843
der - 3means	0,829
ji_tan33	0,799
TSD00	0,750
der - mean2	0,750
der - EWD3	0,745
EFD3	0,722
mean_std50	0,720
topdown40	0,719
3means	0,714
mean2	0,692
EWD3	0,668

Table 5-13 shows the comparison of the methods for real and synthetic fMRI data. Firstly, considering the rankings of the results of a total of 13 methods, a very similar ranking can be seen for real fMRI data cases compared to synthetic data. In both cases, the first six methods found to be the same. Other methods showed very close rankings compared to synthetic sequences. No striking difference was observed for both data. Note that the faster decline in performance rates on real data may be the result of the der-EFD3 method being the ground-truth.

We will discuss this high level of similarity under two headings. First and foremost, testing of the methods performed in this study was carried out correctly. It shows how successful the method of generating synthetic fMRI data specified in this study is. This result signifies that it is a perfect method for generating synthetic fMRI data that possess effective connectivity among its regions. If we consider dDBN modeling as an evaluation metric, one of the results of table 5-3 is how high is the similarity between synthetic fMRI data and actual fMRI data. Second, these results show that how critical is the use of discretization techniques for connectivity modeling brain via dDBN. This issue is not considered for the recent studies conducted effective connectivity [4]–[6]. The choice of their method is not denoted as successful. Burge et al. and Dang et al. used the EWD method where it was denoted as a not successful not robust method for fMRI data. Rajapakse et al. used MaxMin33 in their studies and it was also denoted as an unsuccessful method. This makes their premise of study to be very problematic. All in all, I believe that this chapter of the thesis is very critical for modeling brain connectivity by dDBN. The EFD3 with derivative is a powerful technique to discretize fMRI data.

CHAPTER 6

EFFECT OF SMOOTHING ON EFFECTIVE CONNECTIVITY

In this chapter, our aim is to investigate the effect of smoothing on fMRI data by considering the effective connectivity modeling of the brain using dDBN. In spatial smoothing, data points are averaged using the neighbor information. This has the effect of low pass filtering by removing the high frequencies of data. Therefore, sharp edges of the data are removed, and it is blurred. The standard method for smoothing fMRI data is filtering the data with a Gaussian function on the spatial coordinate. The standard deviation (sigma) of the Gaussian function is the only parameter used.

6.1 Advantages of the smoothing

Smoothing of the fMRI data comprises several advantages for analyzing fMRI data. The most critical advantage is that it increases the signal to noise ratio [62]–[65]. Scanner noise has a negative impact on fMRI data analysis. In chapter 5, we showed how this noise may affect the performance of the discretization. By spatially smoothing the data, the fMRI signal is averaged for a particular voxel by considering its neighbor's information. Therefore, the signal to noise ratio for the voxel signal is improved.

6.2 The impact on effective connectivity

In this part of the study, 4 ROIs of Default Mode Network were used. Firstly, smoothing with various sigma values (1-12 mm) was applied to fMRI data which was downloaded from openfMRI.org. Then a time series for each ROI was extracted from the smoothed data. The length of the time series was appropriate for a model

of 4 nodes (as explained in chapter 4, table 4-4). Data was discretized using der-EFD3 which found to be the best method in chapter 5. Finally, connectivity was modeled by dDBN for each subject. The effective connectivity models learned are a 4 * 4 matrix, and connectivity is indicated by 1 and dis-connectivity by 0. Then, the average connectivity map of the control group was obtained for each smoothing sigma.

Table 6-1: Average connectivity map for smoothing sigma 1 mm

ROIs	MPFC	PCC	LIPL	RIPL
MPFC	1	0	0	0
PCC	0	1	0	0
LIPL	0	0	1	0
RIPL	0	0	0	1

Table 6-2: Average connectivity map for smoothing sigma 5 mm

ROIs	MPFC	PCC	LIPL	RIPL
MPFC	0,579	0,273	0,240	0,281
PCC	0,628	0,430	0,455	0,355
LIPL	0,322	0,438	0,479	0,314
RIPL	0,248	0,372	0,298	0,306

Table 6-3: Average connectivity map for smoothing sigma 10 mm

ROIs	MPFC	PCC	LIPL	RIPL
MPFC	0,826	0,727	0,736	0,661
PCC	0,719	0,686	0,777	0,678
LIPL	0,818	0,694	0,686	0,694
RIPL	0,554	0,727	0,645	0,678

Table 6-[1-3] show the average connectivity map found for different smoothing sigma values. When sigma is low, ROIs are only self-connected. The fact that the connections between different ROIs are 0 indicates that there is no connection for all participants. When the sigma is 5mm, the method drops the self-connections and other connections appear. Statistically, this result shows us how important the smoothing is for the connectivity analysis. For smaller sigma, the connections are self-connected; the data cannot be statistically related to other ROIs. A single time-series data at time t depends only on its value at time t-1, which is expected for any time-series. However, when the smoothing sigma increases, new connections are formed which is also expected. However when sigma increases there is a decrease in the self connections. For example the avarage self-connections represented in table 6-2 is smaller than ones represented in table 6-1. The decrease in the self connections is not expected. Because a single time-series would be always expected to be connected to itself. Observation of decrease in self-connections and increase on other-connections shows that other-connections found by increasing the smoothing sigma are significant. Table 6-3 shows the connectivity map for the smoothing sigma at higher values. First of all, it is clear that the average connectivity map is high; all of the entries are generally higher than 0.5. The reason for this is that the smoothing with very high sigma caused the similarity of the voxels in the ROI voxels to be very high. As a result, the dDBN learning algorithm is beginning to see connectivity due to the repetition of same data distributions, which is caused by oversmoothing and

that should be avoided. These three tables show that the smoothing of fMRI data is critical for the connectivity map found by dDBN.

6.3 Determination of Smoothing Parameter for fMRI data considering dDBN

In order to find the best smoothing sigma, we considered the issues discussed above. The self connections of ROIs and connections between different ROIs were carefully analyzed. Table 6-4 supports our previous statements about the effect of sigma on the effective connectivity graph. The connectivity between different ROIs increases as the smoothing sigma rises. But the average self-connection falls first and this decline continues until the sigma is 4 mm. It is almost stable when sigma is up to 7 mm and then rises again. The first drop is related to finding dependency with other ROIs. When the dependency is much stronger with different ROI, the frequency of observing self-connection reduces. This is actually in line with our purpose in smoothing. This decrease remains constant between 4-7 mm and later increases again. The increase afterward is a problem that arises due to the excessive smoothing of the data. Considering these results, it can be seen that the smoothing sigma should be between 4-7 mm, and according to table 6-4, we suggest to take sigma as 4 mm because it is the sigma that gives the lowest average self-connections.

Table 6-4: Average self- connections and connections between different ROIs for different smoothing sigma

smoothing sigma	self-connection	Connections between different ROIs
1	1,000	0,000
2	0,936	0,030
3	0,597	0,164
4	0,444	0,270
5	0,448	0,352
6	0,475	0,425
7	0,486	0,501
8	0,552	0,568
9	0,626	0,632
10	0,719	0,702
11	0,781	0,789
12	0,855	0,844

In order to strengthen the applicability of the proposed sigma value, we analyzed the connectivity differences between control and schizophrenia. The hypothesis is as follows, if there is a significant connectivity difference between these two groups, the differences would be observed at optimal smoothing sigma. Table 6-5 presents the average connectivity map differences between these groups for various sigma values with corresponding probabilities. The result shown in table 6-5 supports the method proposed for determining the best smoothing sigma. In the previous result, we determine the best sigma as 4-7 mm. When we look at the difference values in Table 6-5, we see the highest difference between the two classes are observed when sigma is 3-6 mm. Note that the p-value between 3-6 mm is less than 0.05, which shows statistically significant differences. In other words, the best sigma we find with the dDBN method gives us the difference between the groups as highest and

statistically meaningful. This result confirms our previous decision. The best smoothing sigma can be found using the control data only, just by analyzing the behavior of connectivity changes while sigma is changed. But including the comparison between the control and schizophrenia group provides further proof about the decision.

Table 6-5: The average connectivity difference between the schizophrenia and control groups. Corresponding probability values that show the probability of getting the same difference in the control group using Monte Carlo simulation, for $p<0.05$ the corresponding p values are bolded.

smoothing sigma	difference between schizophrenia and control	p
1	0,000	1,0000
2	0,499	0,2201
3	1,989	**0,0003**
4	1,927	**0,0052**
5	1,800	**0,0303**
6	1,986	**0,0120**
7	1,686	0,1049
8	1,311	0,5154
9	1,079	0,7957
10	1,136	0,6115
11	1,024	0,6070
12	0,867	0,6888

In addition, how to find the p values in Table 6-5 will be explained. It would not be correct to interpret this table without determining whether there is a significant difference between the control and schizophrenia group. In order to investigate the statistical meaning of the difference, we took the control group as a basis. Namely, since there were 50 schizophrenes and 121 controls in total, we had $50/171 \sim 0.3$

percent of schizophrenic data. Whether the average connectivity graph difference we found between these two groups was significant or not was decided using the control group's data. Here we used the Monte Carlo simulation technique. According to this technique, we divided our control group's data into two parts at a rate of 0.3 each time. Then we found the average connectivity difference between these parts. We did this simulation 100,000 times and recorded the results. We compared the difference between schizophrenia and the control group with the difference values we obtained with these simulations. In total, we calculated how many simulations gave difference larger than the real difference and divide it by the number of simulations. This gave us the probability of finding the actual difference on control group's data. If this value is less than 0.05, it was accepted that there is a statistically significant difference. These differences are shown in bold in the table. Figure 6-1 gives an example of how the Monte Carlo simulation technique is done, the smoothing sigma is 4 mm for this figure. The histogram represents the difference of the average connectivity graph when the Monte Carlo simulation technique was applied to the control group. The red area on the figure illustrates the corresponding observation of the difference in the control group which is higher than the actual difference between control and schizophrenic subjects.

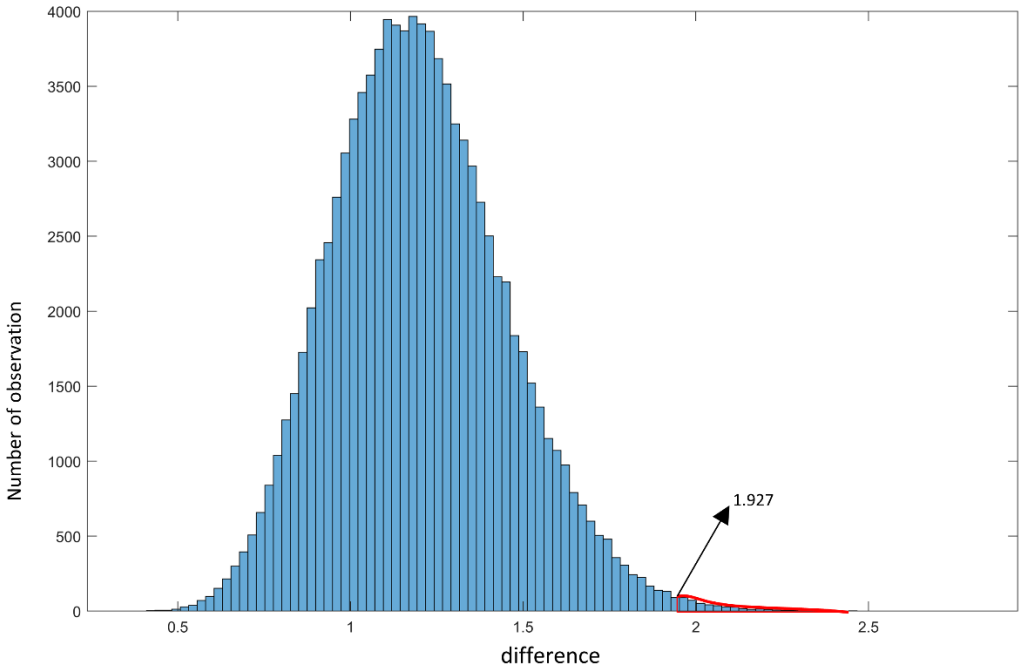

Figure 6-1: Histogram of the difference using Monte Carlo simulation and the corresponding real difference between control and schizophrenia group

6.4 Discussion

All in all, we have concluded from the analysis of this chapter that smoothing has a significant effect on the connectivity graph found by dDBN. Note that in the previous chapter we had noted that scanner noise has a significant effect on model discovery for simulated fMRI data. In order to decrease the scanner noise, smoothing is the only option for fMRI data. Smoothing is performed by using a kernel filter with a Gaussian function and a predetermined standard deviation in mm. Increasing sigma blurs images too much that effective connectivity becomes fully connected. Decreasing the smoothing sigma does not decrease the scanner noise effect hence optimal sigma is needed. We used the internal properties of dDBN to determine the best sigma for smoothing. Our results suggest that a 4 mm sigma is favorable to use in Gaussian smoothing. This smoothing also gave the best discrimination between control and schizophrenia groups. We did not have to find such a result because our

method did not rely on differentiating two groups. However, if there were significant differences between these two groups, we would expect to get this difference for optimal sigma, and we got it for the smoothing sigma we proposed in this study.

CHAPTER 7

EFFECTIVE CONNECTIVITY FOR CONTROL AND SCHIZOPHRENIA SUBJECTS USING THREE DIFFERENT MODELLING APPROACHES

In this section, we will explain how the effective connectivity modeling of the brain is done using resting-state fMRI data. Resting-state fMRI data is obtained when participants do not take any task during fMRI scanning. While the individuals are at rest, activation is observed for certain brain regions, and some connectivity patterns are found between the brain regions. One of the most important connectivity observed during resting state is the Default Mode Network (DMN). In previous studies, both functional connectivity and effective connectivity between brain regions of Default Mode Network have been studied in modeling the brain using resting-state fMRI [3], [18], [66], [67]. In this section, we examined the effective connectivity of the brain using the control and schizophrenic data to underline the differences between each group using default mode network regions. From the previous studies, it has been stated that 4 important regions are found for the default mode network of the brain [66], [67]. Table 7-1 shows these brain regions and their MNI coordinates.

Table 7-1: Corresponding DMN regions and their MNI coordinates

Regions of DMN	MNI Coordinates
Medial Prefrontal Cortex (MPFC)	3,54,-2
posterior cingulate cortex (PCC)	0, -52, 26
left inferior parietal lobule (LIPL)	−50, −63, 32
right inferior parietal lobule (RIPL)	48, -69, 35

OpenfMRI data includes 121 controls, 50 schizophrenia, 49 bipolar disorders and 41 attention-deficit / hyperactivity disorder (ADHD) data. More information about the data and its preprocessing is explained in [68], [69]. The biggest advantages of using this data are: Firstly, it was obtained from a large number of participants, which will increase the reliance on statistical analysis. Secondly, the preprocess of this data had been prepared in openfMRI.org. We would only do modeling without any preprocessing (except smoothing and high-pass filtering). In this thesis, we had only conducted effective connectivity modeling for schizophrenia and control groups.

7.1 Preprocessing

Since data is already preprocessed we downloaded the data from openfMRI.org with its preprocessed version. Data were preprocessed by the following steps: Motion Correction, Slice-Timing Correction, Distortion Correction and Spatial Normalization. Since this data is not spatially smoothed we applied this step by using FSL toolbox. This data is smoothed by the Gaussian kernel with a 4 mm standard deviation. Note that 4 mm was found as the best smoothing sigma in chapter 6. Then data were high-pass filtered by using FSL which contain a high pass filter for removing low-frequency component in the time domain. This component is rather depending on the scanner not related to hemodynamic response. The cut-off frequency was 128 HZ.

7.2 Data generation for ROIs

Several studies have conducted effective connectivity analysis for fMRI data [2], [3], [5], [17], [70]. We have used one of the ROIs that they have found activated in resting-state fMRI data which was collected from healthy subjects. These regions are illustrated in table 7-1, where these regions are the regions considered to be related to the default mode network.

After fMRI data were smoothed, it had to be discretized in order to use it for dDBN modeling. For this, fMRI data was discretized with der-EFD3 which was chosen to be the best in chapter 5. The size of the fMRI time series was 152 scans. After this process, the size became 151. Since we made discretization at 3 levels, cardinality (K) should be taken as 3. Since the maximum number of parents is 4 ROIs, we have taken this number as 4. We did not put any restrictions on the number of parents. As a result, the table 4-4 stated that the amount of data required should be greater than 3819, and we assumed to have more than 4000 number of samples. The size of each voxel was 151 in each discrete-fMRI data. As a result, we have to put the 4000 / 151≈27 voxels' signal in a row by concatenating them. Totally we collected 4*4077 time-series for each participant where 4 denotes the regions and 4077 indicates the temporal length of each regions.

7.3 Effective connectivity approaches

In the literature, there are totally 3 different approaches for brain modeling [71]. In this section, we briefly explain each method, and their results using openfMRI data.

7.3.1 Individual Structure (IS) Approach

The IS approach learns individual networks for each subject separately and makes group analysis on these separate networks. The IS approach definitely considers variability between subjects. However, when individual network models are different, it is not a trivial task to obtain a statistically significant network for a group.

In order to apply this approach for openfMRI data, we examined the following steps. Firstly, each discrete ROI-based fMRI data of each subject was used in dDBN learning to find the model of the data. This model which is a 4*4 matrix consists of 0s and 1s was the effective connectivity graph of each subject. 1s in this graph represent the temporal causality between ROI's and 0s indicate dysconnectivity among the brain regions. Secondly, in order to understand the group level similarity

and differences between the groups, the average effective connectivity graph for each group was calculated which is illustrated in Tables 7-2 and 7-3 for control and schizophrenia.

Table 7-2: Average connectivity graph of the control group. The connections are from rows to columns

ROIs	MPFC	PCC	LIPL	RIPL
MPFC	0,562	0,207	0,182	0,124
PCC	0,562	0,421	0,339	0,231
LIPL	0,248	0,347	0,488	0,306
RIPL	0,174	0,306	0,215	0,306

Table 7-3: Average connectivity graph of the schizophrenia group. The connections are from rows to columns

ROIs	MPFC	PCC	LIPL	RIPL
MPFC	0,860	0,200	0,140	0,180
PCC	0,360	0,700	0,380	0,280
LIPL	0,180	0,260	0,640	0,140
RIPL	0,180	0,220	0,100	0,580

Pearson chi-square ($\chi 2$) test was performed to see if the difference between these two groups is statistically significant. Table 7-4 and figure 7-1 give the corresponding frequency differences between control and schizophrenia groups and corresponding p values for each connection.

Table 7-4: Connectivity differences between schizophrenics and controls obtained by the individual-structure method where the Pearson chi-square test is applied to see the significance of the difference. Green shows for $p<0.01$, red shows for $0.01<p<0.05$ and bolded ones are for $0.05<p<0.1$.

connection	control	schizophrenia	difference	$\chi 2$	p
MPFC->MPFC	0,562	0,860	0,298	13,796	0,000204
PCC->MPFC	0,562	0,360	-0,202	5,774	0,016266
LIPL->MPFC	0,248	0,180	-0,068	0,927	0,335529
RIPL->MPFC	0,174	0,180	0,006	0,010	0,919699
MPFC->PCC	0,207	0,200	-0,007	0,010	0,922352
PCC->PCC	0,421	0,700	0,279	10,978	0,000922
LIPL->PCC	0,347	0,260	-0,087	1,230	0,267332
RIPL->PCC	0,306	0,220	-0,086	1,290	0,256136
MPFC->LIPL	0,182	0,140	-0,042	0,439	0,507444
PCC->LIPL	0,339	0,380	0,041	0,263	0,60798
LIPL->LIPL	0,488	0,640	0,152	3,300	**0,069262**
RIPL->LIPL	0,215	0,100	-0,115	3,146	**0,076126**
MPFC->RIPL	0,124	0,180	0,056	0,921	0,337295
PCC->RIPL	0,231	0,280	0,049	0,451	0,5019
LIPL->RIPL	0,306	0,140	-0,166	5,088	0,024086
RIPL->RIPL	0,306	0,580	0,274	11,225	0,000807

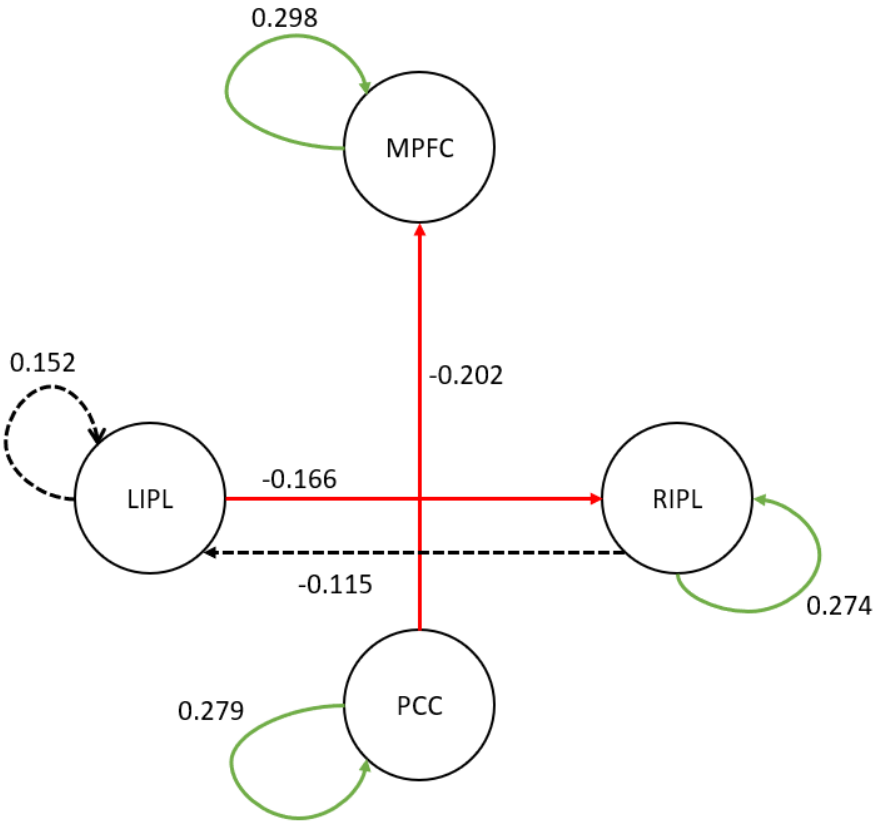

Figure 7-1: The connectivity map for the individual structure approach, only statistically significant connections are illustrated. The green arrows show for $p<0.01$, the red arrows show for $0.01<p<0.05$ and dashed arrows show for $0.05<p<0.1$.

7.3.2 Virtual-Typical Subject (VTS) Approach

The VTS approach assumes that each subject within the group has the same brain network. Data from all individuals are combined and processed as if sampled from a virtual object. Accordingly, a single time series for the groups was obtained by concatenating the data for every subject in each group. Then dDBN effective connectivity models were obtained. In Table 7-5 and 7-6, effective connectivity models of the two groups are given. Different observed connections in both groups are indicated in red. Some connections do not exist in the schizophrenia group.

Table 7-5: The effective connectivity model of the control group using Virtual-Typical Subject approach

ROIs	MPFC	PCC	LIPL	RIPL
MPFC	1	0	0	0
PCC	1	1	1	1
LIPL	**1**	1	1	**1**
RIPL	1	1	**1**	1

Table 7-6: The effective connectivity model of the schizophrenia group using Virtual-Typical Subject approach

ROIs	MPFC	PCC	LIPL	RIPL
MPFC	1	0	0	0
PCC	1	1	1	1
LIPL	**0**	1	1	**0**
RIPL	1	1	**0**	1

One of the most striking results in these tables is about the effectiveness of MPFC. MPFC does not affect any brain region in the default mode network for both groups MPFC (medial-prefrontal cortex) is the region where the decision-making mechanism is located in the frontal area of the brain and where human-kind processes are performed. Considering that it is the region where the information is collected, processed, and the decision is made, this result may be expected because it must be the region where information transformation is ended considering the default mode network regions. Figure 7-2 gives a figure for the easiness of illustration to see the difference between each group.

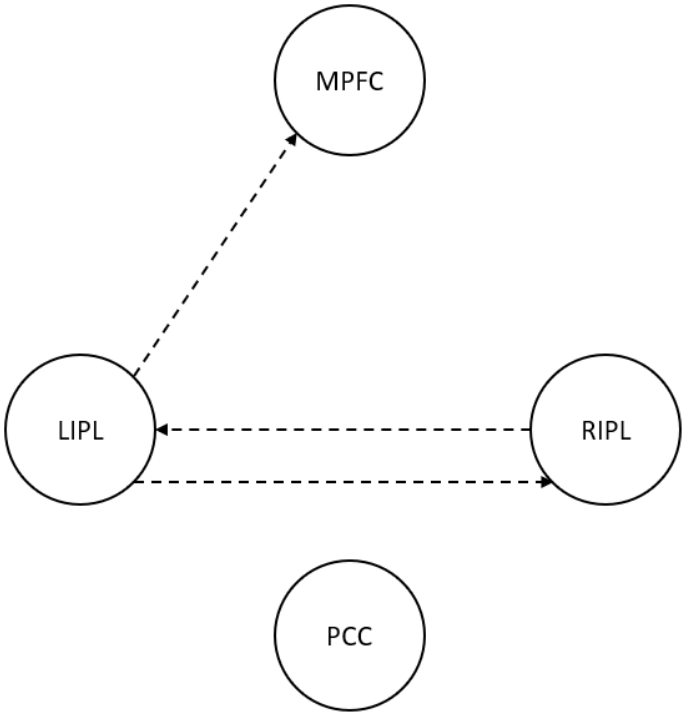

Figure 7-2: The connectivity map for virtually-typical subject approach, only the differences are illustrated. Note that lines in the figure are the connections observed for the control group but not observed for schizophrenia.

7.3.3 Common Structure (CS) Approach

The CS approach allows a common brain network within the group while allowing model parameters to differ among subjects. The CS approach addresses group similarity at the structural level and cross-subject variability at the parameter level. The strength of the connection was considered, and it was assumed that it may differ among the subject within the group. We modeled this method with dDBN by the following steps. First of all, the data for the two groups were combined to obtain connectivity maps with dDBN learning. Tables 7-5 and 7-6 are the graphs obtained for two groups. The connections in these two maps were combined to create a single connectivity map. In this new connectivity map, the connection seen in any group is taken as the connection. If we explain mathematically, the connectivity map of the

two groups goes through the 'OR' operation. Using the discrete fMRI data of the subjects in both groups and the connectivity map, we calculated the strength of the connection between the brain regions of the groups. The strength of connection shows the information transmitted between the two regions. In order to calculate the strength of the connection following expression was used.

$$LS(X \to Y) = \sum_{x,z} P(x,z) \sum_{y} P(y|x,z) \log_2 \frac{P(y|x,z)}{P(y|z)} \qquad (7.1)$$

In this expression, the connectivity strength between Y and X is calculated, where Y, X are discrete-valued random variables. Z represents the random variables in the parent set of Y except for X. This expression finds the information transformed from X to Y. For detail about the strength of the connection see Nicholson and Jitnah [72].

Table 7-7: The average effective connectivity strength of the control group

ROIs	MPFC	PCC	LIPL	RIPL
MPFC	0,082	0,000	0,000	0,000
PCC	0,067	0,043	0,031	0,027
LIPL	0,054	0,033	0,039	0,027
RIPL	0,054	0,031	0,026	0,032

Table 7-8: The average effective connectivity strength of the schizophrenia group

ROIs	MPFC	PCC	LIPL	RIPL
MPFC	0,117	0,000	0,000	0,000
PCC	0,058	0,068	0,029	0,027
LIPL	0,050	0,030	0,049	0,025
RIPL	0,052	0,031	0,022	0,051

Tables 7-7 and 7-8 show the average connectivity strength of the effective connectivity in the control and schizophrenia group. One of the most striking and

remarkable results in this table is that the connectivity to MPFC is dramatically higher than the connectivity to other regions. This can also be seen in schizophrenic subjects. This result is also compatible with the individual structure approach. In order to examine the differences between the two groups, statistical analyses were performed using the two-sample t-test.

Table 7-9: Connectivity differences between schizophrenics and controls obtained by the common-structure approach where a two-sample t-test was applied to see the significance of the difference. Green shows for $p<0.01$, red shows for $0.01<p<0.05$.

connection	control	schizophrenia	difference	p
MPFC->MPFC	0,082	0,117	42	0,00002
PCC->MPFC	0,067	0,058	-14	0,00613
LIPL->MPFC	0,054	0,050	-8	0,04949
RIPL->MPFC	0,054	0,052	-3	0,37631
MPFC->PCC	0,000	0,000	-	-
PCC->PCC	0,043	0,068	60	0,00121
LIPL->PCC	0,033	0,030	-9	0,26947
RIPL->PCC	0,031	0,031	0	0,99835
MPFC->LIPL	0,000	0,000	-	-
PCC->LIPL	0,031	0,029	-4	0,65731
LIPL->LIPL	0,039	0,049	26	0,03166
RIPL->LIPL	0,026	0,022	-14	0,03828
MPFC->RIPL	0,000	0,000	-	-
PCC->RIPL	0,027	0,027	1	0,88367
LIPL->RIPL	0,027	0,025	-8	0,25059
RIPL->RIPL	0,032	0,051	60	0,00003

Table 7-9 shows the comparison of the connectivity strength between the two groups. One of the results is that the average connectivity strength is lower in the schizophrenic group for the connections between different ROIs. On the contrary,

the self-connections of ROIs are higher in the schizophrenic group. The differences are stated as the percentage increase in the connectivity strength of the schizophrenia group compared to the control group. Note that almost the same result was obtained for the individual structure approach.

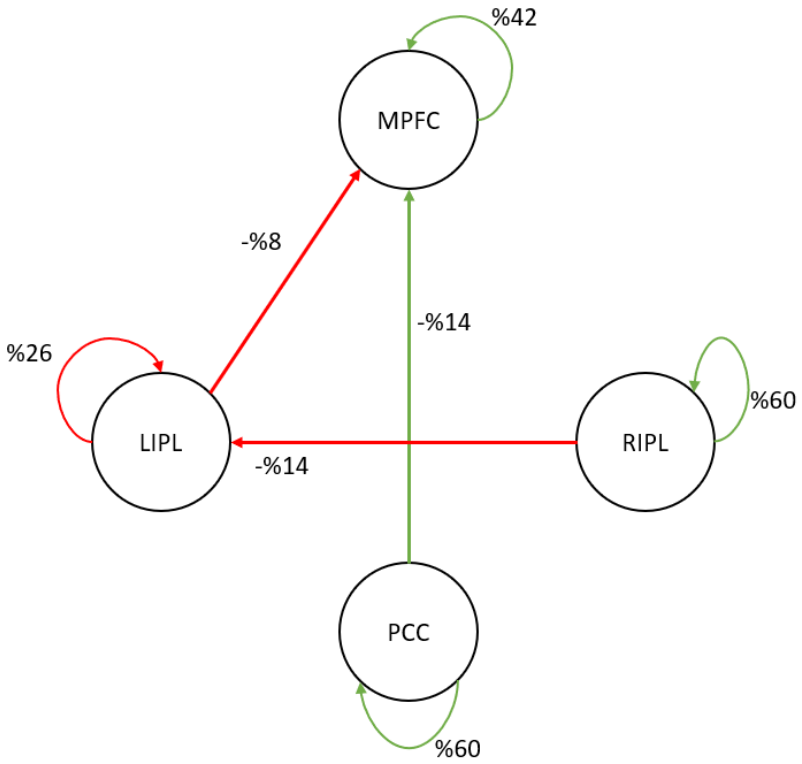

Figure 7-3: The connectivity map for the common structure approach only statistically significant connections are illustrated. Green arrows show for $p<0.01$, the red arrows show for $0.01<p<0.05$.

7.4 Discussion

In this chapter, we examined the effective connectivity of control and schizophrenia using fMRI data. The ROIs were 4 brain regions of the Default Mode Network presented in table 7-1. Effective connectivity shows the temporal effects of these regions on each other. These analyses were made considering three approaches.

These are individual structure approach, virtual typical subject approach and common structure approach. We will discuss the results carried out by considering these approaches in two headings.

Firstly, the three approaches provided similar results. Figure 7-1 and 7-3 give the difference of the effective connectivity between the two groups. If we look at these differences, we see a decrease in effective connectivity in the schizophrenia group between different ROIs. In the IS and CS approach, almost similar connections were observed as statistically different between the two groups. Figure 7-2 (VTS approach) also shows similar differences, but self-connection was observed in both groups. The reason for this is that the method is handled only on the presence and absence of the connection. But the connections that are seen differently in VTS can be observed also in IS and CS. It is a very important result that this similarity is so high. Because this result is a proof that the issues encountered in modeling the brain effective connectivity by dDBN are completely resolved. Thereby, with this thesis, dDBN can be used for any fMRI data, regardless of the problems.

Secondly, it was observed that in the schizophrenia group, effective connectivity decreases compared to the control group. Especially when all groups were modeled by the VTS approach, some connections were not observed for the schizophrenia group although they exist for the control group. Considering the results in the IS approach and CS approach, this situation makes itself more conspicuous. A drop in the strength of connectivity between different ROIs is observed for almost all connections.

CHAPTER 8

CONCLUSION

In this thesis, we examined the important issues that should be considered in modeling the effective connectivity of the brain with dDBN. These important issues were examined in the finest detail and what should be considered in each issue was determined. There are totally three issues, one is to determine the required number of samples for the convergence of the dDBN modeling, the second one is the evaluation of the discretization methods for fMRI data, the last one is determining the most suitable sigma for smoothing. Each issue in this study confirms each other. For example, the sample complexity analysis obtained in chapter 4 was used to compare discretization methods in chapter 5. If the results found in the sample complexity analysis were wrong, there would be no compatible results in chapter 5. This also applies to the smoothing step. The smoothing issue was examined using real data which was processed by considering both sample complexity results in chapter 4 and discretization results in chapter 5. Finally, the connectivity approaches in chapter 7 was made considering all three issues. In chapter 7 the three different approaches for effective connectivity gave very similar results which shows that the three issues identified in this study have been resolved in a proper and reliable way.

In order to find the sample complexity for the convergence of the dDBN structure learning, theoretical approaches were investigated. It was observed that the minimum required number of samples found by theoretical approach is practically very high. Therefore, a practical and systematic approach was investigated and practical sample complexity of dDBN was found. Experiments showed that the sample complexity of structure learning for dDBN is $O(K^{p+1})$. Here K is the cardinality of the network and p is the maximum number of parents present in the network. The experimental results showed that the imaginary sample size is very

critical on the learned model. Less number of samples may be needed with the optimum imaginary sample size, but this issue was left as future work.

Secondly, discretization, an important step for dDBN, was examined. In the literature, generally recommended discretization methods were evaluated for fMRI and the most suitable discretization method was determined for dDBN. While doing this, the properties of fMRI data were considered, and more successful results were obtained when variation between time points (derivative) of fMRI data was used instead of the data itself. The results in Chapter 5 showed that higher performance had always been achieved when the derivative of the data was used. The results of experiments with synthetic data suggest that der-EFD3 discretization is the best method for fMRI discretization. In the same section, experiments with real fMRI data were mentioned. Experiments with real fMRI data were consistent with the results of synthetic data. The extremely high similarity between the results of synthetic and real fMRI data showed how successful the synthetic data generation was.

Chapter 6 covers the issue for the smoothing of fMRI data. Spatial smoothing is the filtering of fMRI data with gaussian function. The objective to be achieved in this chapter was the standard deviation (sigma) of the Gaussian function. For this, data belonging to the control group of openfMRI data was used. This data has been preprocessed for different smoothing sigma values and optimal sigma determined based on the result of the dDBN models. In addition, when we examined the specified smoothing sigma for finding the difference between the control and schizophrenia groups, we found that the optimal sigma distinguished the two groups very successfully. This was further proof of the optimal sigma value. Results signify that smoothing fMRI data with 4 mm gives more accurate results in modeling effective connectivity by dDBN. Consider chapter 5, we had presented the comparison of real and synthetic fMRI data for discretization methods in Table 5-13. It was noted that there is prominent consistency between real and synthetic fMRI data in terms of discretization techniques. This consistency in the ranking also shows

us a result of the smoothing method. Table 5-13, which is the analysis for smoothing 4 mm, also shows the success of the smoothing study (chapter 6).

Finally, using openfMRI data, we examined the effective connectivity between the 4 brain regions belonging to the default mode network. We did statistical tests to determine whether there is a difference in the effective connectivity models of schizophrenia and the control group. The most noticeable result was a decrease in effective connectivity among different ROIs in the schizophrenia group. In other words, in schizophrenic individuals, brain regions affect each other less than healthy individuals. These results were observed similarly for all three different connectivity approaches which are the individual structure, virtual typical subject and common structure approaches.

In this thesis, we left several studies as future work. Foremost, the optimal imaginary sample size may be defined as a function of network parameters. The results in this study signify that a smaller number of samples are adequate to discover the model correctly, the imaginary sample size is now extremely high or low. Hence an optimum value would yield better learning in terms of sample complexity. Secondly, the effective connectivity analysis of bipolar and ADHD was not investigated in this study. The same procedure may also be applied to reveal the brain working mechanism of the corresponding disease. Thirdly, this method is ready to be compared with other effective connectivity models. Since all issues are solved by this study, and there is a practical method to generate synthetic fMRI data which is detailly explained, DCM, Granger Causality etc., may be used to be compared with dDBN.

REFERRENCES

[1] K. J. Friston, A. P. Holmes, K. J. Worsley, J. -P Poline, C. D. Frith, and R. S. J. Frackowiak, "Statistical parametric maps in functional imaging: A general linear approach," *Hum. Brain Mapp.*, 1994.

[2] K. J. Friston, "Functional and effective connectivity in neuroimaging: A synthesis," *Hum. Brain Mapp.*, vol. 2, no. 1–2, pp. 56–78, Jan. 1994.

[3] X. Wu, X. Wen, J. Li, and L. Yao, "A new dynamic Bayesian network approach for determining effective connectivity from fMRI data," *Neural Comput. Appl.*, 2014.

[4] J. C. Rajapakse and J. Zhou, "Learning effective brain connectivity with dynamic Bayesian networks," *Neuroimage*, 2007.

[5] J. Burge, T. Lane, H. Link, S. Qiu, and V. P. Clark, "Discrete dynamic bayesian network analysis of fMRI data," *Hum. Brain Mapp.*, 2009.

[6] S. Dang, S. Chaudhury, B. Lall, and P. K. Roy, "The dynamic programming high-order Dynamic Bayesian Networks learning for identifying effective connectivity in human brain from fMRI," *J. Neurosci. Methods*, 2017.

[7] C. A. Gallo, J. A. Carballido, and I. Ponzoni, "Discovering time-lagged rules from microarray data using gene profile classifiers," *BMC Bioinformatics*, 2011.

[8] Y. Li, T. Jann, and P. Vera-Licona, "Benchmarking time-series data discretization on inference methods," *Bioinformatics*, 2019.

[9] S. C. Madeira and A. L. Oliveira, "An Evaluation of Discretization Methods for Non-Supervised Analysis of Time-Series Gene Expression Data INESC-ID Technical Report 42 / 2005," 2005.

[10] O. Sporns, "Connectome," *Scholarpedia*, vol. 5, no. 2, pp. 55–84, 2010.

[11] O. Sporns, G. Tononi, and R. Kötter, "The human connectome: A structural description of the human brain," *PLoS Computational Biology*. 2005.

[12] H. Johansen-Berg and M. F. S. Rushworth, "Using Diffusion Imaging to Study Human Connectional Anatomy," *Annu. Rev. Neurosci.*, 2009.

[13] B. Horwitz, "The elusive concept of brain connectivity," *Neuroimage*, 2003.

[14] S. L. Bressler and A. K. Seth, "Wiener-Granger Causality: A well established methodology," *NeuroImage*. 2011.

[15] C. Zou and J. Feng, "Granger causality vs. dynamic Bayesian network inference: A comparative study," *BMC Bioinformatics*, 2009.

[16] A. J. Storkey, E. Simonotto, H. Whalley, S. Lawrie, L. Murray, and D. McGonigle, "Learning structural equation models for fMRI," in *Advances in Neural Information Processing Systems*, 2007.

[17] K. J. Friston, L. Harrison, and W. Penny, "Dynamic causal modelling," *Neuroimage*, vol. 19, no. 4, pp. 1273–1302, Aug. 2003.

[18] K. J. Friston, J. Kahan, B. Biswal, and A. Razi, "A DCM for resting state fMRI," *Neuroimage*, 2014.

[19] A. Y. Mutlu and S. Aviyente, "Inferring effective connectivity in the brain from EEG time series using dynamic Bayesian networks," *Proc. 31st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. Eng. Futur. Biomed. EMBC 2009*, pp. 4739–4742, 2009.

[20] D. . G. M. Friedman N.; Geiger, "Bayesian Network Classiers," *Mach. Learn.*, 1997.

[21] K. P. Murphy and B.A., "Dynamic Bayesian Networks: Representation, Inference and Learning," 2002.

[22] G. F. Cooper and E. Herskovits, "A Bayesian Method for the Induction of Probabilistic Networks from Data," *Mach. Learn.*, 1992.

[23] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data," *Mach. Learn.*, 1995.

[24] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. 2009.

[25] A. Carvalho, "Scoring functions for learning bayesian networks," *Inesc-id Tec. Rep*, 2009.

[26] K.-U. Höffgen, "Learning and robust learning of product distributions," pp. 77–83, 2004.

[27] N. Friedman and Z. Yakhini, "On the sample complexity of learning bayesian networks," in *Proceedngs of the Twelfth international conference on Uncertainty in artificial intelligence.*, 1996.

[28] P. Abbeel and A. Y. Ng, "Learning Factor Graphs in Polynomial Time and Sample Complexity," *Jmlr*, 2006.

[29] S. Dasgupta, "The Sample Complexity of Learning Fixed-Structure Bayesian Networks," *Mach. Learn.*, 1997.

[30] O. Zuk, S. Margel, and E. Domany, "On the Number of Samples Needed to Learn the Correct Structure of a Bayesian Network," in *Proceedings of the Twenty-Second Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, 2006.

[31] A. Ghoshal and J. Honorio, "Information-theoretic limits of Bayesian network structure learning," pp. 1–21, 2016.

[32] H. Dai, K. Korb, C. Wallace, and X. Wu, "A study of causal discovery with weak links and small samples," in *IJCAI International Joint Conference on Artificial Intelligence*, 1997.

[33] E. Brenner and D. Sontag, "SparsityBoost: A New Scoring Function for Learning Bayesian Network Structure," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.

[34] H. Steck and T. S. Jaakkola, "On the dirichlet prior and Bayesian regularization," in *Advances in Neural Information Processing Systems*, 2003.

[35] T. Silander, P. Kontkanen, and P. Myllymäki, "On sensitivity of the MAP

Bayesian network structure to the equivalent sample size parameter," in *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence, UAI 2007*, 2007.

[36] H. Steck, "Learning the Bayesian Network Structure: Dirichlet Prior versus Data," 2012.

[37] M. Ueno, "Learning networks determined by the ratio of prior and data," in *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, UAI 2010*, 2010.

[38] M. Ueno, "Robust learning Bayesian networks for prior belief," in *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, UAI 2011*, 2011.

[39] M. Scutari, "An Empirical-Bayes Score for Discrete Bayesian Networks," vol. 52, no. 1, pp. 438–448, 2016.

[40] M. Scutari, "Beyond Uniform Priors in Bayesian Network Structure Learning," 2017.

[41] M. Richeldi and M. Rossotto, "Class-driven statistical discretization of continuous attributes (Extended abstract)," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1995.

[42] B. S. Chlebus and S. H. Nguyen, "On finding optimal discretizations for two attributes," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1998.

[43] G. Bernot, J. P. Comet, A. Richard, M. Chaves, J. L. Gouzé, and F. Dayan, "Modeling and analysis of gene regulatory networks," in *Modeling in Computational Biology and Biomedicine: A Multidisciplinary Endeavor*, 2014.

[44] Salvador Garcıa, Julian Luengo, Jose Antonio Saez, Victoria Lopez, and

Francisco Herrera, “A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning,” *IEEE Trans. Knowl. Data Eng.*, 2013.

[45] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[46] J. Catlett, “On changing continuous attributes into ordered discrete attribute,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1991.

[47] D. Husmeier, “Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks,” *Bioinformatics*, 2003.

[48] J. Sandoval and G. Hernández, “Learning of natural trading strategies on foreign exchange high-frequency market data using dynamic bayesian networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014.

[49] C. A. Gallo, R. L. Cecchini, J. A. Carballido, S. Micheletto, and I. Ponzoni, “Discretization of gene expression data revised,” *Brief. Bioinform.*, 2016.

[50] P. A. Bandettini, “Neuronal or hemodynamic? Grappling with the functional MRI signal,” *Brain connectivity*. 2014.

[51] N. K. Logothetis, J. Pauls, M. Augath, T. Trinath, and A. Oeltermann, “Neurophysiological investigation of the basis of the fMRI signal,” *Nature*, 2001.

[52] C. S. Moller-Levet, K. H. Cho, and O. Wolkenhauer, “Microarray data clustering based on temporal variation: FCV with TSD preclustering.,” *Appl. Bioinformatics.*, 2003.

[53] S. Erdal, O. Ozturk, D. Armbruster, H. Ferhatosmanoglu, and W. C. Ray, “A

time series analysis of microarray data," in *Proceedings - Fourth IEEE Symposium on Bioinformatics and Bioengineering, BIBE 2004*, 2004.

[54] I. Ponzoni, F. J. Azuaje, J. C. Augusto, and D. H. Glass, "Inferring adaptive regulation thresholds And association rules from gene expression data through combinatorial optimization learning," in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2007.

[55] L. A. Soinov, M. A. Krestyaninova, and A. Brazma, "Towards reconstruction of gene networks from expression data by supervised learning.," *Genome Biol.*, 2003.

[56] L. Ji and K. L. Tan, "Mining gene expression data for positive and negative co-regulated gene clusters," *Bioinformatics*, 2004.

[57] Y. Li *et al.*, "Comparative study of discretization methods of microarray data for inferring transcriptional regulatory networks," *BMC Bioinformatics*, 2010.

[58] F. Kruggel and D. Y. Von Cramon, "Temporal properties of the hemodynamic response in functional MRI," *Hum. Brain Mapp.*, 1999.

[59] R. B. Buxton, K. Uludağ, D. J. Dubowitz, and T. T. Liu, "Modeling the hemodynamic response to brain activation," in *NeuroImage*, 2004.

[60] A. L. Vazquez and D. C. Noll, "Nonlinear aspects of the BOLD response in functional MRI," *Neuroimage*, 1998.

[61] A. Roebroeck, E. Formisano, and R. Goebel, "Mapping directed influence over the brain using Granger causality and fMRI," *Neuroimage*, 2005.

[62] J. M. Maisog and J. Chmielowska, "An efficient method for correcting the edge artifact due to smoothing," *Hum. Brain Mapp.*, 1998.

[63] J. B. Poline, K. J. Worsley, A. C. Evans, and K. J. Friston, "Combining spatial extent and peak intensity to test for activations in functional imaging," *Neuroimage*, 1997.

[64] T. White, D. O'Leary, V. Magnotta, S. Arndt, M. Flaum, and N. C. Andreasen, "Anatomic and functional variability: The effects of filter size in group fMRI

data analysis," *Neuroimage*, 2001.

[65] A. Scouten, X. Papademetris, and R. T. Constable, "Spatial resolution, signal-to-noise ratio, and smoothing in multi-subject functional MRI studies," *Neuroimage*, 2006.

[66] X. Di and B. B. Biswal, "Identifying the default mode network structure using dynamic causal modeling on resting-state functional magnetic resonance imaging," *Neuroimage*, 2014.

[67] M. G. Sharaev, V. V. Zavyalova, V. L. Ushakov, S. I. Kartashov, and B. M. Velichkovsky, "Effective connectivity within the default mode network: Dynamic causal modeling of resting-state fMRI data," *Front. Hum. Neurosci.*, 2016.

[68] R. A. Poldrack *et al.*, "A phenome-wide examination of neural and cognitive function," *Sci. Data*, 2016.

[69] K. J. Gorgolewski, J. Durnez, and R. A. Poldrack, "Preprocessed Consortium for Neuropsychiatric Phenomics dataset," *F1000Research*, 2017.

[70] R. Goebel, A. Roebroeck, D. S. Kim, and E. Formisano, "Investigating directed cortical interactions in time-resolved fMRI data using vector autoregressive modeling and Granger causality mapping," *Magn. Reson. Imaging*, 2003.

[71] J. Li, Z. J. Wang, S. J. Palmer, and M. J. McKeown, "Dynamic Bayesian network modeling of fMRI: A comparison of group-analysis methods," *Neuroimage*, vol. 41, no. 2, pp. 398–407, Jun. 2008.

[72] A. E. Nicholson and N. Jitnah, "Using mutual information to determine relevance in Bayesian networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1998.

APPENDICES

A. Proof of taking the first expression for equation 4.2

$$\sigma = \min\left\{\frac{\gamma^n}{2}, \frac{IC_B}{2^{n+2}|nlog\left(\frac{\gamma}{2}\right)+1|}\right\} \tag{8.1}$$

$$IC_B = \min_{i,j}\left\{\min_{s\subset\{x_1,\dots,x_n\}\setminus\{x_i,x_j\}}\{I_{P_{B*}}(x_i,x_j|S)\}\right\} \tag{8.2}$$

$$I_{P_B}(x_i,x_j|S) = -\log\left(P(x_i,x_j|S)\right) \tag{8.3}$$

In this expression γ is the minimum conditional probability distribution in P_{B*}, and IC_B is the minimum information content in P_{B*} and eq. 8.3 defines information content on a probability distribution. The aim is to find the minimum of the σ value when $\gamma \rightarrow 0$. We started by the following condition. Assume there exists a random variable pair (x_i,x_j) such that γ is the minimum conditional distribution in P_{B*} for x_i and x_j.

$$\gamma = \min_{i,j}\{P_{B*}(x_i|S\cup\{x_j\})\} \tag{8.4}$$

Similarly, we found the expression for IC_B in terms of γ. In order to get a minimum value for IC_B, the highest probability of $P(x_i,x_j|S)$ should be considered; see eq. 8.3, and note that a probability of a random variable is in the interval [0 1].

$$P(x_i,x_j|S) = \max\{P(x_i|S\cup\{x_j\}) * P(x_j|S)\} \tag{8.5}$$

To be able to get the highest joint probability over x_i and x_j given S, two expressions on the right-hand side of the eq. 8.5 should be maximum. For the second multiplicand $P(x_j)$ we assumed it to be 1, since we did not have a priori information for the marginal distributions in P_{B*}. If the first expression in eq. 8.1 was found to be

minimum for σ, then the assumption over $P(x_j)$ will not violate the result, in other cases this assumption may not be correct. As a result maximum joint probability in P_{B^*} is defined as.

$$P(x_i, x_j|S) = \max\{P(x_i|S \cup \{x_j\})\} = P(x_i'|S \cup \{x_j\}) = 1 - \gamma$$

$$IC_B = -\log(1-\gamma)$$

One can contradict by the following statement, would not exist another random variable pair (x_k,x_l) rather than (x_i,x_j) such that $P(x_k,x_l|S)$ is greater $P(x_i,x_j|S)$. If there is, then IC_B would depend on (x_k,x_l) pair. The answer for this statement starts by assuming that there is a (x_k,x_l) pair satisfying the statement. Then,

$$P(x_k, x_l|S) = P(x_k|S \cup \{x_l\}) * P(x_l) > P(x_i'|S \cup \{x_j\})$$

$$P(x_k|S \cup \{x_l\}) > P(x_i'|S \cup \{x_j\})$$

$$P(x_k|S \cup \{x_l\}) > 1 - \gamma$$

$$P(x_k'|S \cup \{x_l\}) < \gamma$$

The last expression violates the assumption over γ because it was assumed to be the minimum conditional distribution in the network. Therefore, the expression for IC_B was correct.

Then, we got the following expression for σ when $\gamma \rightarrow 0$.

$$\sigma = \min\left\{ f^1(\gamma) = \frac{\gamma^n}{2}, \qquad f^2(\gamma) = \frac{-\log(1-\gamma)}{2^{n+2}|n log\left(\frac{\gamma}{2}\right) + 1|} \right\}$$

Both expressions are 0 when $\gamma \rightarrow 0$. Therefore, we applied the following criteria:

$$define\ A = \lim_{\gamma \to 0^+} \frac{f^1(\gamma)}{f^2(\gamma)}\ then\ \sigma = \begin{cases} f^1(\gamma) & if A < 1 \\ f^2(\gamma) & if A > 1 \\ no\ desicion & if A = 1 \end{cases}$$

Now the aim is to evaluate the limit behavior of two functions when $\gamma \rightarrow 0^+$.

$$\lim_{\gamma\to 0^+}\frac{f^1(\gamma)}{f^2(\gamma)}=\lim_{\gamma\to 0^+}\frac{\frac{\gamma^n}{2}}{\frac{-\log(1-\gamma)}{2^{n+2}|nlog\left(\frac{\gamma}{2}\right)+1|}}=\lim_{\gamma\to 0^+}\frac{\frac{\gamma^n}{2}}{\frac{-\log(1-\gamma)}{2^{n+2}*-(nlog\left(\frac{\gamma}{2}\right)+1)}}$$

$$=\lim_{\gamma\to 0^+}\frac{\frac{\gamma^n}{2}}{\frac{\log(1-\gamma)}{2^{n+2}(nlog\left(\frac{\gamma}{2}\right)+1)}}=\lim_{\gamma\to 0^+}\frac{\frac{\gamma^n}{2}}{\frac{\log(1-\gamma)}{2^{n+2}(nlog(\gamma)-nlog2+1)}}$$

-nlog2+1 is finite, however, *log(γ)* $\to\infty$ when $\gamma\to 0^+$. Therefore,

$$\lim_{\gamma\to 0^+}\frac{f^1(\gamma)}{f^2(\gamma)}=\lim_{\gamma\to 0^+}\frac{\frac{\gamma^n}{2}}{\frac{\log(1-\gamma)}{2^{n+2}nlog(\gamma)}}=\lim_{\gamma\to 0^+}\frac{\gamma^n 2^{n+1}nlog(\gamma)}{\log(1-\gamma)}$$

$$=n2^{n+1}\lim_{\gamma\to 0^+}\frac{\gamma^n log(\gamma)}{\log(1-\gamma)}=n2^{n+1}\lim_{\gamma\to 0^+}\frac{\gamma^{n-2}\gamma\gamma log(\gamma)}{\log(1-\gamma)}\ since\ n\geq 2$$

$$\lim_{\gamma\to 0^+}\frac{f^1(\gamma)}{f^2(\gamma)}=n2^{n+1}\lim_{\gamma\to 0^+}\frac{\gamma^{n-2}\gamma log(\gamma)}{\log(1-\gamma)*1/\gamma}$$

$$=n2^{n+1}\lim_{\gamma\to 0^+}\gamma^{n-2}\lim_{\gamma\to 0^+}\frac{\gamma}{\log(1-\gamma)}\lim_{\gamma\to 0^+}\frac{log(\gamma)}{1/\gamma}$$

The last partition of the limit would be feasible if and only if three limits exist.

$$\lim_{\gamma\to 0^+}\gamma^{n-2}=\begin{cases}0\ if\ n>2\\1\ if\ n=2\end{cases}$$

$$\lim_{\gamma\to 0^+}\frac{\gamma}{\log(1-\gamma)}\xrightarrow{L'Hospital}\lim_{\gamma\to 0^+}\frac{1}{-1/(1-\gamma)}=\lim_{\gamma\to 0^+}\gamma-1=-1$$

$$\lim_{\gamma\to 0^+}\frac{log(\gamma)}{1/\gamma}\xrightarrow{L'Hospital}\lim_{\gamma\to 0^+}\frac{1/\gamma}{-1/\gamma\hat{}2}=\lim_{\gamma\to 0^+}-\gamma=0$$

$$\lim_{\gamma\to 0^+}\frac{f^1(\gamma)}{f^2(\gamma)}=0$$

Therefore, we get the first expression for σ when $\gamma\to 0$

B. Analysis of Figure 4-7 with BDeu and BIC Relation

In this section, our aim is to analyze the characteristics of the BDeu metric on structure learning for various numbers of data samples M. In chapter 3 it was described that the BDeu score converges to BIC score when M converges to infinity. Nonetheless, practically, for larger values of M both scores still perform the same characteristics. Figures 8-1 and 8-2 give simulation results to compare the BDeu and BIC scores for finite sample sizes, which points that for sample size greater than a threshold both metrics behave the same. Note that, prior parameters α_{ijk} in Equation 3.2 have a significant effect on M in the decision when the BDeu score acts like the BIC score. When the prior distributions are high, more samples are needed to see this effect. In this thesis, we had also provided the effect of the prior distribution on it was found that for smaller imaginary sample size we get a result which is similar to the BIC score. The reason behind having the same relation for BDeu and BIC scores for finite sample sizes is the use of smaller imaginary sample sizes. Notice that we had discussed this issue in chapter 4 and concluded that smaller but fair imaginary sample sizes are suitable for model discovery on dDBNs.

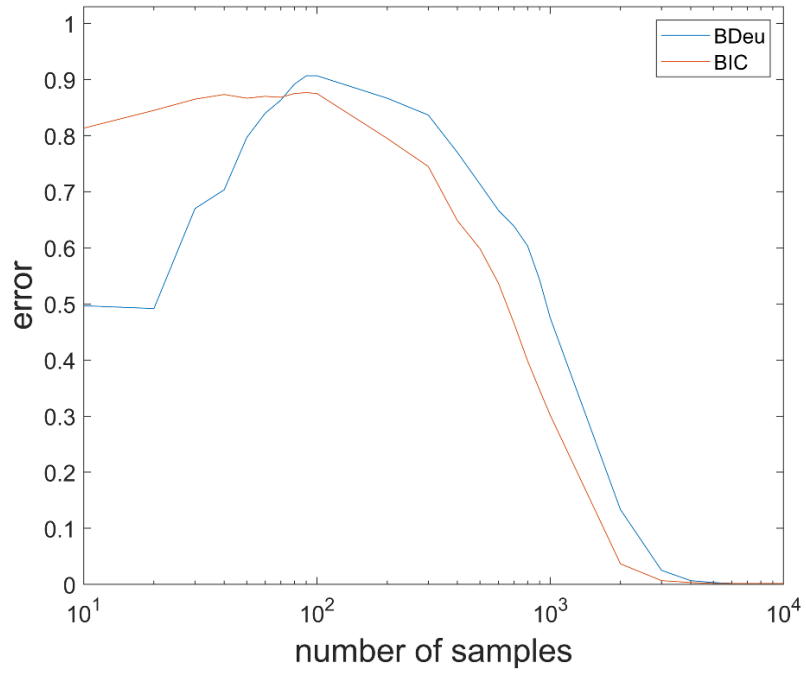

Figure 8-1: Mean error vs number of samples of a binary node has six parents for 6-node network using BDeu and BIC scores.

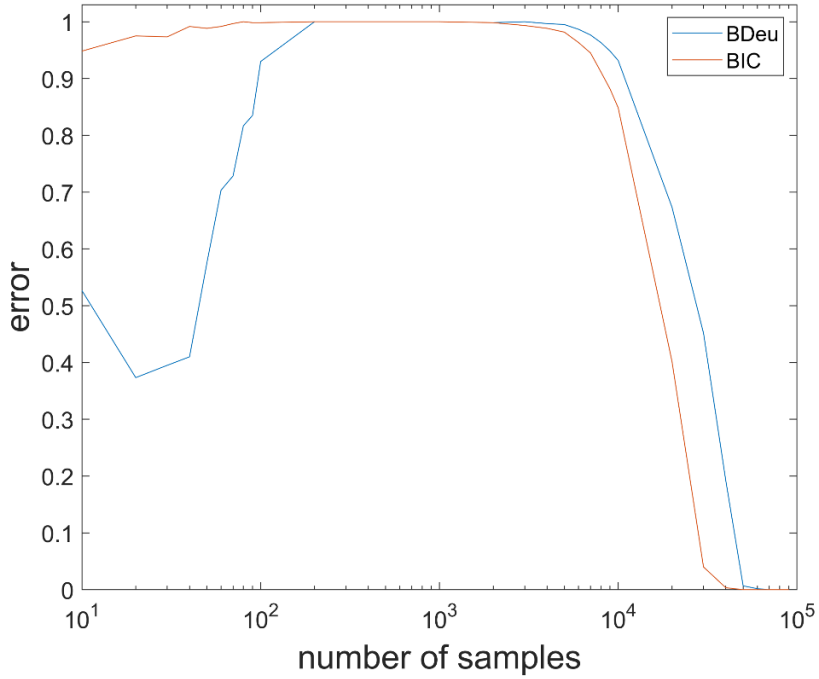

Figure 8-2: Mean error vs number of samples of a ternary node has six parents for 6-node network using BDeu and BIC scores.

Figure 8-3 shows the structural error for a ternary valued network consisting of five variables where only the mean error of the node that has 5 parents is shown. This figure illustrates the error between the true structure from which the data was sampled, and the structure found with dDBN learning using this data. The graph seems to have three regions. In the first region, the error is around 0.5. However, when more data is provided to the dDBN learning procedure, error reaches to 1 and stays constant, despite the sample size increases. In this second region, although the true structure contains five parents, the found structure does not contain any edges. In other words, learning ended up with an empty structure. If the amount of data is further increased, in the third region, the algorithm starts to add some parents to the structure and error starts to decrease. When a sufficient amount of data is provided, all parental relations are found correctly by the dDBN structure learning, and error reaches 0.

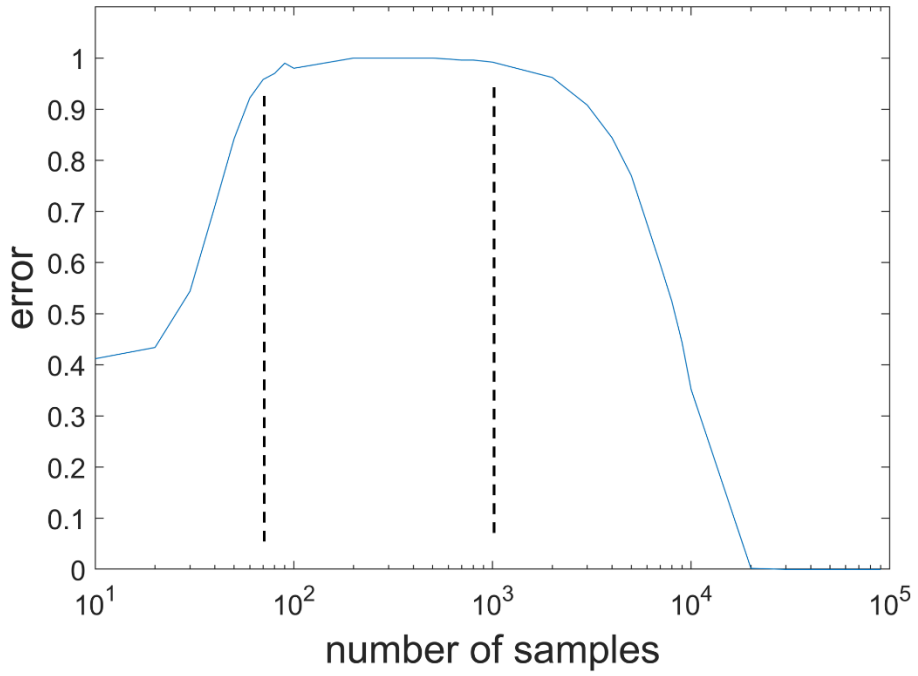

Figure 8-3: Mean error vs number of samples for the node which has 5 parents in a network of 5 ternary variables

In the first region, the amount of data samples is smaller than 70. When the data size is very small, dDBN structure learning ends up with a structure as if it was chosen randomly and does not contain any information about the actual structure. It means that this amount of data has no effect on learning and the BDeu score of each possible graph is likely to be randomly generated numbers. The maximum scored structure is just a randomly generated array which consists of 0's and 1's. Figures 4-2 and 4-3 also illustrate this behavior. When the data size is smaller than a threshold, the structural error is around 0.5 and changing the parent size does not affect this result. This shows that the found best structures consist of edges as if they were chosen randomly.

In the second region, the error is highest and stays so for the number of samples M from 70 to 1000. The actual structure from which the data was generated contains all the edges, i.e., fully connected. Getting structural error to be 1 means that the structure found by the dDBN learning did not include any 1's, hence the learning procedure tries to obstruct any edges and the BDeu score of the empty structure is higher than any other possible structures. The reason for this behavior can be explained by analyzing Equation 3.3. Since the sample size is not sufficient, the likelihood term of Equation 3.3 does not differentiate the actual structure from any other structure and gives almost the same likelihood score for every possible structure. As a result of that, the second term related to the model dimension dominates the BDeu score. The BDeu score tries to decrease model complexity by this term. Therefore, the structure with the lowest model complexity or least number of edges will get the highest score. As a result, the learning procedure ends up with an empty structure and this is the reason why the structural hamming distance between the actual and learned structure is 1.

Finally, in the third region, the dDBN learning algorithm starts to learn the parents of the node correctly. While M increases, the mean error on the structures reduces and after a certain value of M error becomes 0, which means that the structure is learned from the data correctly.

C. The Comparison of Discretization Techniques

Table 8-1: The comparison of discretization methods. The table is sorted according to the accuracy of the methods. “der” means that firstly, the derivative of the synthetic data is computed then discretization methods are applied.

Name	level	TP	FP	TN	FN	TPR	FPR	PPV	Accuracy
der + EFD3	3	14,95	2,45	15,55	3,05	0,83	0,14	0,86	0,8473
der + topdown40	3	14,59	2,23	15,77	3,41	0,81	0,12	0,87	0,8432
der + mean_std25	3	14,58	2,23	15,77	3,43	0,81	0,12	0,87	0,8429
der + mean_std50	3	14,94	2,60	15,40	3,06	0,83	0,14	0,85	0,8427
der + topdown30	3	14,89	2,66	15,34	3,11	0,83	0,15	0,85	0,8397
der + 3means	3	14,77	2,92	15,08	3,23	0,82	0,16	0,83	0,8291
ji_tan33	3	13,67	2,92	15,09	4,33	0,76	0,16	0,82	0,7988
der + topdown20	3	14,28	3,53	14,47	3,72	0,79	0,20	0,80	0,7985
der + mean_std100	3	13,92	3,88	14,12	4,08	0,77	0,22	0,78	0,7789
ji_tan50	3	13,31	3,33	14,67	4,69	0,74	0,19	0,80	0,7772
ji_tan67	3	12,80	3,70	14,30	5,20	0,71	0,21	0,78	0,7529
TSD00	2	15,64	6,63	11,37	2,36	0,87	0,37	0,70	0,7503
der + mean2	2	15,63	6,64	11,37	2,37	0,87	0,37	0,70	0,7498
der + 2means	2	15,62	6,63	11,37	2,38	0,87	0,37	0,70	0,7498
der + midRange	2	15,62	6,64	11,36	2,38	0,87	0,37	0,70	0,7495
der + top50	2	15,62	6,64	11,36	2,38	0,87	0,37	0,70	0,7495
der + EFD2	2	15,62	6,64	11,36	2,38	0,87	0,37	0,70	0,7495
der + maxmin67	3	13,26	4,29	13,72	4,75	0,74	0,24	0,76	0,7492
der + TDT	2	15,62	6,77	11,23	2,38	0,87	0,38	0,70	0,7458
der + EWD3	3	13,24	4,43	13,57	4,76	0,74	0,25	0,75	0,7449
der + topdown10	3	13,01	4,30	13,70	4,99	0,72	0,24	0,75	0,7419
der + EWD2	2	15,48	7,04	10,96	2,52	0,86	0,39	0,69	0,7345
EFD3	3	12,40	4,39	13,61	5,60	0,69	0,24	0,74	0,7224
mean_std50	3	12,46	4,54	13,46	5,54	0,69	0,25	0,73	0,7199
mean_std25	3	11,96	4,05	13,95	6,04	0,66	0,23	0,75	0,7198

topdown40	3	11,96	4,07	13,93	6,04	0,66	0,23	0,75	0,7193
topdown30	3	12,49	4,60	13,40	5,51	0,69	0,26	0,73	0,7190
3means	3	12,51	4,80	13,20	5,49	0,70	0,27	0,72	0,7143
der + mean_std150	3	12,05	4,40	13,60	5,96	0,67	0,24	0,73	0,7123
topdown20	3	12,34	5,24	12,76	5,66	0,69	0,29	0,70	0,6973
midRange	2	14,47	7,52	10,48	3,53	0,80	0,42	0,66	0,6930
EFD2	2	14,47	7,52	10,48	3,53	0,80	0,42	0,66	0,6930
top50	2	14,46	7,53	10,47	3,54	0,80	0,42	0,66	0,6925
mean2	2	14,45	7,53	10,47	3,55	0,80	0,42	0,66	0,6921
2means	2	14,44	7,55	10,46	3,56	0,80	0,42	0,66	0,6915
TDT	2	14,47	7,66	10,34	3,53	0,80	0,43	0,65	0,6893
TSD25	2	14,86	8,24	9,76	3,14	0,83	0,46	0,64	0,6840
EWD2	2	14,40	7,79	10,21	3,60	0,80	0,43	0,65	0,6837
mean_std100	3	12,07	5,47	12,53	5,93	0,67	0,30	0,69	0,6834
der + w_maxmin10	3	10,14	3,64	14,36	7,86	0,56	0,20	0,74	0,6805
der + maxmin50	3	10,57	4,13	13,87	7,43	0,59	0,23	0,72	0,6789
maxmin67	3	11,70	5,54	12,46	6,31	0,65	0,31	0,68	0,6709
EWD3	3	11,79	5,73	12,27	6,21	0,66	0,32	0,67	0,6684
der + top25	2	15,05	9,03	8,97	2,95	0,84	0,50	0,63	0,6673
der + top75	2	15,02	9,02	8,98	2,98	0,83	0,50	0,62	0,6666
topdown10	3	11,41	5,54	12,46	6,59	0,63	0,31	0,67	0,6631
der + w_maxmin15	3	9,13	3,33	14,67	8,87	0,51	0,18	0,73	0,6613
der + max75	2	14,65	9,27	8,73	3,35	0,81	0,52	0,61	0,6493
mean_std150	3	10,71	5,38	12,62	7,29	0,60	0,30	0,67	0,6482
w_maxmin10	3	10,52	5,26	12,74	7,48	0,58	0,29	0,67	0,6462
TSD50	2	13,66	8,69	9,31	4,34	0,76	0,48	0,61	0,6379
w_maxmin15	3	10,03	5,07	12,93	7,97	0,56	0,28	0,66	0,6376
mean_time	3	11,16	6,28	11,73	6,84	0,62	0,35	0,64	0,6357
der + maxmin33	3	7,66	2,92	15,08	10,34	0,43	0,16	0,72	0,6317
maxmin50	3	9,70	5,04	12,96	8,30	0,54	0,28	0,66	0,6293
top25	2	14,25	9,65	8,35	3,75	0,79	0,54	0,60	0,6277
top75	2	14,20	9,68	8,32	3,80	0,79	0,54	0,59	0,6254
max75	2	13,84	9,80	8,20	4,16	0,77	0,54	0,59	0,6122

der + bi2means	2	10,51	6,50	11,50	7,49	0,58	0,36	0,62	0,6114
maxmin33	3	7,45	3,87	14,13	10,55	0,41	0,22	0,66	0,5994
der + max50	2	10,26	6,87	11,13	7,74	0,57	0,38	0,60	0,5944
der + bi3means	3	5,51	2,50	15,50	12,49	0,31	0,14	0,69	0,5836
TSD100	2	11,15	8,21	9,79	6,85	0,62	0,46	0,58	0,5819
bi2means	2	10,64	7,81	10,19	7,37	0,59	0,43	0,58	0,5786
max50	2	9,54	7,10	10,90	8,46	0,53	0,39	0,57	0,5678
bi3means	3	5,78	3,84	14,16	12,22	0,32	0,21	0,60	0,5539
TSD150	2	9,63	7,74	10,26	8,37	0,54	0,43	0,55	0,5526
der + max25	2	4,82	3,01	14,99	13,19	0,27	0,17	0,62	0,5501
max25	2	5,40	4,12	13,88	12,60	0,30	0,23	0,57	0,5356

D. Figures for the Effect Imaginary Sample Size on Model Discovery

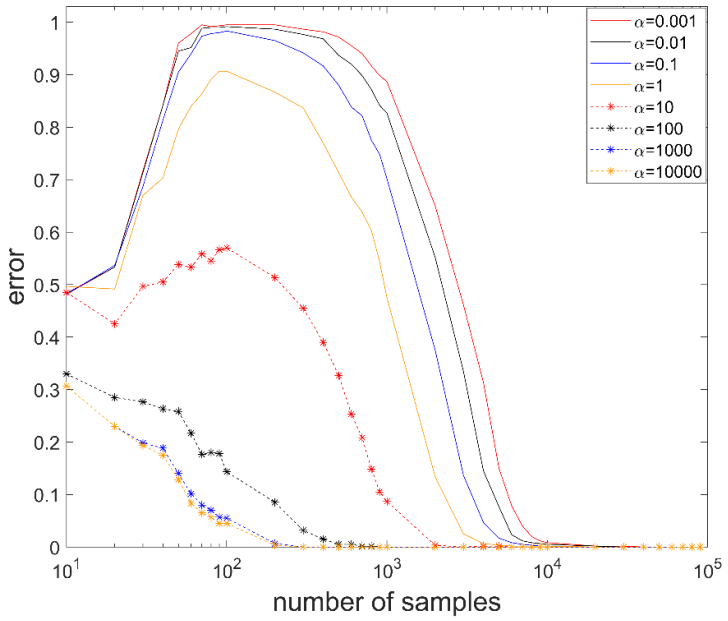

Figure 8-4: Mean error vs number of samples for different alpha values used for six-node binary-valued network. It presents for a node that has six parents.

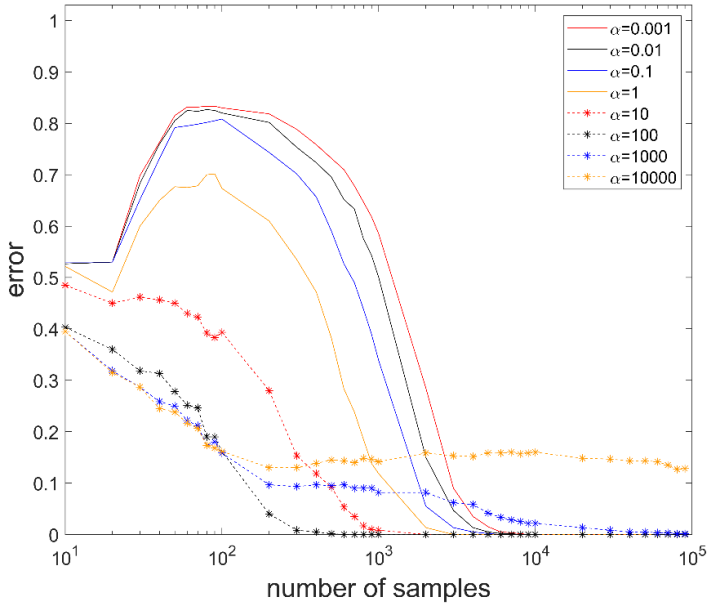

Figure 8-5: Mean error vs number of samples for different alpha values used for six-node binary-valued network. It presents for a node that has five parents.

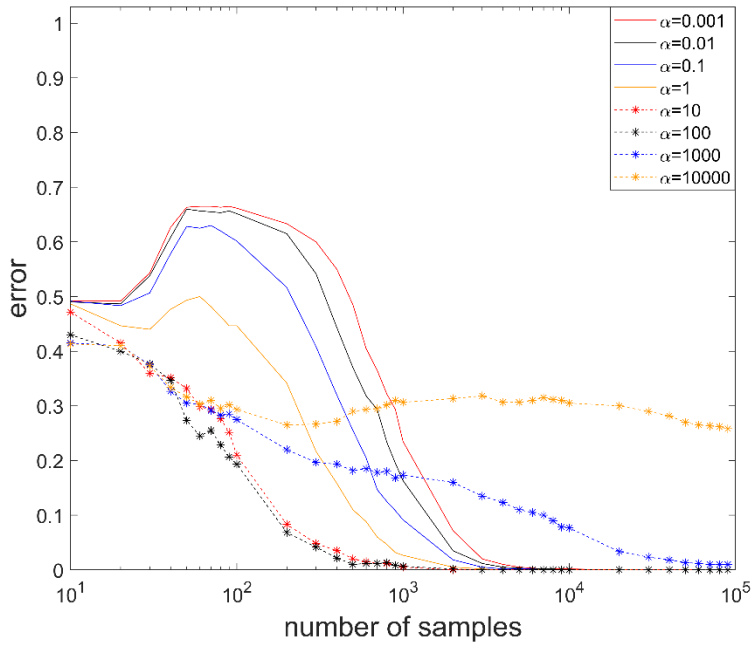

Figure 8-6: Mean error vs number of samples for different alpha values used for six-node binary-valued network. It presents for a node that has four parents.

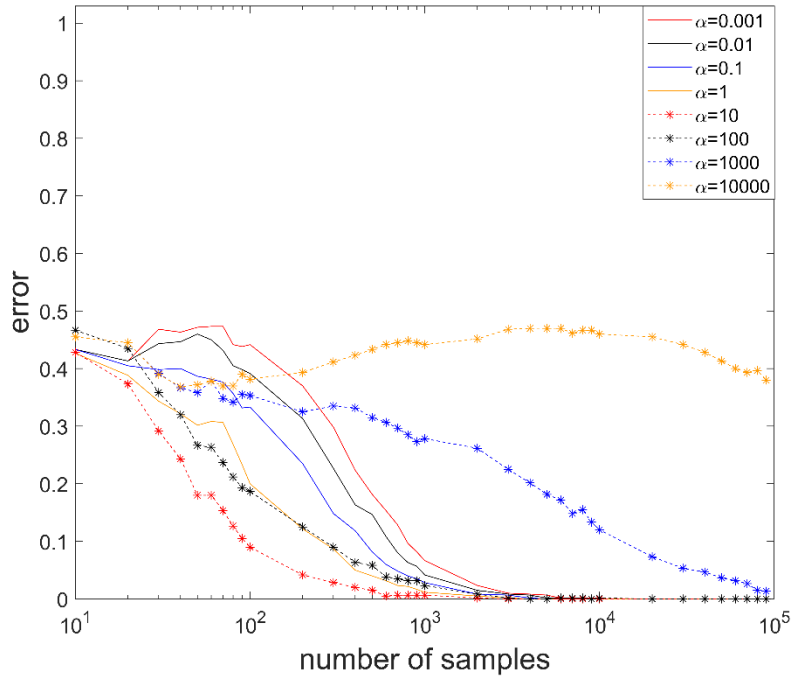

Figure 8-7: Mean error vs number of samples for different alpha values used for six-node binary-valued network. It presents for a node that has three parents.

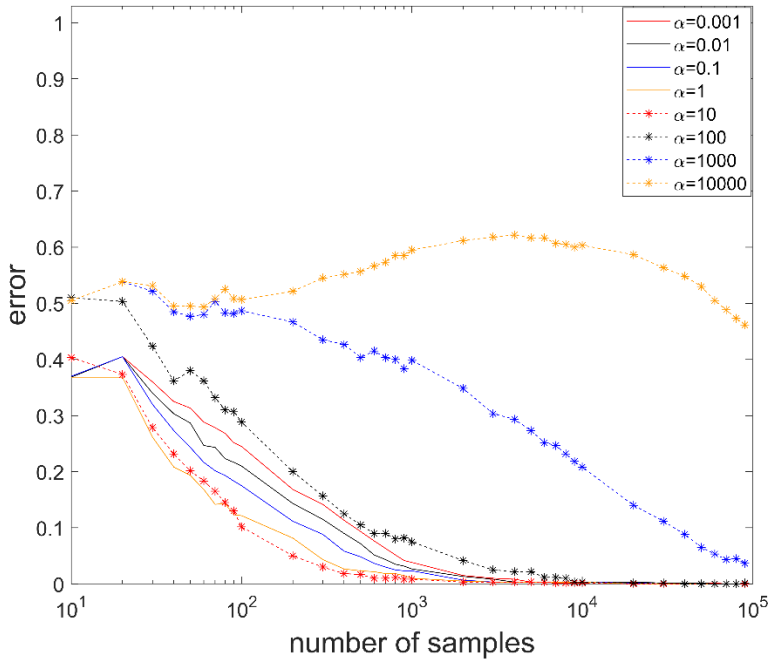

Figure 8-8: Mean error vs number of samples for different alpha values used for six-node binary-valued network. It presents for a node that has two parents.

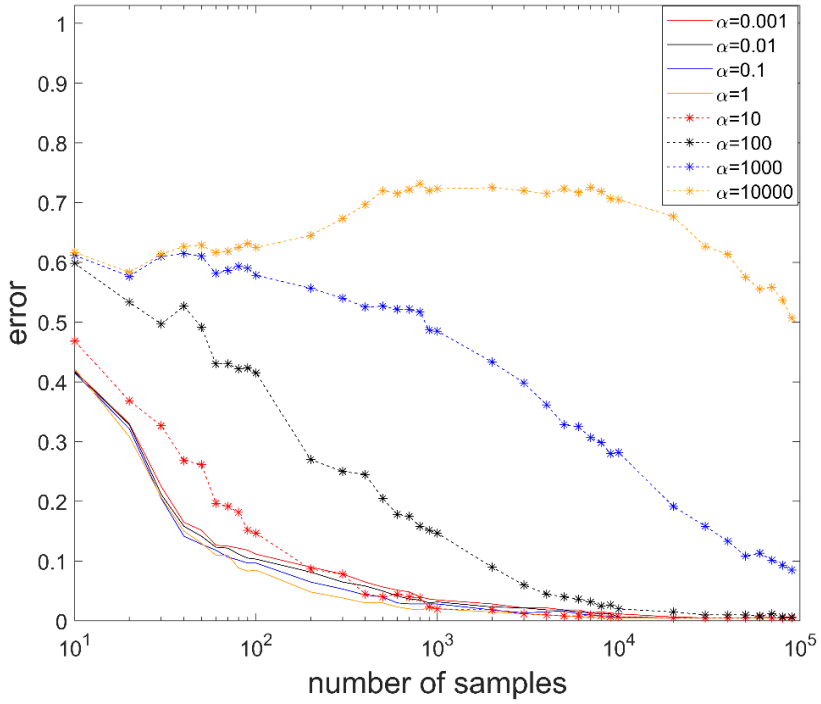

Figure 8-9: Mean error vs number of samples for different alpha values used for six-node binary-valued network. It presents for a node that has one parent.